Research Product 97-15

Training Computer Skills for the Future Battlefield: A Review and Annotated Bibliography

May H. Throne and Carl W. Lickteig
U.S. Army Research Institute

19980217 557

August 1997

Armored Forces Research Unit

U.S. Army Research Institute for the Behavioral and Social Sciences

U.S. ARMY RESEARCH INSTITUTE
FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Director

Technical review by

Michael Drillings
Ted Schlecter

NOTICES

REPORT DOCUMENTATION PAGE

1. REPORT DATE (dd-mm-yy) 1997, August	2. REPORT TYPE Final	3. DATES COVERED (from. . . to) June 1996-July 1997
4. TITLE AND SUBTITLE Training Computer Skills for the Future Battlefield: A Review and Annotated Bibliography		5a. CONTRACT OR GRANT NUMBER
		5b. PROGRAM ELEMENT NUMBER 0602785A
6. AUTHOR(S) May H. Throne and Carl W. Lickteig		5c. PROJECT NUMBER A791
		5d. TASK NUMBER 2228
		5e. WORK UNIT NUMBER H01
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Institute for the Behavioral and Social Sciences Armored Forces Research Unit ATTN: PERI-IK Fort Knox, KY 40121-5620		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Institute for the Behavioral and Social Sciences 5001 Eisenhower Avenue Alexandria, VA 22333-5600		10. MONITOR ACRONYM ARI
		11. MONITOR REPORT NUMBER Research Product 97-15

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

May H. Throne is a graduate student at the University of Louisville working as a consortium fellow at the U.S. Army Research Institute's Fort Knox Research Unit.

14. ABSTRACT *(Maximum 200 words)*:

As the Army moves toward a digital battlefield, the nation's defense will become reliant on the computer skills of its leaders, soldiers, and civilians. To embody this future force, Army training must successfully address the acquisition, retention, and transfer of computer skills. As a first step toward this goal, this research product reviews the literature concerning the acquisition, retention, and transfer of computer-based skills. A review of 76 articles examining the training domains of programming, software, simulation, and gaming ability was performed. General conclusions for each training area (acquisition, retention, and transfer) are presented. In general, the research does not build on previous findings in the area. In addition, many areas, such as the long-term retention of computer skills and individual difference variables, remain to be explored.

15. SUBJECT TERMS

Retention Acquisition Transfer Computer skills End-user skills

SECURITY CLASSIFICATION OF			19. LIMITATION OF ABSTRACT	20. NUMBER OF PAGES	21. RESPONSIBLE PERSON (Name and Telephone Number)
16. REPORT Unclassified	17. ABSTRACT Unclassified	18. THIS PAGE Unclassified	Unlimited	67	

Research Product 97-15

Training Computer Skills for the Future Battlefield: A Review and Annotated Bibliography

May H. Throne and Carl W. Lickteig
U.S. Army Research Institute

Armored Forces Research Unit
Barbara A. Black, Chief

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

August 1997

Army Project Number
20262785A791

Education and Training Technology

FOREWORD

The U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) is dedicated to the development of training and evaluation methods to meet future battlefield conditions. The Future Battlefield Conditions' work is performed under Work Package 2228, FASTTRAIN, Force XXI Training Methods and Strategies. This research on training requirements and evaluation methods is supported by a Memorandum of Agreement between the U.S. Army Armor Center and ARI titled Manpower, Personnel and Training Research, Development, Test, and Evaluation for the Mounted Forces, 16 October 1995.

As the Army forms an information-age force for the 21st century, the nation's defense will progressively rely on the computer-based skills of its leaders, soldiers, and civilians. To embody this future force, Army training must successfully address the acquisition, retention, and transfer of computer skills. The need to ensure soldiers can skillfully perform computer-based tasks was underscored by the Army's ongoing series of Advanced Warfighting Experiments.

This review and bibliography on computer skills training supports the military's requirement for a better understanding of soldier and system performance on the future battlefield. The bibliography assembles available literature on computer skills training from commercial, educational, and military domains. The annotations summarize training issues, methods, and findings to direct future training efforts. The review consolidates the referenced literature by extracting key lessons learned categorized by the acquisition, retention, and transfer of computer skills. The information contained in this research product will be provided to commanders, trainers, and civilian training developers at the U.S. Army Armor Center and School at Fort Knox and will be available Army-wide.

ZITA M. SIMUTIS
Technical Director

EDGAR M. JOHNSON
Director

TRAINING COMPUTER SKILLS FOR THE FUTURE BATTLEFIELD: A REVIEW AND ANNOTATED BIBLIOGRAPHY

CONTENTS

LIST OF TABLES

Training Computer Skills for the Future Battlefield: A Review and Annotated Bibliography

Introduction

The U.S. Army's force modernization plan assumes future force dominance depends on the exploitation of advanced information technologies (U.S. Department of the Army, 1995). This transformation to an information age force presumes that all soldiers in the areas of combat, combat support, and combat service support will be provided with access to networked computer-based information systems (Decker, 1996). As the Army shapes this digital battlefield, the nation's defense will progressively rely on the computer-based skills of its leaders, soldiers, and civilians. To embody this future force, Army training must successfully address the acquisition, retention, and transfer of computer skills.

Overview

The need to ensure soldiers could skillfully perform computer-based tasks was underscored by the Army's ongoing series of Advanced Warfighting Experiments (AWEs). The AWEs are an integrated series of military exercises in which combatants and their supporters, at all echelon levels and across all functional areas, are equipped with the networked digital information systems anticipated on the digital battlefield. These computer-based digital systems, referred to as Command, Control, Communication, Computer and Intelligence (C^4I) Systems, serve as the tools required to perform the command and control functions directed at mission accomplishment.

One of the first AWEs, Desert Hammer VI, concluded that computer-based information systems are no substitute for fundamental combat skills (U.S. Army Armor Center, 1994). This AWE proposed that the digital battlefield would require a three phase, hierarchical approach to training: fundamental combat skills; computer skills; and, finally, the integration of combat and computer skills in a warfighting environment. More recently, the AWE Focused Dispatch reinforced these conclusions and stressed that future military training must address acquisition and retention issues associated with computer-based skills (U.S. Army Armor Center, 1996). A resounding conclusion from Focused Dispatch was that: "Digital skills are highly perishable" (p. 1-7).

The Army's modernization effort will profoundly impact how the Army trains, maintains, and operates as an information-age force in the 21st century. Accordingly, a recent study by the Army Science Board (1995) challenged training researchers to address some of the fundamental concerns about job and educational technology.

> ...how long is the material retained? How is it useful throughout the course of an officer's career? Is interactive computer-based learning a better way to have students participate in the learning process? Or is pure experiential learning most effective? These are difficult questions, but their complexity should not hinder attempts to answer them. (p. 36)

This annotated bibliography focuses on the training of computer skills to support the military's requirement for a better understanding of soldier and system performance on the future battlefield. The bibliography assembles available literature on computer skills training from commercial, educational, and military domains. The annotations summarize training issues, methods, and findings to direct future training efforts. The introduction provides a background for the military requirement for computer skills training and provides a working definition of computer skills to demarcate the area of concern. It also provides a training context for computer skills amid contemporary learning theory and traditional principles and guidelines for skills training. The key lessons learned from the annotated bibliography are then consolidated and categorized by the acquisition, retention, and transfer of computer skills. Finally, several perceived limitations of the findings are identified as well as some directions for future research on computer skills training.

Computer Skills

For this report, computer skills are defined as learned perceptual, affective, mental, and behavioral processes required for acceptable performance levels in computer-mediated work. This definition is based on Gattiker's computer-based efforts in the areas of skill acquisition (1992) and end-user training (1990), and Adams' (1987) more basic work in the area of human skills. Adams readily admits to the difficulty of providing a simple and useful definition of a skill, but cites three defining characteristics. According to Adams, skill is a wide behavioral domain that includes complex behaviors; a skill is learned or trained; and, goal attainment is importantly dependent upon motor behavior.

This report's broad definition of computer skills reflects the holistic nature of contemporary learning theories and the complexity of customizing training to individuals and realistic job situations (Reigeluth, 1996). Advances in computers and the information they provide will continue to redefine a worker's job domain, task requirements, and computer skill requirements. This progression is directing learning theorists' attention to notions such as continuous learning, life-long training, automated training, and just-in-time training, that may redefine the very concept of training.

It may be helpful to distinguish computer skills, as defined herein, from the terms computer literacy and computer-based instruction. A formal definition of computer literacy is "...whatever understanding, skills, and attitudes one needs to function effectively within a given social role that directly or indirectly involves computers" (Andersen & Klassen, 1981, p. 131). The level of literacy required for effective functioning is clearly dependent upon the user's role and its functional relationship to computers. For example, the potential training audience requiring computer skills to apply C^4I systems encompasses many roles. Overall, computer literacy is a multi-dimensional term that includes: hardware, software, programming, applications, social and organizational impact, and affect.

For this report, the application dimension of computer literacy is the subject-matter, in particular the computer skills required for C^4I system applications. Applying C^4I systems also includes Gattiker's (1990) reference to Level 2 computer skills directed at understanding the *possibilities* of a C^4I system for mission accomplishment.

Computer-based instruction is a broad category that generally refers to the use of computers as a tutor or trainer (Kulik, 1994). As a tutor, the computer may be used to present material, evaluate responses, determine what comes next, record progress. The focus of this product on computer skills equates to the computer's role as a tool, even an intelligent or adaptive tool, that mediates end-user job requirements. The computer tool supports doing something, such as computer-mediated work, and the computer tutor supports learning something (not necessarily computer related), such as math or physics.

Computers also serve multiple roles or purposes, especially in the area of computer-mediated work, since more than 50 million workers use computers everyday. Therefore, for reasons including efficiency and economy, the worker's tool is rapidly becoming the worker's tutor (Cascio, 1995). This trend spurs the growing interest in continuous or life-long training and the increased emphasis on computer-based performance support systems, embedded training, and just-in-time training (Collins & Throne, 1996).

Computer Skill Acquisition, Retention, and Transfer

This product focuses on the acquisition, retention, and transfer of computer skills. It does not encompass the general literature on learning, instruction, and training. An underlying assumption of this review, however, is that much of what is known about the acquisition, retention, and transfer of skills is independent of the skill domain or training medium. Until evidenced otherwise, general skill training principles and guidelines should apply to training for computer skills. A recent and comprehensive set of training guidelines for acquisition, retention, and transfer of skills, in general, is provided by Swezey and Llaneras (1997).

Military training principles and guidelines for procedural skills training should also apply to computer skill training (e.g., Rowatt and Shlechter, 1993). These principles and guidelines support fundamental training issues, to include: level, duration, amount, and spacing of training; type, complexity and organization of tasks; and ability, experience, style, and motivation of trainees. Efforts to use the computer as a tutor have resulted in medium-specific principles and guidelines for skill training (e.g., Eberts & Brock, 1984). Such principles and guidelines should also apply to computer skills training.

Finally, computer skills training should heed recent reminders to avoid the illusions of training and ensure training for performance (Druckman & Bjork, 1994). These authors' empirically-based analysis documents the tendency of trainees and trainers to inadvertently fool themselves and each other about what and how much has been learned and trained. They also suggest that the traditional climate of military training, including pressures for efficiency and intolerance of errors, may reinforce such tendencies. The contemporary focus on learner-controlled training may also accentuate a trainee's illusions about knowing. It may also reduce the trainer's ability to check-and-balance the trainee's and organization's illusions about preparedness.

In particular, Druckman & Bjork (1994) stress that the purpose of training is post-training performance. Their analysis bolsters the proposition that training conditions that enhance skill acquisition may retard skill retention and transfer to job performance. Prevalent training conditions--massed practice, immediate and frequent feedback, and constant task conditions--deter retention and transfer. In sum, a more effort, more effect approach to training may slow skill acquisition but enhance retention and transfer. The principles and guidelines Druckman & Bjork (1994) provide for avoiding training illusions and ensuring skilled performance should also apply to computer skills training.

Method

This product is based on a systematic search of the literature related to computer skills acquisition, retention, and transfer from commercial, educational, and military domains. Search terms for each of the literature databases, listed below, include individually pairing computers and digital skills with the following: acquisition, learning, memory, retention, recall, decay, transfer, and training. The phrase computer skills was paired with acquisition, learning, retention, and decay. The phrase computer literacy was paired with retention. Finally, the phrase software skills was paired with learning, acquisition, and retention.

Based on these terms, a search was conducted of the following databases: PsycLIT (both journal articles and book chapters), Educational Resources Information Center (ERIC), NewsBank, Dissertation Abstracts on Disc (1861-1995), Government Documents Catalog Service, Air University Index, PubQ, Manpower and Training Research Information System (MATRIS), and Defense Technical Information Center (DTIC). In addition, recent publications (1990-1996) of the following journals were searched by table of contents and indices: Annual Review of Psychology, Cognitive Science, Computers in Human Behavior, Education and Computing, Educational Research, Educational Researcher, Educational Technology Research & Development, Educational Technology, Electronic Learning, Journal of Educational Computing Research, Journal of Research on Computing in Education, Review of Educational Research, and THE Journal: Technological Horizons in Education.

The articles presented herein reflect a detailed search of the specified resources. This product, however, is not considered an exhaustive search of computer skills training. A perceived difficulty in conducting a comprehensive search on computer skills training, is an inconsistency in the use of keywords among researchers in this and related areas. As with computer-based instruction, computer skills training is a broad category rather than a technical term (Kulik, 1994). Related disciplines examining computer skills have not developed precise definitions or a coherent conceptual base. For example, use of the key phrase computer skills resulted in only a few articles. Additionally, some articles that could not be obtained (e.g., were not available through interlibrary loan, were not yet published, or were out of print publications) are referenced here in a separate appendix, but not annotated or included in the Findings section.

For each article in the bibliography, an annotated summary identifies and compares training issues, methods, and findings that might inform future training efforts. These summaries are intentionally more detailed than abstracts in order to provide more information on the procedures, findings, and conclusions of the articles presented. All results reported in the findings and annotations are significant at $p < .05$, unless otherwise noted in an annotation. In the annotations, the words experiment and subjects are reserved for research efforts with designs and procedures that yielded more reliable and comparable results. The words study and participants are reserved for annotated efforts based on less discriminating designs and procedures. These classifications of the annotated efforts are based on the judgment of the authors.

Findings

A summary table of the articles found in the annotated bibliography is provided in Table 1. This table categorizes each article's primary focus with respect to computer skill training areas: acquisition, retention, and/or transfer. This table also categorizes each article's training domain: software programming, simulation and games, and/or software applications. Appendix A provides an annotated bibliography of each article on computer skills training included in this product. The articles in this appendix are ordered alphabetically by first author. As noted, the annotations identify training issues, methods, and findings that might guide future training efforts. Appendix B lists related articles not obtained for annotation, as previously noted, and Appendix C defines any acronyms used in this product.

The following section briefly consolidates the referenced literature by extracting key lessons learned categorized by the acquisition, retention, and transfer of computer skills. There is a wide but shallow body of research on the acquisition, retention, and transfer of computer skills. The area is relatively new and not well defined or organized, as indicated by the discussion of search terms. Therefore, some tentative lessons learned are highlighted that might guide research and training on computer skills. Each of these lessons learned is based on the efforts annotated, with particular emphasis on findings corroborated across multiple efforts.

Computer Skill Acquisition Findings

1. Declarative and procedural knowledge of basic job skills should be acquired before acquiring supporting computer skills.

2. Novices apply their available mental models to computer skills, sometimes erroneously (e.g., typewriter model to word processing applications).

3. Guided exploration improves computer skill acquisition more than just reading a manual.

4. Active participation increases learning.

5. Novice-level skills are acquired faster with direct manipulation or windows-based applications (e.g., Macintosh & Windows) than with command-based systems (e.g., DOS).

6. Advance organizers, also known as concrete models (e.g., the desktop metaphor), may improve computer skill acquisition.

Computer Skill Retention Findings

1. Overpractice increases the retention of computer skills.

2. Computer-skills training based on a problem-solving strategy may slow acquisition, but increase retention.

3. Process knowledge about how an application operates is retained better than the interface-based operating procedures.

4. Commands with more specific names are retained better than commands with more generic names.

Computer Skill Transfer Findings

1. Part training of computer skills may transfer more effectively than adaptive (simple to more complex) training.

2. Guided exploration may improve computer skill transfer more than just reading a manual.

3. Partial or abbreviated training may result in less transfer than more complete training.

4. Computer-assisted instruction (CAI) and simulation may lead to greater transfer than actual system training, and CAI is often cheaper than simulation.

5. Error-correction training may result in more accurate performance on computer-based transfer tasks.

6. Computer-skills training based on a problem-solving strategy may slow acquisition, but increase transfer.

Discussion and Conclusions

Our dependency on computers and the skills to use and apply them in our daily work and other activities will increase as the information age matures. Currently, a lack of computer skills often decreases one's chances of finding or doing a good job. Similarly, the nation's defense relies heavily on the computer-based skills of its leaders, soldiers, and civilians.

Overall, it is suggested that the research reviewed does not afford a solid empirical base for consensus on findings about computer skills training, but it is a valid starting point. Until evidenced otherwise, it is concluded that much of what is known about the acquisition, retention, and transfer of skills is independent of the skill domain or training medium. Therefore, general principles and guidelines on skill training should apply to computer skills training.

The highlighted lessons learned and the annotated summaries provide more specific guidance the Army might apply to computer skills training. To promote the acquisition of computer skills, for example, training programs might address trainee's mental

Table 1

Summary Table of Annotated Articles by Training Area and Domain

Authors	Training Area			Training Domain		
	Acquisition	Retention	Transfer	Software Programming	Software Applications	Simulation and Games
Allwood et al., 1987	✓				✓	
Baker et al., 1989		✓				✓
Barnard et al., 1982	✓	✓			✓	
Battista et al., 1984	✓			✓	✓	
Booth et al., 1989	✓					
Britton, 1995	✓					
Callahan, et al., 1993	✓				✓	
Carroll et al., 1983	✓				✓	
Carroll et al., 1984	✓				✓	
Carroll et al., 1985	✓		✓		✓	
Catrambone, 1990	✓		✓		✓	
Charney et al., 1990	✓	✓			✓	
Christoffersen et al., 1996	✓		✓			✓
Cicchinelli et al., 1980	✓		✓			✓
Czaja et al., 1986	✓				✓	
Czaja et al., 1989	✓				✓	
Davies et al., 1989	✓				✓	
Davis et al., 1992	✓				✓	
Deck et al., 1984	✓	✓		✓	✓	
Dray et al., 1981	✓				✓	
Dyck, 1995	✓		✓		✓	
Eason, 1989	✓	✓				
Egan et al., 1985	✓	✓			✓	
Elias et al., 1987	✓				✓	
Finnegan, 1977			✓			✓
Frese et al., 1989	✓				✓	
Frese et al., 1991	✓		✓		✓	
Gattiker, 1990	✓				✓	
Gattiker, 1992	✓					
Gattiker et al., 1987	✓			✓		
Gerhardt-Powals, 1996	✓					✓
Gist et al., 1988	✓				✓	
Gist et al., 1989	✓				✓	
Goettl, 1993			✓			✓
Gomez et al., 1986	✓	✓	✓		✓	
Harrison et al., 1992	✓				✓	
Jones, 1989		✓				✓
Kagan et al., 1984	✓			✓		
Kay, 1990	✓			✓	✓	
Keeler et al., 1995	✓				✓	
Kennedy, 1975	✓				✓	
Kiger, 1984	✓				✓	
Kozma, 1991	✓					
Lee et al., 1994	✓				✓	
Lewis et al., 1981	✓				✓	
Mack et al., 1983					✓	
Mandinach et al., 1987	✓			✓		

(table continues)

Authors	Training Area			Training Domain		
	Acquisition	Retention	Transfer	Software Programming	Software Applications	Simulation and Games
Mané et al., 1989	✓		✓			✓
Marcoulides, 1988	✓				✓	
Margono et al., 1993	✓				✓	
Marshall et al., 1989		✓			✓	
Martocchio et al., 1992	✓				✓	
Mayer, 1975	✓			✓		
Mayer, 1981				✓		
O'Quinn et al., 1987	✓				✓	
Palmiter et al., 1989	✓	✓	✓		✓	
Palmiter et al., 1991	✓	✓	✓		✓	
Palmiter et al., 1993	✓	✓	✓		✓	
Parton et al., 1985	✓				✓	
Pea et al., 1984	✓			✓		
Piedmont, 1991	✓				✓	
Reiser, 1995	✓			✓		
Ricci et al., 1996	✓	✓				✓
Ross et al., 1991	✓	✓			✓	
Russon et al., 1994	✓				✓	
Salisbury, 1990	✓			✓		
Salomon, 1990					✓	
Savage et al., 1984	✓			✓	✓	
Sein et al., 1989	✓				✓	
Shneiderman, 1987	✓	✓				
Shute et al., 1995	✓	✓			✓	
Simon et al., 1996	✓	✓			✓	
Singley et al., 1989			✓		✓	
Wallace et al., 1993	✓				✓	
Webster et al., 1993	✓				✓	
Wilson et al., 1985	✓				✓	

models of the task situation and inherent errors in those models. Similarly, to improve retention, the Army's training programs might place a greater emphasis on training methods directed at retention and transfer, rather than mere acquisition. Findings on computer skill transfer suggest that CAI and simulation may provide effective and efficient training alternatives to actual system training. Transfer findings also emphasize that the Army's computer skills training should provide training guidance, problem solving, and error correction to improve job-based performance.

Concerns and Limitations

The major concern is that the research annotated appears disjointed, not a sustained and coherent body of work. Admittedly, the area is relatively new and that, in part, may account for this perceived status. With the exception of Gattiker (1990; 1992), the efforts annotated do not effectively build on the results of prior research, even their own.

Experimental conditions are rarely designed to clarify a theoretical issue or replicate an empirical finding. Similarly, Gattiker (1992) concludes there is little communication among researchers in related areas such as personnel psychology, cognitive psychology, ergonomics, education, and training. In sum, the need for systematic empirical and conceptual replications to understand and improve computer skills training is stressed.

Additional problems contributing to this status include inconsistent and imprecise terminology, insensitive training methods and schedules, and a general disregard for the wealth of objective performance data readily obtained from computer-mediated work. The reported difficulties with search terms substantiate that there is no consistent definition of computer skills or the supporting keywords and constructs needed to integrate disparate research and training efforts.

The research annotated also confirms Gattiker's (1992) assessment that most efforts in the area have used similar training methods and schedules, such as continuous training over a short-term period with only one training method. Finally, researchers must work harder to determine more precisely what accounts for improvements in computer skills training, what is only ancillary, and what may actually impair the retention and transfer of computer skills to job performance.

Future Directions

Many areas in the acquisition, retention, and transfer of computer skills remain to be explored. The concerns and limitations discussed previously, identify a number of directions for future research on computer skills. From a C^4I applications perspective, one important area in need of study is the long-term retention of computer skills. Many people work with computers on a daily basis or for prolonged periods of time. Vagaries in such jobs, however, often create a need to reuse an application not common to the recent daily routine. In the military, such vagaries are a norm due to frequent changes in assignments, organizations, equipment, and training agendas. Computer-skill retention is a key concern, therefore, accented by the high costs incurred when potentially life-saving computer skills are forgotten. Future research needs to examine how long it takes for computer skills to decay as well as what types of information are lost to design better computer skill training and sustainment programs.

Other suggested directions for future research are provided by Gattiker (1990). For example, more information is needed on how people learn about computers in order to better design training materials. Also, even though the learning style of students has been examined by some studies, the learning style of instructors may influence how easily students acquire the material. Finally, many individual difference variables, such as self-expectancy, amount of effort, type of software most commonly used, and related abilities such as math skills, have yet to be examined.

Research is sorely needed to examine the tradeoffs introduced by life-long training notions, computer-based performance support systems, embedded training, and just-in-time training. Additionally, research should focus on better ways to package such on-line training products to avoid mere page turners. Research should examine the application of training guidelines, as referenced, to these electronic products that may replace more conventional training programs.

Regarding past and future research, computers and the skills required to apply them are inconstant. Advances in software and hardware routinely transform computers into more intuitive and user-friendly tools. Such advances may reduce the skill requirements for applying a system. More common, industry standard interfaces (e.g., Windows-based) may increase the transfer of skills between applications and systems. A similar assessment of computer-based instruction was made by Kozma (1994, p. 11): "Much of the prior research on CBI was actually studying a different, less capable medium." Research on computer skills, therefore, should evolve to reflect current and anticipated medium capabilities.

References

Adams, J.A. (1987). Historical review and appraisal of research on the learning, retention, and transfer of human motor skills. Psychological Bulletin, 101(1), 41.74.

Anderson, R.E., & Klassen, D.L. (1981). A conceptual framework for developing computer literacy instruction. AEDS Journal, 14, 128-150.

Army Science Board. (1995). Use of technologies in education and training (Ad Hoc Study Final Report). Washington DC: Army Science Board (SARD-ASB) Research, Development and Acquisition. (AD A303 504)

Cascio, W.F. (1995). Whither industrial and organizational psychology in a changing world of work? American Psychologist, 50(11), 928-939.

Collins, J.W., & Throne, M.H. (1996). Annotated bibliography of training technologies and methods for teaching the use of advanced technology. (ARI Research Product 96-02). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Decker, G.F. (1996). The U.S. Army towards Force XXI. Military Technology, 20, 111-117. (ISSN 0722-3226)

Druckman, D. & Bjork, R.A. (Eds.). (1994). Learning, remembering, believing: Enhancing human performance. Washington, DC: National Academy Press.

Eberts, R., & Brock, J.F. (1984). Computer Applications to instruction. In F.A. Muckler (Ed.), Human factors review: 1984 (pp. 239-284). Santa Monica, CA: Human Factors Society, Inc.

Gattiker, U.E. (1990). Individual differences and acquiring computer literacy: Are women more efficient than men? In U.E. Gattiker, L. Larwood, & R.S. Stollenmaier (Eds.), End-user training. Technological innovation and human resources (Vol. 2, pp. 141-179). Berlin, Federal Republic of Germany: Walter de Gruyter.

Gattiker, U.E. (1992). Computer skills acquisition: A review and future directions for research. Journal of Management, 18(3), 547-574.

Kozma, R.B., (1994). *Will* media influence learning? Reframing the debate. Educational Technology Research and Development, 42(2), 7-19.

Kulik, J.A. (1994). Meta-analytic studies of findings on computer-based instruction. In E.L. Baker & H.F. O'Neil, Jr. (Eds.), Technology assessment in education and training (pp. 9-33). Hillsdale, NJ: Lawrence Erlbaum Associates.

Reigeluth, C.M. (1996, May-June). A new paradigm of ISD? Educational Technology, 13-20.

Rowatt, W. & Shlechter, T.M. (1993). A review and annotated bibliography of literature relevant to armor skill retention (Research Product 93-03). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A263 407)

Swezey, R.W., & Llaneras, R.E. (1997). Models in training and instruction. In G. Salvendy (Ed.), Handbook of human factors and ergonomics (2nd ed., pp. 514-577). New York: John Wiley and Sons, Inc.

U.S. Army Armor Center (1994). Advanced warfighting experiment: Operation Desert Hammer VI. Fort Knox, KY: Author. (Available from the Commander, U.S. Army Armor Center and Fort Knox, ATTN: ATZK-PTF-D, Fort Knox, KY 40121-5200).

U.S. Army Armor Center (1996). Advanced warfighting experiment focused dispatch final report, Mounted Battlespace Battle Lab. (Available from the Commander, U.S. Army Armor Center and Fort Knox, ATTN: ATZK-PTF-D, Fort Knox, KY 40121-5200).

U.S. Department of Army (1995). Force XXI Operations: A concept for the evolution of full-dimensional operations for the strategic army of the early twenty-first century. (TRADOC Pamphlet 525-5). Fort Monroe, VA: Author. (Available from the Commander, U.S. Army Training and Doctrine Command, ATTN: ATDO-P, Fort Monroe, VA 23651-5000).

Appendix A

Annotated Bibliography

Allwood, C.M., & Eliasson, M. (1987). Analogy and other sources of difficulty in novices' very first text-editing. *International Journal of Man-Machine Studies*, *27*, 1-22.

The purpose of this study was to determine why novices experience difficulties when using a word processor for the first time. The researchers evaluated 28 novices learning to use the text-editing program, WordStar. Participants performed four tasks, which were divided into 17 subtasks. Sample subtasks included starting the system, selecting a file, moving the cursor to a specified screen location, and saving a file. Participants were given a short instruction manual to read before completing the computer tasks. The number of errors was calculated for each participant by subtask. Errors were divided into three categories: a) syntactic errors (e.g., errors when inserting or taking out disks, illegal combination of letters and symbols in a command, or physical execution errors); b) semantic errors (e.g., errors when writing a file name or other writing errors); and c) inefficient commands. The best predictor of errors was priming analogies, followed by typewriter analogies. Priming analogies were defined as performing a task correctly but performing it incorrectly later. Typewriter analogies were defined as performing a task incorrectly by applying knowledge of typewriters to the task. The authors provide suggestions for reducing analogy errors to aid novices' learning and retention of computer programs.

* * * *

Baker, S., & Marshall, E. (1989). Simulators for training and the evaluation of operator performance. In L. Bainbridge & A. Ruiz Quintanilla (Eds.), *Developing skills with information technology* (pp. 293-314). Chichester: John Wiley & Sons Ltd.

The authors discussed some of the reasons for using simulators in addition to real training. For instance, some real training situations may be dangerous, it may be too expensive to use real equipment for training, and it allows for failures that may be rare in real situations to be practiced. Regular assessment of trainees is encouraged to monitor not only how much trainees have learned but also how much they have forgotten as well as the type of information forgotten. Although the authors have not conducted any empirical studies of retention, they have observed differences in the type of forgetting that takes place over training sessions. After a gap in training of approximately three months, trainees still have a good working knowledge of the general process, but they have forgotten the procedures for working with the interface. For example, they typically need some time to re-familiarize themselves with the trackball or other parts of the system. The authors suggest that systematic research needs to be conducted in the area of retention to verify their observations.

* * * *

Barnard, P.J., Hammond, N.V., MacLean, A., & Morton, J. (1982). Learning and remembering interactive commands in a text-editing task. Behaviour and Information Technology, 1(4), 347-358.

This study was concerned with how well people remember text-editing commands since the commands usually are made of already familiar words that are given slightly new meanings. Forty-eight novices were shown jumbled proverbs which they had to fix by using the text-editing commands. Half of the participants were given general commands (e.g., transfer, add, restore) while the other half were given more specific commands (e.g., fetch, insert, join). In the first session, participants were asked to correct eight proverbs with the use of 12 commands. Two weeks later, participants returned for a memory test and to correct four more proverbs. In the memory test, participants recalled as many commands as they could and gave a description. Next, they were given 36 items and asked to indicate the 12 commands they had used earlier and how confident they were of each choice. Finally, they were presented with the 12 commands and were asked to describe the purpose of as many as they could recall. Although there was no difference in the recall for command names, recall for meanings of commands was more accurate for specific commands. Also, participants accessed help information more with the general commands. The authors conclude that different command names may influence performance, especially when first learning a computer system.

* * * *

Battista, M.T., & Krockover, G.H. (1984). The effects of computer use in science and mathematics education upon the computer literacy of preservice elementary teachers. Journal of Research in Science Teaching, 21(1), 39-46.

This study addressed methods for improving computer literacy among elementary school teachers. Participants consisted of 94 teachers, each enrolled in one of four classes. However, no mention is made of their previous computer experience. Teachers enrolled in an earth science course were given four hours of computer-assisted instruction (CAI), while teachers enrolled in a course on teaching mathematics were assigned to the programming condition. The control group consisted of two other teaching mathematics classes. The CAI group took quizzes on the computer and participated in three earth science computer simulations, but they were not taught anything more about the computers. All the programs that this group used were loaded onto the computer by the instructor. The programming group was taught how to use the computers and how to program in BASIC during two 50-minute class sessions. The control group was not exposed to computers at all. Participants were given a pretest the second week, and received the posttest the fourteenth week of the 16 week semester. Both tests measured the affective and cognitive knowledge participants had about computers. Results showed no effect on computer literacy for the programming group relative to the control group. The CAI group, however, was higher than the other two groups on the affective domain of computer literacy. The authors explain that one reason for these findings could be that subjects in the CAI group did not have any demands placed upon them whereas the programming group did. Therefore, the CAI group did not experience the frustration of not being able to get the computer to do what they wanted all the time. The researchers concluded that elementary teachers may not be able to independently learn, experiment, and interact with computers.

* * * *

Booth, P.A., & Marshall, C.J. (1989). Usability in human-computer interaction. In P.A. Booth (Ed.), An introduction to human-computer interaction (pp. 103-136). Hillsdale, NJ: Lawrence Erlbaum Associates.

In this book chapter, the significance of user errors is discussed. The authors stress that errors are useful because they can lead to the cause of user difficulty and point to problems in the system. Different classifications of errors are also discussed in this chapter. For example, a distinction is made between a mistake (an error in intention), and a slip (an error in carrying out an intention). Finally, the authors present variables that are important for the usability of a system. However, they also mention that although the usability of a system is an important evaluation issue for system improvement, it cannot provide a complete understanding of the human-computer interaction process.

* * * *

Britton, R. (1995). The Navy needs to get serious about computer training. U.S. Naval Institute Proceedings, 121(8), 65.

In this opinion article, the author feels that computer training in the Navy is "...a mish-mash of programs accomplished at the local command level, with most formal Navy-wide training being done only on specific technical systems that are often antiquated and inefficient" (p. 65). Training people on general computer skills on modern equipment is hardly ever done. According to the author, osmosis is the most common method of computer training in the Navy. In order for the Navy to progress into the future, Sailors must be trained in computer skills and must use these skills to carry the Navy into the future.

* * * *

Callahan, J., Hopkins, D., Weiser, M., & Shneiderman, B. (1993). An empirical comparison of pie vs. linear menus. In B. Shneiderman (Ed.), Sparks of innovation in human-computer interaction (pp. 79-88). Norwood, NJ: Ablex Publishing Corporation.

The effect of item positioning in two types of menus was considered: linear menus, which are used in most software applications, and pie menus, which display menus in a circular format. Items in a pie menu are placed at equal radial distances along the circumference of the circle. The hypotheses were that pie menus would decrease search time and error rates and that they would be especially useful in menu applications "...suited for a circular format, diametrically opposed item sets (e.g. open/close), [or] directions (e.g. up/down)..." (p. 82) while linear menus would be more useful for sets of linear items (e.g. one, two, three). All 33 subjects were repeatedly tested on both menu types and their error rates and search times were measured. Results showed that search times for target items were consistent for pie menus while search times increased proportionally to the distance of the target items from the initial position of the cursor for linear menus. In addition, error rates were lower for pie menus than for linear menus, although this difference was only marginally significant. The authors conclude that pie menus seem promising, but more research is needed. One problem with pie menus is that only a limited number of items can be placed in the menus before the menu size becomes too large. Several recommendations are given for adaptations of the pie menu.

* * * *

Carroll, J.M., & Mack, R.L. (1983). Actively learning to use a word processor. In W. Cooper (Ed.), Cognitive aspects of skilled typewriting (pp. 259-281). New York: Springer-Verlag.

For a review, see Carroll & Mack, 1984.

* * * *

Carroll, J.M., & Mack, R.L. (1984). Learning to use a word processor: By doing, by thinking, and by knowing. In J.C. Thomas & M.L. Schneider (Eds.), Human factors in computer systems (pp. 13-51). Norwood, NJ: Ablex Publishing Corporation.

The purpose of this study was to examine the learning strategies of novices. Ten office temporaries learned a word processing system through self-study of the reference manual over the course of four half-days. After 12 hours of reviewing, the participants were asked to complete some basic computer tasks such as typing, revising, formatting, and printing. Even after 12 hours of study, none of the participants could accomplish all the tasks without serious difficulty in at least one of the tasks. Part of the participants' difficulties was due to wanting to learn by experimenting instead of reading the manual. Many novices inappropriately apply their knowledge of typewriters to computers. For example, pressing the return or enter key on a computer keyboard will insert a blank line in the text while pressing the return key on a typewriter will advance the cursor to the next line without inserting a blank line. In the word processing program that the participants used, many were confused by an inconsistency between the way the cursor moved around in a menu and the way it moved around the typing display. The authors conclude by emphasizing the importance of active learning and suggesting that computer programs need to have a more interactive approach in order to better teach word processing to novices.

* * * *

Carroll, J.M., Mack, R.L., Lewis, C.H., Grischkowsky, N.L., & Robertson, S.R. (1985). Exploring exploring a word processor. Human-Computer Interaction, 1, 283-307.

For this study, the researchers created a "guided exploration" (p. 286) instructional manual for a word processing program. Participants trained with this manual and were asked to talk aloud about what they were attempting to do and what kinds of problems they were encountering. The instructional manual consisted of a set of 25 cards, each of which covered a basic (e.g., text entry) or an advanced (e.g., justifying the right margin) skill. The information presented was intentionally incomplete in order to allow for exploration by the participants. No step-by-step instructions were presented. Two groups of six participants each were used; one group used the guided exploration technique while the other used the commercial word processing manual. Both groups consisted of novices who were hired from a temporary employment agency. Participants in the commercial manual group were given four half-days to train, while participants in the guided exploration group were only given half the amount of time (two half-days).

After the training period, all participants were given a transfer task of entering a one-page letter and printing it. Afterwards, they were given an edited version of the letter and were asked to revise it and print out the revised version. Participants in the guided exploration group spent less time on the transfer task than the commercial manual group. Also, the guided exploration group was able to accomplish more of the transfer tasks. Finally, the guided exploration group spent more time exploring the capabilities of the program than the other group. The authors conclude that this finding provides support for their assumption that exploratory behavior can be encouraged in the right kind of learning environment.

* * * *

Catrambone, R. (1990). Specific versus general procedures in instructions. Human-Computer Interaction, 5, 49-93.

The author begins by discussing the problem of instructional specificity for the novice computer user. Novices need instructions that are specific so they can perform the tasks correctly, but this specificity leads to poor transfer to novel tasks. Novices, who were paid for their participation, were given either general or specific instructions on how to delete and format text on a word processor. It was hypothesized that those who received general deletion procedures would outperform those who received specific deletion procedures on a transfer task. However, the formatting groups should perform equally well on the transfer task since all the tasks are similar enough that novices can generalize from specific cases. The 56 subjects were randomly assigned to either the general or specific group and either the deletion or formatting group. The experiment lasted one hour and consisted of training and transfer phases. During training, subjects practiced various deletion or formatting tasks on three documents. During testing, subjects attempted deletion or formatting tasks that were different from those learned during training on four new documents. In the deletion group, 13 of the 14 in the specific instruction group needed assistance on the transfer tasks whereas none in the general instruction group needed assistance. In the formatting group, an equal number in both the specific and general instruction groups required assistance with the transfer tasks. In addition, in both the deletion and formatting groups, subjects in the specific instruction group completed the initial training tasks more quickly than subjects in the general instruction group.

In the second experiment, the instructions for the specific instruction deletion group were changed to examine whether there was a way in which to alter the instructions so that they would still be specific yet lead to generalization. Subjects were 15 female undergraduate students with little or no computer experience who were all in the deletion group with either general or specific deletion procedures. Once again, those in the specific deletion group outperformed the general deletion group on the initial deletion tasks. However, there was no difference on the transfer tasks. In other words, the specific instructions allowed the subjects to learn the basic tasks and yet generalize to novel tasks as well. The authors conclude that it is useful to combine "...features of general and specific instructions and mental model information. Although specific instructions may help the novice get started more quickly, additional general instructions or mental model information could help them when they get to novel situations..." (p. 88).

* * * *

Charney, D., Reder, L., & Kusbit, G.W. (1990). Goal setting and procedure selection in acquiring computer skills: A comparison of tutorials, problem solving, and learner exploration. Cognition and Instruction, 7(4), 323-342.

These authors state that initial skill learning consists of three components: learning the purpose of a new procedure, learning the conditions under which the new procedure can be applied, and learning how to use the procedure. This study explored the differences between learner initiated and instructionally provided goals with the use of three conditions: tutorials, problem solving, and exploration in learning to use an electronic spreadsheet. The dependent variables included training time, percentage of responses correct at test, and mean time to a correct solution at test. The testing session took place two days after the training session. The results showed that: a) the exploration group took less time than the problem solving group for training; b) the problem solving group solved more test items than the tutorial group and the exploration group; and c) the problem solving group solved test problems faster than the tutorial group. The authors summarize that even though training with problem solving took longer, it produced faster and more successful performance. The reason that the exploration group did poorly might be due to the subjects' minimal knowledge of electronic spreadsheets. The authors conclude that for procedural skills, solving problems given by the instructor is more effective training than independent goal setting if the learners have little knowledge of the skill domain.

* * * *

Christoffersen, K., Hunter, C.N., & Vicente, K.J. (1996). A longitudinal study of the effects of ecological interface design on skill acquisition. Human Factors, 38(3), 523-541.

This study examined the effect of interface design on acquisition of computer skills over a six month period. The interface contained either physical and functional representations that "...describe the state of the functions that the physical components are intended to achieve..." (pp. 526-527) or only the physical representation of the system. The participants were six men, who were paid for their participation. Participants controlled the Dual Reservoir System Simulation II (DURESS II) for one hour five days a week for six months. The DURESS II is a simulation system that can supply water to either, both, or neither of two reservoirs. The goal is to keep each of the reservoirs at a specific temperature and to meet the current water output demand rates by controlling eight valves, two pumps, and two heaters.

During the introductory session, participants in both groups were given a description of DURESS II and an explanation of the interface they would be using. During the next phase, the introductory practice phase, participants were introduced to the tasks that they would be performing over a one month period. The extended practice phase lasted approximately four months. During this phase, participants performed standard trials repeatedly. The standard trials included a start-up task, followed by a tuning task, and finally a shutdown task. After this phase, there were six transfer trials, during which the participants switched interfaces. Analyses revealed that the group receiving both physical and functional information was more consistent and had less variance than the group receiving only physical information. The authors conclude that the interface providing both physical and functional information has a clear advantage over the more traditional type of interface. They then discuss some of the companies that are now beginning to study the use of ecological interface designs.

* * * *

Cicchinelli, L.F., Harmon, K.R., Keller, R.A., & Kottenstette, J.P. (1980). Relative cost and training effectiveness of the 6883 three-dimensional simulator and actual equipment. AFHRL-TR-80-24. Texas: Air Force Human Research Laboratory, Brooks Air Force Base.

The effectiveness of a three-dimensional simulator for training aviation maintenance personnel was compared to use of the actual equipment. It was hypothesized that those who trained on the simulator would have equal performance, trouble-shooting skills, and proficiency when tested on the actual equipment as those who had trained with the actual equipment. A sample of 115 trainees were randomly placed in one of four conditions: simulator training-simulator testing, simulator training-actual equipment testing, actual equipment training-simulator testing, or actual equipment training-actual equipment testing. As expected, no difference was found on performance measures between simulator training and actual equipment training. The authors conclude that simulator training for aviation maintenance is preferable because the skills transfer to the actual equipment and it is more cost effective. However, although the authors do not discuss this, the low number of subjects in each group could have led to the finding of no differences.

* * * *

Czaja, S.J., Hammond, K., Blascovich, J., & Swede, H. (1986). Learning to use a word-processing system as a function of training strategy. Behaviour and Information Technology, 5(3), 203-216.

In this study, 135 women were trained on a word processing system using one of three techniques: instructor, reference manual, or computer tutorial. These methods were evaluated in order to determine if alternative training strategies are needed to teach people to use word processing systems. After their eight hour training session, subjects were asked to complete five tasks.

These tasks included: a one page formatted memo, a short business letter, a printed copy of a hand revised report, an address list, and a copy of a 13-page newspaper story. Analyses were conducted on the number of tasks attempted, the number of tasks completed, the time to complete each task, and the number of errors on each task. For the number of tasks attempted, subjects in the computer-based training groups attempted fewer tasks, completed fewer tasks, and made more errors than subjects in the other two groups. The authors conclude that training strategy has an effect on learning to use a word processor. They suggest that training should attempt to incorporate more active participation of the learner through the use of problem solving exercises. More work needs to be conducted in developing new training strategies which are effective in teaching novices word processing. Although the instructor-based and reference manual-based methods were superior to the computer-based training method, subjects made many errors in those conditions as well.

* * * *

Czaja, S.J., Hammond, K., Blascovich, J.J., & Swede, H. (1989). Age related differences in learning to use a text-editing system. Behaviour and Information Technology, 8(4), 309-319.

This experiment examined age related differences in learning to use a word processing program using data collected by Czaja et al. (1986), with age as another independent variable. The age ranges were: young (25-39 years), middle (40-54 years), and old (55-70 years). Younger subjects were more successful at learning word processing tasks than older subjects. In addition, older subjects in the computer-based training condition made more errors than any of the other subjects. Finally, the amount of time spent per task increased with age while the number of tasks attempted and completed decreased with age across all conditions. Overall, the results indicate that older people have a harder time learning word processing. In order to help older people achieve at the same level as younger people, it is necessary to understand the sources of age differences in learning computer tasks and find ways to reduce these differences.

* * * *

Davies, S.P., Lambert, A.J., & Findlay, J.M. (1989). The effects of the availability of menu information during command learning in a word processing application. Behaviour and Information Technology, 8(2), 135-144.

In this experiment, the availability of menus was manipulated during the learning stage of basic word processing commands to find out how subjects recall the commands. With menus present, the need to memorize all the commands is no longer necessary. However, subjects still have to remember the purpose of the displayed commands. Even though the presence of menus may help reduce the memory load, it does have some disadvantages, especially for more skilled users. For example, experienced keyboard users may find it time consuming to take a hand off the keyboard to manipulate a pointing device. Similarly, when menus are permanently visible, they make the screen look cluttered. Because of such factors, permanently visible menus may be useful for novices but a nuisance for experts.

Thirty-six subjects were randomly placed in one of four groups: mouse selection, menu present at practice and test; keyboard entry, menu present at practice and test; keyboard entry, menu absent at practice and test; and keyboard entry, menu present at practice and absent at test. Subjects were trained on the word processing program Microsoft Word™. After the 40 minute tutorial session, subjects were given 3 minutes to review and study the commands they had been taught. Afterwards, they were tested on the same commands. Subjects performed more quickly and accurately in the condition where menus were absent both during practice and test time. No differences were found between the performance of subjects using a mouse and subjects using a keyboard. From the results, it appears that a 40 minute tutorial was enough to learn the basic commands so that permanent menus were no longer necessary. When the menu was always absent, it encouraged active learning of the commands which led to more efficient performance at test time. However, menus are still useful for less frequently used commands.

* * * *

Davis, S., & Bostrom, R. (1992). An experimental investigation of the roles of the computer interface and individual characteristics in the learning of computer systems. International Journal of Human-Computer Interaction, 4(2), 143-172.

Learning performance and attitudes toward computers were measured using two computer interface systems: a command-based interface (DOS), and a direct manipulation interface (Apple Macintosh). Learning was measured by the number of correct responses on a set of computer-related tasks and scores on a comprehension test. Attitudes were measured by a questionnaire evaluating perception of ease of use of the system. Hypotheses included: the direct manipulation group will perform better and find the system easier to use than the command-based group; concrete learners will perform better using direct manipulation whereas abstract learners will perform better using commands; and high visuals will perform better and find the system easier to use than low visuals. Learning tasks were 12 items which included making directories and copying files. Eighty computer novices participated in the study. Participants were first given a series of learning style tests and were then given 60 minutes to work through exercises in a training manual. Participants received either instruction-based training (gave participants no control over how they learned the material) or exploration training (allowed participants to experiment with the system).

After the training period, participants were given 30 minutes to complete a set of computer tasks.

Participants in the direct manipulation group outperformed the command-based group, high visuals outperformed low visuals, and high visuals found the system easier to use than low visuals. In conclusion, direct manipulation systems are superior to command-based systems for novices. In order to make systems easier for low visuals, designers should include "undo" commands and incorporate more intelligent help systems.

* * * *

Deck, J.G., & Sebrechts, M.M. (1984). Variations on active learning. Behavior Research Methods, Instruments, & Computers, 16(2), 238-241.

The goal of this article was to provide a more precise understanding of the role of active learning in the acquisition of computer skills. Results from previous studies have found that even simple computer tasks can be very complicated for novices. For this study, the researchers examined in detail the way in which four participants learned to use a new computer system. Participants were presented with the manuals and told that they could work at their own pace with the goal of becoming experts at using the system. They were asked to read aloud and verbalize their thoughts. In addition, they were to complete four tasks: a) writing and running a Pascal program, b) creating a memo, c) reorganizing data in a document, and d) revising a long document. Finally, a few weeks after the training, the participants were asked to describe how the system worked in either words or diagrams. It was determined that participants differed in the extent to which they used either procedural or relational information. The procedural participants were more likely to focus on the specific details of each operation, while relational participants were more likely to focus on the global organizational properties of the system. The authors conclude that some people are externally driven, and might benefit from manuals and tutorials. Others are internally driven, and might not use manuals as much as rely on trial and error techniques. Therefore, instructors should first determine what kind of material each individual would benefit from most. Some people may need step-by-step tutorials while others might just need a reference manual.

* * * *

Dray, S.M., Ogden, W.G., & Vestewig, R.E. (1981). Measuring performance with a menu-selection human-computer interface. Proceedings of the 25th annual meeting of the human factors society (pp. 746-748). Rochester, New York: The Human Factors Society, Inc.

This study compared the use of computer menus which present all the information on one screen to menus calling sub-menus. Ten employees at Honeywell volunteered to participate. Although the computer background of the participants was not mentioned, it appears they were all novices. The participants were to select a target item from the appropriate menu for each trial. Response times and errors were recorded. All participants received 138 experimental trials in each condition. Novices are faster when all the information is presented at once, but as they acquired more experience, menus calling sub-menus became the better format. The authors suggest that rarely used functions should be presented in large menus that do not require memorization of location.

* * * *

Dyck, J.L. (1995). Problem solving by novice Macintosh users: The effects of animated, self-paced written, and no instruction. Journal of Educational Computing Research, 12(1), 29-49.

Previous studies have found that novice users of the Macintosh computer encounter problems when learning to first use the computer. This experiment randomly placed 48 subjects in one of four instructional conditions (no instruction, partial written instruction, complete written instruction, or complete animated instruction) to identify how much and what type of instruction is necessary in order to acquire a basic understanding of the Macintosh computer. It was hypothesized that subjects who received the animated instruction would perform faster and more accurately than all other subjects since the animated instruction combined the written instructions with animated demonstrations. After completing their tutorial, subjects were tested on mouse skills, file manipulation skills, and transfer tasks. The results showed that subjects in both complete instruction groups finished more tasks than subjects in the partial instruction and no instruction groups. However, there were no differences between the two complete instruction groups (written and animated). In summary, subjects in the two complete instruction groups acquired a more basic understanding of the computer system. The author did not discuss the implications of the results as far as which type of tutorial would be best to use: written or animated.

* * * *

Eason, K. (1989). Meeting the information technology learning needs of professional office workers. In L. Bainbridge & A. Ruiz Quintanilla (Eds.), Developing skills with information technology (pp. 227-240). Chichester: John Wiley & Sons Ltd.

In this article, the author argues that the current methods of teaching novices computer skills are not successful and new teaching methods need to be developed so that computers will be used to their full potential. Since most jobs only require intermittent use of the computer, computer skills are presently acquired at intervals. With this type of learning, the user may not develop an understanding of the entire system. If each step is learned in isolation and stored in memory separately, details will be easily forgotten from one session to the next. In addition, when users come upon a new situation, they may feel uncomfortable exploring the system when they do not have a basic understanding of the way the computer works. A better approach to teaching computer skills would be to give the user a small number of general principles on which the computer operates consistently and would be easy to understand. With this method of teaching, the user would feel more comfortable exploring in new situations and could even develop a general understanding of the entire system.

* * * *

Egan, D.E., & Gomez, L.M. (1985). Assaying, isolating, and accommodating individual differences in learning a complex skill. In R.F. Dillon (Ed.), Individual Differences in Cognition (Vol. 2, pp. 173-217). New York: Academic Press.

The goal of this series of studies was to find a way to teach people word processing that may be part of a job requirement. The authors feel that different occupations require different skills and the backgrounds of the people in those occupations may also vary. Therefore, it is better to conduct studies using people in a wide variety of occupations rather than people in only one educational level, age group, or occupation. In one of the studies, five people were interviewed about potential distinctions between successful and unsuccessful learners of computer text editing. Potentially successful word processing learners were described as more confident, motivated, and willing to try things on their own than potentially unsuccessful learners. In another study (for a more complete review of this study, see Gomez, Egan, & Bowers, 1986), 33 participants were taught to use a word processor and were then asked to complete some basic tasks. The results showed a large amount in variability between participants for the number of errors. Most of this variability, however, could be predicted by participants' spatial memory and age. In a third study, 41 participants worked through a tutorial and came back a week later to work through the same tutorial for a second time. This study found that participants learned from their first tutorial and remembered enough to improve their performance a week later. Again, age and spatial memory accounted for the greatest amount of variability. In general, more research needs to be conducted to find better training and retention strategies for teaching computer skills to older people and people with poor spatial memory.

* * * *

Elias, P.K., Elias, M.F., Robbins, M.A., & Gage, P. (1987). Acquisition of word-processing skills by younger, middle-age, and older adults. Psychology and Aging, 2(4), 340-348.

The goal of this study was to examine the effects of age on performance in a word processing training program. Participants were 45 females, with 15 in each age group: younger (18-28), middle (37-48), and older (55-67). They were taught a basic word processing program that came with an audio and textbook training program. Training and evaluation consisted of seven sessions, each lasting 3 1/2 hours. The sessions were conducted over a two-week period. The first session was devoted to filling out questionnaires and taking tests of typing and verbal ability. The next four sessions were spent on the training of basic word processing procedures. During the last two sessions, participants were required to input, edit, print, and re-edit and print a 7-page document with the use of the training manual. Overall, the older group took longer to complete training and evaluation than the younger and middle groups. The older group also performed more poorly than the other two groups on tasks such as setting margins and moving a block of text. Finally, the older group required more assistance getting through the training program than the other two groups. The authors conclude that older adults have difficulty "...retaining information when the procedure produced very fast, extensive screen changes or when... 'what you see is not what you get'" (p. 346). Several recommendations are given for making training programs simpler for older adults, such as providing take-home materials and keeping training sessions short.

* * * *

Finnegan, J.P. (1977). Evaluation of the transfer and cost effectiveness of a complex computer-assisted flight procedures trainer. Technical Report ARL-77-7/AFOSR-77-6. Illinois: Institute of Aviation.

The author's purpose was to examine the amount of simulator fidelity needed for high transfer effectiveness. The effectiveness of CAI in teaching the cognitive aspects of flying holding patterns was compared to the effectiveness of using the actual aircraft or the ground-based simulator. Advertisements were placed in campus newspapers for free instruction and flight time in return for participating in the experiment, and 48 subjects were chosen. Training in the aircraft was used as the control group. Subjects were also divided by skill level (high and low) on the basis of "...ability to maintain aircraft control and execute basic flight maneuvers" (p. 26). After the preliminary flight to assess skill level, subjects in the aircraft and simulator conditions participated in a ground school class to learn the basic aspects of flying holding patterns, while CAI subjects learned the basic aspects by reading a booklet and training on the computer system. Overall, subjects in the CAI and simulator training groups made fewer errors and achieved criterion in fewer trials than those trained in the aircraft. The CAI and simulator training groups saved 25% of the total aircraft time that the aircraft group needed to reach criterion. The use of CAI is considered positive for larger groups since it is cheaper than simulator training yet just as effective, even with less fidelity.

* * * *

Frese, M., & Altman, A. (1989). The treatment of errors in learning and training. In L. Bainbridge & A. Ruiz Quintanilla (Eds.), Developing skills with information technology (pp. 65-86). Chichester: John Wiley & Sons Ltd.

The authors advocate teaching novices how to deal with errors during training rather than avoiding errors. There are two basic types of errors: slips, which occur due to wrong plans but right intentions (accidentally deleting a file), and mistakes, which occur due to wrong intentions but right plans (purposely deleting a file and then realizing that it is needed later). A classification of errors is developed with four levels of regulation in which actions regulated by lower levels are mostly automatic, while those at higher levels require conscious attention. The lowest level is the sensorimotor level, where actions are very automatic with no conscious attention. Slips are most common at this level. Novices begin learning word processing at the third level where "...complex analyses of situations and of problems are regulated" (p.70). Since regulation occurs at a higher level for the novice, they are more likely to make mistakes, whereas experts are more likely to have slips.

In a study using 24 novices, the researchers observed learning and rate of errors for two word processing packages--WordStar and MacWrite. The participants were given a manual and attempted to solve specific problems on their own. At first, participants had problems with the return key and spacebar on both systems since they were using a typewriter metaphor and expected these keys to function the same way they do on a typewriter. After these basic problems were overcome, the participants using WordStar developed other errors including forgetting and mixing up the commands. In addition, error feedback was not clear for novices and could not be understood. In contrast, novices using MacWrite did not encounter these types of problems. Instead, they made errors when the correct procedure was easier than they expected, and instructions were taken too literally. For example, the instructions told participants to select an object by pointing the cursor at it. Novices would place the cursor next to the object and point at it instead of placing the cursor on top of the object. Finally, errors were made because some objects were designed inconsistently from their everyday use. For example, rulers are used to set tabs, margins, and spacing, which is not consistent with what actual rulers are used for in real life. Although no conclusions are made regarding either system, the authors acknowledge that through their qualitative analysis, it appeared that participants in the WordStar group made approximately twice as many errors as participants in the MacWrite group.

* * * *

Frese, M., Brodbeck, F., Heinbokel, T., Mooser, C., Schleiffenbaum, E, & Thiemann, P. (1991). Errors in training computer skills: On the positive function of errors. Human-Computer Interaction, 6(1), 77-93.

Errors are a common problem when working with computers, and even experts spend time dealing with errors. The authors feel that most trainers want to reduce the number of errors that learners make by preventing the errors from occurring. However, the authors argue that there are four positive roles that errors have in training. First, people expand their mental models when they make errors. Errors indicate that the person may not know a particular area of the system well enough. Second, when errors are identified, the learner will know where the problem areas are and can avoid them, take extra care, or recognize errors in those areas more quickly. Third, when error free training is used, the trainer will restrict the learning strategies used by the learners. Finally, errors occur in the workplace, so it is better to learn about them during training when support is present.

The hypothesis of this experiment was that subjects receiving error training would perform better on a performance test than subjects who were not allowed to make errors. Twenty-four subjects went through six hours of training on a word processing program, followed by two hours of testing. The training for the error avoidant group consisted of detailed instructions for the learners to follow. If anyone in this group made an error, the experimenter would quickly intervene and correct the error with no further explanation. The error training group, however, was not given detailed instructions and they were encouraged to solve their errors on their own. The testing was broken down into four sections: free recall, where subjects were asked to recall all the commands they knew and what they could be used for; competence, where subjects were presented with an incorrect text on the screen and were asked to make the necessary corrections; transfer competence, where subjects had to solve three problems that they were not taught during training; and overall performance, where subjects were given an overall rating of how well they could use the commands they had been taught. The results showed that the error training group was better in free recall and higher in competence.

* * * *

Gattiker, U.E. (1990). Individual differences and acquiring computer literacy: Are women more efficient than men? In U.E. Gattiker, L. Larwood, R.S. Stollenmaier (Eds.), End-user training. Technological innovation and human resources (Vol. 2, pp. 141-179). Berlin, Federal Republic of Germany: Walter de Gruyter.

This article examined gender differences in acquiring computer skills and contradicted previous research in three ways. First, this experiment found that women tend to perform computer tasks better than men. Second, previously acquired technical skills do not always improve computer performance. Third, short term computer courses may be deleterious to the acquisition of computer skills. Four experiments were carried out to answer specific questions about acquisition of computer skills. In Experiment 1, it was found that students' past grade point average scores were significant predictors of performance in a computer course. It was also found that a previous computer course was beneficial to the performance of low and average ability students. In Experiment 2, it was found that the performance of high ability students (both men and women) in a course on computer literacy for the office setting was not affected by having previously attended a computer science course. However, the performance of average ability students was positively affected from a previous computer science course. Finally, the men with below average ability benefited from a previous course only in the lab portion, while women with below average ability benefited in all portions of the course. In addition, it was found that if both genders had previously attended a computer science course, men would have to invest an extra six hours to equal the performance of the women. Without previous experience, however, women would have to invest an extra 21 hours to equal the performance of the men.

In Experiment 3, it was found that previously attending a computer course did not benefit females with lower ability at all. Lower ability female students would receive greater benefit from practicing newly acquired skills than by attending a computer science course. Men, on the other hand, benefited from a previous computer science course. The final study, Experiment 4, found that lower and average ability men benefited from a previous computer course when they were put under time constraints. However, only average ability females benefited from a previous computer course when they were put under time constraints. Overall, it would seem that practice of newly acquired skills would be more beneficial to most people rather than taking a computer science course.

* * * *

Gattiker, U.E. (1992). Computer skills acquisition: A review and future directions for research. Journal of Management, 18(3), 547-574.

The purpose of this article was to review research in the area of acquisition of computer skills. The author discusses several individual difference factors that may influence learning and retention such as age, gender, general ability, psychomotor ability, motivation, and expectancy. In addition, training transfer may allow those with previous computer skills to perform at a higher level than novices. These factors, along with the person's required job skills, should determine the type of training that would be most beneficial. Current training methods include: computer-aided learning, behavioral modeling, peer training, self-study, goal setting, or combinations of the above methods. Future research should assess the effects of perceptual speed ability and psychomotor ability as well as identify the level of complexity of the skills that are acquired during computer training.

* * * *

Gattiker, U.E., & Paulson, D. (1987). The quest of effective teaching methods: Achieving computer literacy for end-users. Information Systems and Operational Research (INFOR), 25(3), 256-272.

These researchers were interested in examining training methods for making students more computer literate. Current training methods include computer-aided learning, lectures, and self-study. The authors believe that a wider range of methods need to be evaluated and used for training. Students with little or no previous computer experience enrolled in a BASIC programming course participated in the study. Their performance throughout the semester on tests and homework assignments was used as an evaluation of effective training. The researchers found that previous experience in a computer science course was helpful. Traditional classroom lectures worked better for above average students than for average students. In addition, homework seemed to be helpful for average students more than for above average students. Finally, the results show that there is no best way to train students and that different teaching methods are better for different groups.

* * * *

Gerhardt-Powals, J. (1996). Cognitive engineering principles for enhancing human-computer performance. International Journal on Human-Computer Interaction, 8(2), 189-211.

In this experiment, the researchers extracted a set of cognitive design principles and evaluated the performance of an interface built on these principles. It was hypothesized that a cognitively-engineered interface would outperform other interfaces in performance, satisfaction, and workload. The 10 cognitive principles used were: a) automate unwanted workload; b) reduce uncertainty; c) fuse data; d) present new information with meaningful aids to interpretation; e) use names that are conceptually related to function; f) group data in consistently, meaningful ways; g) limit data driven tasks; h) include in the displays only that information needed by the operator at a given time; i) provide multiple coding of data; and j) practice judicious redundancy.

The interfaces were designed for an antisubmarine warfare system. The baseline interface had features typical of a military workstation display, with reliance on alphanumerics and redundancy of information. The alternate interface was designed without the use of the cognitive principles mentioned earlier, but with the intent of improving the baseline interface. Finally, the cognitive interface was designed with the use of the cognitive principles. Subjects were 24 university students who had a minimum level of computer experience. The experiment was conducted over three sessions, each one week apart, lasting 45 minutes. Subjects worked with a different interface each session. Each session started with an overview of the interface being used and the instructions on how to perform the system tasks. Then subjects were given practice trials followed by nine actual trials. It was found that the cognitive interface was faster than both of the other interfaces, more accurate than the baseline interface, and preferred by 20 of the subjects over the other interfaces. In addition, the cognitive interface had less workload than the other interfaces. According to the authors, future work is necessary to identify a set of "critical cognitive-design principles" (p. 210).

* * * *

Gist, M., Rosen, B., & Schwoerer, C. (1988). The influence of training method and trainee age on the acquisition of computer skills. <u>Personnel Psychology</u>, <u>41</u>(2), 255-265.

These researchers were interested in the importance of age and training method on the acquisition of computer software skills. Two training techniques were evaluated: interactive tutorials and videotape presentations. It was hypothesized that participants involved in the videotape training would outperform participants in the tutorial training, that younger participants would outperform older participants, and that older participants would benefit more from videotape instruction than younger participants. After receiving a maximum of three hours of training, participants were asked to perform specific tasks which were used as measures of performance. It was found that the first two hypotheses were supported, but the videotape advantage also extended to younger participants. The researchers argue that although the costs of making videotape presentations are higher than the cost for tutorials, the increased performance may justify their use. However, more research is needed to identify why older people have more difficulty learning computer skills and how these difficulties can be reduced.

* * * *

Gist, M.E., Schwoerer, C., & Rosen, B. (1989). Effects of alternative training methods on self-efficacy and performance in computer software training. Journal of Applied Psychology, 74(6), 884-891.

The effectiveness of behavioral modeling was compared to tutorial training in teaching managers to use computer software. Six hypotheses were tested, some of which included: a) trainees in behavioral modeling will outperform trainees in tutorial training on a test of software mastery; and b) trainees with low computer self-efficacy will perform better in the modeling than in the tutorial condition, while those with high computer self-efficacy will not exhibit a difference in performance. Subjects were 108 managers and administrators from a state university who volunteered to enroll in a three hour training course to learn to use a popular computer software package.

In the behavioral modeling condition, trainees watched a model in a video perform specific steps necessary to complete each task. After each step, the video was stopped and the trainees were given time to execute that step. In the tutorial condition, subjects learned the same concepts by using an interactive computer diskette. However, subjects in this condition were told what to do rather than told and then shown. At the end of training, all subjects were asked to complete timed, objective measures of their performance. From the performance measures, it was clear that trainees in the behavioral modeling condition outperformed trainees in the tutorial training. However, computer self-efficacy was not a good predictor of performance. Overall, behavioral modeling appears to be more effective, but more studies are needed to examine the effects of type of training on the transfer of computer skills.

* * * *

Goettl, B.P. (1993). Analysis of skill on a flight simulator: Implications for training. Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting, pp. 1257-1261.

In this experiment, subjects were tested on various skills necessary for a flight simulator. Subjects were either first trained on the simulator and then tested on the component skills (backward transfer) or first taught the component skills and then trained on the flight simulator (forward transfer). The goal was to find the skills that are most important for accurately completing the simulation exercise so that these skills could be emphasized in future training. The testing took place over three days. The backward transfer group practiced on the whole simulation task the first day while the forward transfer group worked on the component skills. The second day, all subjects worked on the whole task. The third day, all subjects were tested on the component skills. Based on their performance on the second day, the subjects in the two groups were divided into high and low ability subgroups. Comparison of the high and low ability groups showed that type of learning made a difference. Although high ability subjects in both groups did equally well, low ability subjects in the backward transfer group outperformed the low ability subjects in the forward transfer group on most of the component skills. An unexpected finding was that there were more males than females in the high ability groups. Therefore, more research needs to be conducted to analyze gender differences in computer simulation tasks.

* * * *

Gomez, L.M., Egan, D.E., & Bowers, C. (1986). Learning to use a text editor: Some learner characteristics that predict success. Human-Computer Interaction, 2(1), 1-23.

The goal of this study was to understand why some people have more difficulty learning computer skills than others. Although this is the same as one of the studies presented in Egan and Gomez (1985), it will be discussed in greater detail. Participants were 33 women who knew how to type but had little or no computer experience. They were taught to use three commands on the text editor, ED. Teaching consisted of reading the instructional manual and then working the exercises after each section of the manual. The commands they learned to use were: "append," which is used to add text; "delete," which is used to remove text; and "substitute," which is used to exchange text for another. Participants were given 2 hours to work through the tutorial at their own pace and their performance on the exercises was measured. It was found that age and spatial memory accounted for most of the variance in performance.

In a second study (which is the same as one of the studies presented in Egan & Gomez, 1985), 41 women with the same background as those in the first study learned the same tasks. However, they either learned the standard command names or randomly chosen command names. Participants were given 3 hours to work through the tutorial. They then returned a week later and performed the same exercises and eight new ones. At this time, they were also asked to recall the name and definition of each command they had previously learned. In addition, they were given a series of texts, each requiring a single edit. They were asked to write down exactly what they would have to type to make the editing change. Those in the standard commands group were able to recall more of the command names, but not definitions or how to solve the editing problems. Again, age and spatial memory accounted for most of the variance in performance within groups.

As part of a follow-up, 25 participants from the second experiment returned after four months and received a memory test. After four months, there were no group differences in the number of command names or definitions recalled. Both groups had considerably lower rates of recall than initially.

* * * *

Harrison, C., Hay, D., Pierson, A., & Burton, J.K. (1992). Computer literacy skills among school students and employees in industry. Journal of Computer Assisted Learning, 8(4), 194-205.

Failure to understand or correctly use the information contained in a computer manual can have serious consequences, such as the loss of production in industry, or vulnerable defense systems in the military. The authors were looking for differences between students about to graduate, and employees already in the workforce as well as gender differences in the learning of computer tasks. The results show that although some differences were not significant, males outperformed females on all the tasks. In addition, the school students outperformed the employees on all the tasks. The article concludes by suggesting that although females did not perform as well as males on the computer tasks, there are indications that when females have as much familiarity with computers as males, they may actually perform at a higher level.

* * * *

Jones, M.B. (1989). Individual differences in skill retention. American Journal of Psychology, 102(2), 183-196.

This article consisted of two experiments that looked at skill retention in playing video games after periods with no practice. In the first experiment, 27 males, who were paid for their participation, practiced six video games over a period of 15 days and were tested 4 to 18 months later. In the second experiment, 159 college students, who were paid for their participation, practiced one video game in five sessions and were tested four months later. The results from both studies imply that when subjects are given a set amount of time for practice, those who learn rapidly tend to overpractice. This overpractice leads to better retention.

* * * *

Kagan, D.M., & Esquerra, R.L. (1984). Personality and learning BASIC. Journal of Instructional Psychology, 11(1), 10-16.

This study was aimed at answering two questions: whether introverted students are better at learning computer programming, and if they are, whether this higher performance remains or diminishes over time. The researchers examined students who were either majors or non-majors in computer programming in an introductory BASIC course. Performance in the course was evaluated by three separate exams. The only correlation was that those with a strong need for uniqueness obtained higher scores on the second exam than others. However, by the end of the course, personality variables had no relation to the learning of computer programming. Finally, there was no difference in learning BASIC between computer programming majors and non-majors.

* * * *

Kay, R. (1990, Summer). The relation between locus of control and computer literacy. Journal of Research on Computing in Education, 22, 464-474.

When it comes to computers, people have either an external or an internal locus of control. People who have an external locus of control for computers believe that they have no control over what the computer accomplishes. People who have an internal locus of control for computers believe that interaction with computers is determined by personal actions. The goal of this article was to determine whether a relationship exists between locus of control and computer literacy. Participants received a questionnaire to determine computer locus of control, and then were tested on four subscales of computer literacy: basic skills, software skills, computer awareness, and programming. The results showed that internal locus of control was positively correlated with computer literacy. However, no suggestions are presented for how to help people with an external locus of control become more advanced or change their attitudes toward computers.

* * * *

Keeler, C.M., & Anson, R. (1995). An assessment of cooperative learning used for basic computer skills instruction in the college classroom. Journal of Educational Computing Research, 12(4), 379-393.

This study used students enrolled in a computer course and analyzed their acquisition of computer skills. Students were randomly placed in either a traditional classroom teaching format, or in a cooperative teams format. Previous studies have found that the use of cooperative teams results in better problem-solving, higher scores on quizzes and tests, and even higher grades. However, this type of research had never been conducted on computer skills. It was hypothesized that students in the cooperative learning section would get better grades, that fewer would drop out, and that those with computer anxiety would especially perform better in this type of environment. Overall, the results favored the cooperative learning approach, which improved performance and reduced drop out rate.

* * * *

Kennedy, T.C.S. (1975). Some behavioural factors affecting the training of naive users of an interactive computer system. International Journal of Man-Machine Studies, 7, 817-834.

The basis of this study was to teach novice computer users how to operate a system that would help hospital staff keep track of patient records. The attitudes of "...a large sample..." (p. 817) of hospital clerical staff were assessed and participants were split into two groups: those who were opposed or indifferent to computers, and those who were favorable to computers. These two groups were further split with half of each group receiving training with either reference manuals or the system itself. Finally, these four groups were tested under one of two conditions: half were given assistance during the tests and the other half had to use the system and manual to get help. The tests were composed of three to five sessions, each test lasting approximately 15 minutes. Sessions consisted of various tasks including data entry and record and file manipulation. A posttest was administered to track the hospital staff's progress. However, no analyses are presented, as the authors feel "it is beyond the scope of this paper..." (p. 825). The authors found no difference in performance between participants who read the manual and those who worked with the system, or participants who had a negative attitude and those who had a positive attitude. The authors conclude that their sample size may have been too small to find any differences, however no mention is made of the sample size used.

* * * *

Kiger, J. (1984). The depth/breadth trade-off in the design of menu-driven user interfaces. International Journal of Man-Machine Studies, 20, 201-213.

Performance of novices using either broad and shallow or narrow and deep menus was evaluated. The breadth of the menu structure refers to the number of items present on any menu in the tree, while the depth of the menu structure refers to the number of levels that exist. Five menu structures were compared. Each subject was asked to locate target items in each of the five menus. It was found that the narrowest and deepest menu structure (two choices at each menu with six menus) was the slowest, least accurate, least preferred and most difficult to use. The best arrangement was found to be somewhere in the middle (eight choices at each menu with two menus), but closer to the broad and shallow side. The author concludes that whenever possible, menu breadth should be maximized as to avoid more than two or three levels.

* * * *

Kozma, R.B. (1991). Learning with media. Review of Educational Research, 61(2), 179-211.

This article reviews research on learning with media, including books, television, computers, and multimedia. The review suggests that learning is an active and constructive process and that defining media capabilities directly influence learning activities and knowledge construction. A number of unique training capabilities afforded by computers are identified, and then theoretical and empirical research on such capabilities are considered. Media influences on the structure, formation, and modification of trainee mental models are stressed. For example, the computer's ability to graphically represent and provide trainees control over concrete objects and abstract concepts may help novices form more advanced mental models. In conclusion, the review proposes that training methods and media should have an integral relationship; that the medium enables and constrains the method. Training, including computer skills, should exploit media capabilities to improve training.

* * * *

Lee, D.M.S., Pliskin, N., & Kahn, B. (1994). The relationship between performance in a computer literacy course and students' prior achievement and knowledge. <u>Journal of Educational Computing Research</u>, <u>10</u>(1), 63-77.

This study examined the effects of different factors on performance in a college level computer literacy course. In the present study, researchers evaluated the performance of 140 undergraduate students in an introductory level computer class. The researchers found many predictors of performance in the course. First, the students' high school academic achievements (total Scholastic Aptitude Test (SAT) scores, quantitative SAT scores, and high school class rank) were significant predictors. Second, previous programming experience was a predictor, even though programming was not required of students in the present computer course. Finally, students who expected to perform well outperformed those who did not have such an initial expectation. The authors suggest that by dividing computer courses into levels based on students' abilities, all students would benefit greatly because the more computer literate students would not be held back and the slower students would not be as frustrated.

* * * *

Lewis, C., & Mack, R. (1981). Learning to use a text-processing system: Evidence from "thinking aloud" protocols. <u>Proceeding of the Conference on Human Factors in Computer Systems</u> (pp. 387-392). New York: Association for Computing Machinery.

Six office temporaries were asked to learn to use a word processing system over four half-days. The methods and results of this study were exactly the same as later studies conducted by Carroll and Mack (1983; 1984). (For a complete review, see Carroll & Mack, 1984.) The authors attribute learners' problems with computer systems to not reading the manual before they try to do something. Other factors that can lead to problems include: failing to understand that the system is in constant insert mode, presenting too much material for learners to remember, and using computer jargon that novices do not understand. The authors conclude that novices should not be exposed to computers through unaided self-study. Instead, novices should be led through some form of learning through discovery.

* * * *

Mack, R.L., Lewis, C., & Carroll, J.M. (1983). Learning to use word processors: Problems and prospects. ACM Transactions on Office Information Systems, 1(3), 254-271.

This article described in greater detail the types of problems subjects in the Carroll and Mack (1984) study faced and practical solutions for those problems. (For a complete review of the study, see Carroll & Mack, 1984.) Improvements to word processing systems are needed to make the programs more user friendly for novices. Several suggestions are offered on how to solve the difficulties people face in learning to use a computer system. For example, learners should be given realistic expectations about the time needed for learning. Another suggestion is to use less jargon and to use more examples. Finally, more user testing is needed to get rid of even the minor rough spots in a system.

* * * *

Mandinach, E.B., & Linn, M.C. (1987). Cognitive consequences of programming: Achievements of experienced and talented programmers. Journal of Educational Computing Research, 3(1), 53-72.

The goal of this study was to gain a better understanding of the problem solving skills possessed by students who are at the top of their classes in computer programming. The authors were interested in finding out what factors contribute to advanced computer programming at the middle school level. Students from several middle schools were nominated by their teachers as being excellent programming students. These students were then tested and the top 24 were chosen to participate as the talented programmers, while the rest (98) represented successful students. The students were tested on a variety of programming skills. Surprisingly, it was found that programming performance could not be predicted by general ability, amount of access to computers, or gender. The most influential factor in programming ability appeared to be computer instruction in school. Those students who had more knowledgeable and talented computer instructors outperformed other students. In conclusion, it is possible for students to become quite proficient at computer skills with appropriate instruction and well-trained teachers.

* * * *

Mané, A.M., Adams, J.A., & Donchin, E. (1989). Adaptive and part-whole training in the acquisition of a complex perceptual-motor skill. Special Volume: The Learning Strategies Program: An Examination of the Strategies in Skill Acquisition. Acta Psychologica, 71 (1-3), 179-196.

Two transfer of training strategies, part training and adaptive training, were examined to find out if either could lead to better acquisition of a computer task than training on the target task itself. In part training, parts of the target task are presented separately and the subject practices each subtask in isolation before receiving the whole task. In adaptive training, the subject starts out with a simpler form of the target task. As the subject progresses, his or her performance is evaluated and the level of difficulty is increased based on performance. This experiment examined subjects' performance on a computer game after training in either a part training, adaptive training, or control condition. Subjects in the control condition received fixed training, in which they practiced the task at the highest level of difficulty the entire time. Subjects who received part training performed better and required less training time than other subjects. Subjects in the control group required more than twice the training time of subjects in the part training group. The subjects in the adaptive training condition, on the other hand, performed no better than the control group. In fact, the adaptive training seemed to lead to some negative transfer in the target task.

* * * *

Marcoulides, G.A. (1988). The relationship between computer anxiety and computer achievement. Journal of Educational Computing Research, 4(2), 151-158.

This study was designed to examine the relationship between computer aptitude, experience, and anxiety. Participants were students that had signed up for a required computer course. At the beginning of class, students were first presented with questionnaires designed to measure computer aptitude, computer literacy, computer anxiety, and computer interest. The dependent variable was the number of completed homework assignments. The results showed that although computer experience plays a significant role in computer achievement, computer anxiety plays an even greater role. Therefore, it is suggested that instructors should try to create less stressful environments for students by using small computer support groups.

* * * *

Margono, S., & Shneiderman, B. (1993). A study of file manipulation by novices using commands vs. direct manipulation. In B. Shneiderman (Ed.), Sparks of innovation in human-computer interaction (pp.39-50). Norwood, NJ: Ablex Publishing Corporation.

This experiment compared file manipulation operations of novices on an Apple Macintosh to the IBM PC with MS-DOS. The hypothesis was that novices would find the Macintosh easier to use since it had a more user-friendly interface than MS-DOS. Thirty subjects performed various tasks on both computer systems. Subjects made significantly more errors using MS-DOS than the Macintosh. Errors that occurred when using the Macintosh included not using the mouse properly and incorrectly selecting, opening, dragging, or renaming a document. Errors that occurred when using MS-DOS for the IBM included forgetting correct punctuation, adding extra spaces, and omitting parameters in the commands. The authors conclude that novices were more satisfied working with the Macintosh, but more studies need to be conducted to examine the performance of frequent and expert users.

* * * *

Marshall, E., & Baker, S. (1989). Training operators to use advanced display systems for process control. In L. Bainbridge & A. Ruiz Quintanilla (Eds.), <u>Developing skills with information technology</u> (pp. 217-226). Chichester: John Wiley & Sons Ltd.

This article examines training techniques for introducing nuclear plant control room operators to two new display systems. These systems are the handling alarms using logic (HALO) system and the critical function monitoring system (CFMS), both of which are only used occasionally, but require that operators be well versed in their use. In order to achieve mastery of these systems, operators must have access and opportunity for frequent practice. Also, better instruction and training are necessary to prepare operators for unexpected events. Through regular repetition, operators will maintain their levels of performance, but some events, such as handling severe disturbances, are so rare that most operators only experience them in simulator training. The authors have observed that process knowledge is retained over time, however, use of the interface is forgotten fairly quickly. Although there is little research pertaining to retention of these skills, the authors feel the annual retraining that most power plants undergo is too infrequent for adequate retention. However, more research is needed in this area.

* * * *

Martocchio, J.J., & Webster, J. (1992). Effects of feedback and cognitive playfulness on performance in microcomputer software training. <u>Personnel Psychology</u>, <u>45</u>(3), 553-578.

The effect of amount of cognitive playfulness in human-computer interactions, which affects motivation, was examined. Cognitive playfulness is defined as "...a multi-faceted construct encompassing five distinct factors: cognitive spontaneity, social spontaneity, physical spontaneity, manifest joy, and sense of humor" (p. 557). Of these five factors, cognitive spontaneity is the one that plays a role in human-computer interactions. The authors hypothesized that those higher in cognitive playfulness would outperform those lower in cognitive playfulness on an objective test of software knowledge. Also, it was hypothesized that those who receive positive feedback would outperform those who receive negative feedback on an objective test of software knowledge. Participants were 68 employees of a public university who were enrolled in a course to learn the merging feature of WordPerfect 5.0. Although the sample is the same as in the Webster and Martocchio (1993) study, the authors state that these data were collected at a different point in time. Training consisted of one four-hour session on the mail merge feature of WordPerfect. Training began with a class on the features of WordPerfect. After the instructor lectured on a topic, participants were asked to practice the feature.

Immediately after one hour of training, participants completed an objective test of performance. After completion of the test, participants took a break while the instructor provided feedback on the tests. Feedback was either positive or negative and was not based on actual performance. After reviewing the feedback, participants were taught additional features of the mail merge feature in the same format as before and received a second test of performance followed by a second evaluation that was consistent with the first (either positive or negative). Analysis of the data showed that both hypotheses were supported. Positive feedback resulted in better performance than negative feedback and higher cognitive playfulness resulted in better performance than lower cognitive playfulness. Future research should examine the effects of cognitive playfulness as well as type of feedback on training transfer.

* * * *

Mayer, R.E. (1975). Different problem-solving competencies established in learning computer programming with and without meaningful models. Journal of Educational Psychology, 67(6), 725-734.

These experiments attempted to examine how the amount and type of computer programming exercises would affect later programming performance. In Experiment 1, four methods of instruction were used--a diagram model that compared computer processes to tasks already familiar to the subjects, a rule model that introduced the same information without the practical framework, a rule-flowchart model that was the same as the rule model with a basic flowchart of how the information was connected, and a diagram-flowchart model that presented the diagram model as well as the flowchart. In addition, the amount of practice was varied (practice vs. no practice). All eighty subjects read an instructional booklet at their own pace. When they were finished, those in the practice condition were given eight programming practice items with feedback for each item. All subjects were then presented with the posttest which had 18 problems with no feedback. Although practice had no effect on performance on the posttest, subjects who were in either the diagram or diagram-flowchart conditions solved more problems than the other two groups.

In Experiment 2, 40 subjects were randomly assigned to one of the four conditions: diagram, rule, rule-flowchart, or diagram-flowchart. In addition, general mathematics ability (high or low) was measured. Subjects were given the same instruction sets as in Experiment 1. Afterwards, they were asked to complete three sets of problems. Sets had 24 (12 generation and 12 interpretation) problems, and subjects had to solve to a criterion of four correct problems in a row in both Sets 1 and 2. In Set 3, subjects had to correctly complete all 12 items. The results showed that subjects who were in one of the diagram conditions outperformed subjects in the rule conditions on interpretation items, but were poorer at solving generation items. In addition, low ability subjects benefited more from the diagram conditions while high ability subjects benefited more from the rule conditions.

Experiment 3 manipulated the type of problems the 56 subjects practiced. Subjects were placed in either a diagram or rule condition and were given either interpretation or generation practice problems. Again, mathematical ability was assessed in the same manner as in Experiment 2. Generation practice problems increased posttest scores for subjects who received the diagram instructions, while interpretation practice problems were more helpful for subjects in the rule condition. The author concludes that diagram models are more helpful for lower ability learners because they provide the learner with a set of familiar experiences which can provide a framework for understanding and organizing new information. However, a diagram model may actually interfere with the learning of higher ability subjects.

* * * *

Mayer, R.E. (1981). The psychology of how novices learn computer programming. <u>ACM Computing Surveys</u>, <u>13</u>(1), 121-141.

Eight experiments that either provide novices with a concrete model of the computer or ask learners to put technical information in their own words are reviewed. According to the author, these two techniques will aid novices in learning computer programming. The techniques will also facilitate the retention process and more importantly, make transferring of the material to new situations easier. Specifically, concrete models (otherwise known as advance organizers) are most useful when learners are unfamiliar with the area, have low ability, are inexperienced, or are expected to transfer the information to new situations. In addition, the results of four experiments are presented that support the use of advance organizers in the previously mentioned situations.

The second technique, asking novices to put the information into their own words, is helpful because it allows learners to connect the new information with their existing knowledge. Again, four experiments are presented that provide support for the idea of having novices put new or technical information into their own words. The author concludes that both techniques are useful for helping novices learn computer programming.

* * * *

O'Quin, K., Kinsey, T.G., & Beery, D. (1987). Effectiveness of a microcomputer-training workshop for college professionals. Computers in Human Behavior, 3(2), 85-94.

The authors developed a workshop for college faculty and staff in order to increase computer proficiency, understanding, and knowledge, and decrease nervousness about using computers. Only new and relatively inexperienced computer users were allowed to participate. Participants received a notebook with reference material and exercises that were completed in the workshop. The participants were taught in a group with either one or two instructors over three weeks. By the end of training, participants increased in computer proficiency, reported being more knowledgeable, thought computers were more understandable, and were not as nervous about using computers. Proficiency did not vary between males and females or younger and older participants. The authors suggest using plenty of hands-on training and allowing enough practice time so that everyone gets as much practice as they need.

* * * *

Palmiter, S., & Elkerton, J. (1993). Animated demonstrations for learning procedural computer-based tasks. Human-Computer Interaction, 8(3), 193-216.

The researchers assessed acquisition, retention, and transfer on an Apple Macintosh application named HyperCard. Participants were 59 paid volunteers who were assigned to one of three groups: text only (participants were shown procedural instructions on the computer screen and were asked to read at their own pace), demonstration only (participants viewed animated demonstrations of the same tasks on the computer screen but without any accompanying text), or demonstration text (the exact text used in the text only condition was presented in spoken form along with the animated demonstration). It was expected that the demonstration groups would be faster and more accurate than the text only group during training but may not remain superior over time. A variety of tasks from simple (three procedural steps) to more complex (12 procedural steps) were learned in a training session. Participants were tested immediately after the session on tasks identical or similar to those in training. Participants had one opportunity to perform these tasks without any instructions. Finally, participants came back a week later and were again tested on the same tasks as in the previous test. All participants received approximately 30 minutes of instructional time. The first day of the experiment lasted 90-120 minutes while the second day (seven days later) lasted between 20-40 minutes.

After a week, participants returned to perform the same tasks as in the immediate test session. In this session, participants received five attempts at correctly performing each task. Accuracy between training and the delayed test decreased for the two demonstration groups and increased for the text only group. Although both demonstration groups had more correct trials at training than the text only group, the text only group had more correct trials than the demonstration only group at the delayed test. In addition, during training, the text only group was slower at completing the tasks than the other two groups, but by the time of the delayed test, the text only group was faster and the demonstration groups were slower. Finally, the text only group had an easier time transferring the skills they had acquired to a similar task than the other two groups. These results suggest that the demonstration groups "...did not maintain their superior training performance..." (p. 207) whereas the text only group was consistent. One possible explanation for the results is that the demonstration groups were engaging in superficial processing of the information and had not sufficiently learned the procedures necessary to complete the tasks. Although the findings supported the use of text only, the authors argue that participants prefer demonstrations and more research needs to be conducted to improve ways of presenting animated demonstrations.

* * * *

Palmiter, S., Elkerton, J., & Baggett, P. (1989). <u>Animated demonstrations versus written instructions for learning procedural tasks</u>. Technical Report C4E-ONR-2. Center for Ergonomics, Department of Industrial and Operations Engineering: The University of Michigan.

For a review, see Palmiter, Elkerton, & Baggett, 1991.

* * * *

Palmiter, S., Elkerton, J., & Baggett, P. (1991). Animated demonstrations vs written instructions for learning procedural tasks: A preliminary investigation. <u>International Journal of Man-Machine Studies</u>, <u>34</u>(5), 687-701.

Interest has been increasing the use of animated demonstrations for teaching computer programming. However, little data exists on whether animated demonstrations help users and increase the efficiency of training. The researchers hypothesized that animated demonstrations would improve initial learning in comparison with written instructions since demonstrations would reduce the amount of cognitive processing necessary during learning. Twenty-eight students and staff from a major university participated in the study. Participants received either animated demonstrations or written instructions and either one or three practice trials during instruction. All participants received a training session, a test immediately following training, and a delayed test approximately three days after training. Tasks for the two test sessions were either the same as the tasks in the training session, or similar to the tasks in the training session. Training time was approximately 90 minutes for all participants.

During training, participants in the written text condition took 50% more time to complete the designated tasks. For the identical tasks, those who were in the written text condition took less time than participants in the animated demonstration condition on both test sessions. Users of the written text improved over the three sessions, while users of the animated demonstrations showed no improvement. In addition, the animated demonstration group was more accurate than the written text group during training, but showed no improvement across the three sessions while the text group improved over the three sessions. For similar tasks, the written text group experienced positive transfer and completed the tasks more quickly than the original tasks, whereas the animated demonstrations group experienced negative transfer and took more time to complete the similar tasks than the original tasks. These results suggest that those in the animated demonstrations group were not able to generalize the training they received. A list of guidelines are suggested for creating animated demonstrations for procedural tasks. Some of the guidelines include: adding a verbal component to the demonstration, not making procedures that are too long or complex, creating animations that will not simply be mimicked, and carefully pacing the demonstration.

* * * *

Parton, D., Huffman, K., Pridgen, P., Norman, K., & Shneiderman, B. (1985). Learning a menu selection tree: Training methods compared. Behaviour and Information Technology, 4(2), 81-91.

This research project was designed to examine what type of menu structure is easiest for novices to learn and recall. The experimenters designed a simple menu structure consisting of three levels with three choices at each level. The menu structure was presented to the subjects in one of four ways: trial and error, in which subjects went through the menu system on their own for 12 minutes; command sequence, in which subjects studied all possible command sequences; frame presentation, in which subjects were presented with each of the 13 menu frames; or global tree, in which subjects were presented with the overall menu structure. After the study period, subjects were given 10 minutes to find as many of the target words as they could. The number of target words found as well as the average number of tries to reach target words was measured for each subject. Finally, subjects were presented with a blank menu tree and were asked to fill in as many of the menus as they could recall. Analysis of the data showed that the global tree group recalled more menu options than the other three groups. Although there were no differences for the number of targets found between groups, the global tree and trial and error groups made marginally fewer errors. The authors conclude that the global tree and trial and error methods are the best ways to teach menu structures to novices. However, these methods may not be the most efficient for experts.

* * * *

Pea, R.D., & Kurland, D.M. (1984). On the cognitive effects of learning computer programming. New Ideas in Psychology, 2(2), 137-168.

The aim of this article was to examine how computer programming is learned based on two opposing beliefs. The first belief is based on the behaviorist tradition and states that learning computer programming is just an accumulation of facts. The second belief is a reaction to the behaviorist view and states that by learning programming, people are acquiring much more than just facts. According to this second view, programming students should also be learning problem solving skills as well as higher mathematical skills. The authors believe that the amount of information that programming students learn is not based on the programming language, but on the quality of the teacher available to them. The article then goes on to discuss the four different levels of programming skill: program user, code generator, program generator, and software developer. Although no research has been conducted to find out how people reach these different ability levels, most researchers mention at least six factors: mathematical ability, memory capacity, analogical reasoning skills, conditional reasoning skills, procedural thinking skills, and temporal reasoning skills. The authors conclude by saying that the "...cognitive constraints on developing programming skills are currently unknown" (p. 157). Future research needs to focus more on the cognitive effects of programming before we will know if computer programming really has more far reaching effects.

* * * *

Piedmont, M. (1991). Put computer software training $$$ to better use. HR Focus, 68(11), 13.

Training efficiency is being affected by two different realities. New software programs have become much more complex and contain many more options. However, most company employees have a narrow range of duties which do not require the use of all the software features. An analysis of the way employees use software programs found that some employees used only 10 percent of a program's features, while most employees never used more than 25 percent. Unfortunately, most training has become generic and is not tailored to individual needs. By finding out what employees need to know, trainers can ignore the features of a software program that do not apply to the people they are training. Learning less and relating the learning more to an individual's job will help employees retain the information more than current training methods.

* * * *

Reiser, B. (1995). Causal models in the acquisition and instruction of programming skills. (ARI Research Note 95-15). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

This researcher constructed a tutoring system to teach students to program in List Processing (LISP). The system can function either as a tutor or as an exploratory learning guide. Although no data is presented, the author believes that his computer program can be as successful as human tutors. This computer program, Graphical Instruction in LISP (GIL), seems to facilitate learning and retention by helping students understand more of what they are doing as they are solving problems.

* * * *

Ricci, K.E., Salas, E., & Cannon-Bowers, J.A. (1996). Do computer-based games facilitate knowledge acquisition and retention? Military Psychology, 8(4), 295-307.

Previous research has found that computer-based games increase retention. The purpose of this study was to evaluate specific components of computer-based games to find out what components may lead to effective military training. Participants (60 students at the Naval Training Center in Orlando, Florida) received Chemical, Biological, and Radiological Defense (CBRD) training in one of three randomly assigned conditions: text, test, or game. In the text condition, participants learned the material through a text-based format. In the test condition, participants learned the material through paper-based questions and answers. In the game condition, participants learned the material through computer-based gaming. All participants first completed a pretest and were then presented with the materials corresponding to their training condition. They were told to learn as much as they could about CBRD during the 45 minutes of training. After this training period, participants were presented with a posttest. Four weeks later, 58 participants returned to take a multiple choice retention test. Participants in the game and test conditions scored higher on the posttest than participants in the text condition. In addition, participants in the game condition performed better on the retention test than participants in the text condition, and were the only group to perform better on the retention test than on the pretest. Finally, participants in the game condition rated their training as more enjoyable and preferable than those in the text and test conditions. The authors conclude that computer-based gaming can increase the learning and retention of knowledge through increased motivation.

* * * *

Ross, S.M., Smith, L.S., & Morrison, G.R. (1991). The longitudinal influences of computer-intensive learning experiences on at-risk elementary students. Educational Technology, Research, & Development, 39(4), 33-46.

The present study was designed to assess the impact of a special computer intensive learning environment for elementary school students. Twenty-five students participated for one year in a classroom environment that consisted of computer hardware, software, specially trained teachers, and technical assistance. Each student and teacher had an Apple IIe™ computer at school and another at home. A group of 30 students who were not enrolled in the computer-intensive classroom and did not work with computers at school were used as a control group. These students lived in the same neighborhood and were comparable to the computer-intensive group in motivation, socioeconomic status, and home environment. Students were tested one year later on a variety of dimensions, but results were inconclusive. Although there were some differences favoring the computer-intensive group over a control group, most were not significant. The only advantage of the computer group was in their keyboarding skills. The authors explain their results by saying that after the intensive training, students did not have access to computers for the year before they were evaluated. Working with computers in the past is of little value if they are no longer available for practice and use.

* * * *

Russon, A.E., Josefowitz, N., & Edmonds, C.V. (1994). Making computer instruction accessible: Familiar analogies for female novices. Computers in Human Behavior, 10(2), 175-187.

These researchers were interested in teaching basic computer skills to females who had no previous experience with computers. Other research has found that females are often underrepresented in computer courses and computer-related jobs. In addition, women avoid formal computer instruction more than men. Therefore, this experiment was designed to find a better way to teach women computer skills. One group of subjects received a packet of normal instructional materials while the other group received the same materials and a practical analogy for understanding the use of word processing. In this experimental group, word processing was compared to everyday items one could find on a secretary's desk, such as in and out baskets, scissors, tape, and white-out. The researchers hypothesized that the experimental group would make fewer mistakes and ask for help less than the control group on a letter writing task. The results showed that the experimental group outperformed the control group, but there were no differences in the number of requests for help. According to the authors, tutorials that use analogies that people can relate to are more effective than standard tutorials, at least when teaching females computer skills.

* * * *

Salisbury, D.F. (1990). Cognitive Psychology and its implications for designing drill and practice programs for computers. Journal of Computer-Based Instruction, 17(1), 23-30.

The author suggests that in order to perform more complex tasks such as computer programming, many of the subskills need to be automatized. The focus of the article was on five issues in cognitive psychology that are relevant to the formation and optimization of computer drills. The first cognitive issue is interference. In order to reduce this, learners should work on only a small subset of items at a time rather than on all items in a single practice session. Once a subset of items has been learned, new items can be introduced in addition to reviewing the old information. The second issue is spaced practice. Computer drills should not be completed in one sitting, rather, learners should be allowed to stop and resume practice sessions without having to start the drill from the beginning each time. The third issue is spaced review. Computer drills should gradually increase spacing between presentations of the same material in order to increase retention for longer periods of time. The fourth issue is the capacity of short-term memory. Since the capacity of short-term memory is limited, beginning practice sessions should present the smallest possible stimuli. Over time, chunking will occur and meaningful groups of stimuli will be considered as single items. The last issue is how information is represented in memory. Since people are better at remembering meaningful information, the objective of computer drills should not be to make learners memorize information. Instead, drills should attempt to convert tasks which do not have much inherent meaning into more meaningful concepts. If these suggestions are followed, computer drills can be a more effective and efficient way to lead toward mastery of computer skills.

* * * *

Salomon, G. (1990). Cognitive effects with and of computer technology. Communication Research, 17(1), 26-44.

The author suggests that the use of some types of computer software can affect children positively and have lasting effects. Four qualities are identified in computer programs that can affect children's cognitions: a) interactivity--active interaction between the user and the comptuter; b) intelligent guidance and dynamic feedback--providing informative feedback based on the user's current activities; c) multiplicity of symbol systems--providing multiple ways of presenting the same information; and d) supplantation of users' memories-freeing users from having to rely on their own memory to carry out a task. While most studies have failed to find any evidence that the use of computer programs can lead to better planning ability or other real life situations, some studies have found this to be the case in regard to problem solving skills. The author suggests that transfer of these positive qualities can be achieved through activation and internalization.

* * * *

Savage, R.E., & Habinek, J.K. (1984). A multilevel menu-driven user interface: Design and evaluation through simulation. In J.C. Thomas & M.L. Schneider (Eds.), Human factors in computer systems (pp. 165-186). Norwood, NJ: Ablex Publishing Corporation.

The goal of this research article was to design and test a human-computer interface that would be useful to a broad range of people, from programmers to novices. The proposed interface would guide first time users through the steps required for the desired procedures, but it would also allow more experienced users to take shortcuts. For the first part, 12 participants performed programmer tasks, 9 participants performed system operator tasks, and 10 performed work station operator tasks. The participants' keying responses and time on each menu were recorded. The participants read the instructions and used the system menus to accomplish their tasks. They also had access to manuals in case they ran into trouble. If they could not accomplish a certain task, they moved on to the next task. The participants' errors were divided into four sections: a) inconvenience errors which included the participant taking the wrong path but ending up at the correct function, searching and exploring menu options and paths and ultimately ending up at the correct function, or searching and exploring menu options and paths and taking the wrong path but ending up at the correct function; b) path errors which included the participant taking the wrong path to the wrong function, searching and exploring and taking the wrong path to the wrong function, or taking the correct path but selecting the wrong function; c) function errors which occurred when the participants filled in the wrong parameters on the prompt screen when they had reached the correct function; and d) miscellaneous errors which did not fit any of the other categories.

Analysis of the data showed that inconvenience errors were the same for the three tasks and were due to unclear wording of the menu options, lack of understanding of some of the terms, and misinterpretation of how the task needed to be accomplished. Path errors were highest for the work station operator tasks followed by the system operator tasks while function errors were highest for the system operator tasks followed by the work station operator tasks. Programming tasks had the lowest error rates for these two types of errors. Overall, the participants felt that the interface was easy to use and their success at completing the tasks confirmed this to be the case. After revising the interface to meet the users' needs more closely, the authors again tested the system. The second evaluation showed that path and function errors were significantly reduced. In addition, most tasks took less time to accomplish and all had a higher rate of success. In conclusion, the authors suggest that more research is needed to be able to design better user-computer interfaces that match the users' perceptions about computer systems.

* * * *

Sein, M.K., & Bostrom, R.P. (1989). Individual differences and conceptual models in training novice users. Human-Computer Interaction, 4(3), 197-229.

Spatial ability differences (high and low), learning mode differences (concrete and abstract), and mental model representations (concrete and abstract) were examined in learning to use an electronic mail system. Mental models are defined as "...conceptual representations of the system that provide predictive and explanatory powers to the user in understanding the system and guide their interaction with it" (p. 199). It was hypothesized that those with low spatial ability would benefit from a concrete model while those with high spatial ability would benefit from an abstract model. It was also expected that abstract learners would benefit from an abstract model while concrete learners would benefit from a concrete model. Participants were 104 students enrolled in an introductory computer course. They were given 15 minutes to study the training booklet and were asked to focus on the portion that contained the conceptual model. They were then asked to perform 13 tasks, 4 of which had been demonstrated to them (near-transfer), and 9 of which had to be inferred from the training materials (far-transfer). Finally, they were asked to describe the relationship between the various electronic mail system components. Both hypotheses were confirmed. Those with low spatial ability as well as concrete learners benefited most from a concrete model while those with high spatial ability as well as abstract learners benefited most from an abstract model.

* * * *

Shneiderman, B. (1987). Designing the user interface, strategies for effective human-computer interaction (pp. 41-81). Reading, MA: Addison-Wesley.

In Chapter 2, "Theories, Principles, and Guidelines," Shneiderman discusses the types of information people need to remember in order to successfully use a computer system. The first type of information is syntactic knowledge, such as knowing which key erases a character or which function key brings up the previous screen. Syntactic knowledge is very important. However, the learning and retention of this knowledge is hindered by the fact that these details vary across systems and these variations are quite unpredictable. For example, one system may use "S" to save a file while another system uses "S" to send a file. Unless this type of syntactic knowledge is regularly used, it fades from memory. The second type of information is semantic knowledge about computer concepts which is expected to be stable in memory.

The effects of syntactic knowledge can probably be seen most in what the author calls knowledgeable but intermittent users of computer systems. He feels that these users are able to maintain the necessary semantic knowledge, but will have difficulty maintaining the syntactic knowledge. In order to help these users, programs should be designed using "...simple and consistent structure in the command language, menus, terminology, and so on, and by the use of recognition rather than recall" (p. 54). The author also discusses the criteria that determine the quality of a computer program. These five criteria are: time to learn the system, speed of performance, rate of errors, subjective satisfaction, and retention over time. Programmers need to develop a test plan against which they can rate the system for each of these criterion. In conclusion, the most important goal in producing better programs is for researchers to develop working theories and models of human behavior with interactive systems.

* * * *

Shute, V.J., & Gawlick, L.A. (1995). Practice effects on skill acquisition, learning, outcome, retention, and sensitivity to relearning. Human Factors, 327(4), 781-803.

The purpose of these experiments was to find out whether amount of practice has an effect on learning and retention of flight skills. In Experiment 1, 356 novice subjects used a tutoring system designed to teach flight engineering knowledge and skills. The hypotheses were that subjects who received limited practice (abbreviated condition) would take less time to complete the program and would not perform as well as subjects who had more practice opportunities (extended condition). Subjects were randomly assigned to one of four conditions: a) all abbreviated; b) first-half abbreviated, second-half extended; c) first-half extended, second-half abbreviated; or d) all extended. The results from Experiment 1 showed that there were no group differences in learning and accuracy. In Experiment 2, 34 subjects from the first experiment came back two years later and they were tested on retention and sensitivity to relearning. The results from Experiment 2 showed that after two years, those who had learned the information in mixed practice conditions had retained more skill and knowledge than those who had received consistent practice (either all abbreviated or all extended). Since the findings did not support the original hypotheses, the authors attributed the results to the small sample size in Experiment 2 and low power. They recommend another study to replicate these findings.

* * * *

Simon, S.J., & Werner, J.M. (1996). Computer training through behavior modeling, self-paced, and instructional approaches: A field experiment. Journal of Applied Psychology, 81(6), 648-659.

Three training techniques were compared in this experiment: instruction, self-paced/exploration, and behavior modeling. These techniques were compared to each other as well as to a control group. The authors propose that two types of cognitive learning are necessary for computer training. First, learners must have an understanding of the big picture, and second, they must know the specific commands needed for the computer to function. The hypotheses included: a) trainees in the instruction condition would have greater general comprehension than trainees in the exploration condition; b) trainees in the exploration condition would outperform trainees in the instruction condition in skill demonstration; and c) trainees in the behavior modeling condition would outperform all others in general comprehension and skill demonstration. Participants were 160 members of a naval construction battalion who were all novice computer users. Participants were trained for two hours on a data processing system that was being introduced in the Naval Construction Force. Data were collected immediately after training and again four weeks later.

In the instruction condition, training was provided in a lecture format in a conference room. Trainees were shown slides and given written information about the data processing system. In the exploration condition, trainees were seated at a workstation and given a manual to work independently through the exercises. In the behavior modeling condition, trainees were given the same information as in the instruction condition. In addition, they observed the trainer who performed the procedures on a computer, and then they tried to perform the tasks themselves. In the control condition, trainees received no training. Participants spent the two hours in a conference room with the computer manuals which were already available to them.

As predicted, those in the instruction condition had higher levels of general comprehension than those in the exploration condition immediately after training, but comprehension was equivalent between the two groups four weeks later. In addition, skill demonstration was higher in the exploration condition than in the instruction condition at both time periods. Finally, participants in the behavior modeling condition outperformed all others in general comprehension and skill demonstration at both time periods. In conclusion, behavior modeling was the superior technique for both understanding and retention of computer skills. Finally, it should be noted that "trainees in all conditions performed better than participants in the control group, even after a 1-month opportunity to learn on the job" (p. 675).

* * * *

Singley, M.K., & Anderson, J.R. (1989). The transfer of cognitive skill. Cambridge, MA: Harvard University Press.

Chapter 3 book deals with lateral transfer, which is transfer between skills at the same level of complexity. Subjects were 24 women who were taught to use either one or two line editors (ED and EDT) to see if learning on these editors would transfer to a screen editor (EMACS). The difference between the two types of editors is that "line editors display the contents of the file only upon request and force users to enter abstract commands that specify edits on a line-by-line basis. Screen editors, however, fill the screen with the contents of the file and allow users to edit the contents explicitly by moving to a particular location by means of a cursor" (p. 70). Two control groups were also included in the experiment. The experiment was conducted over six consecutive days with three hours of training each day. The first two days were spent practicing on one of the line editors. On the third and fourth days, those in the two line editor condition switched to the other line editor while the other subjects continued working on the same line editor. On the fifth day, all subjects switched to EMACS. When EDT was used on the first two days, there was an improvement in performance on day three with subjects who were switched to ED, which implies almost total transfer. Transfer from ED to EDT was even greater. When subjects were switched to EMACS on the fourth day, those who had prior training on either one or both of the line editors outperformed those in the control groups. From these results, it was concluded that degree of transfer is a function of the overlap in number of shared elements between the text editors.

Chapter 4 deals with negative transfer. Subjects were eight women from the same population as the previous experiment. They were first taught EMACS, then "perverse EMACS" (a jumbled version of EMACS in which the control and escape keys were replaced or scrambled), and finally EMACS again. The hypothesis was that each switch would lead to strong positive transfer, and that negative transfer would not take place. This group was compared to a group that learned EDT and EMACS before perverse EMACS. The researchers hypothesized that the group that learned both text editors before exposure to perverse EMACS would have higher rates of transfer than the other group. The group known as the screen-only group trained two days on EMACS, two days on perverse EMACS, and two days on EMACS again. The group known as the line-and-screen group trained two days on EDT, one day on EMACS, two days on perverse EMACS, and one day on EMACS again. On the first day of using the perverse EMACS, performance declined, but by the second day, performance was no different from those who had used EMACS all along. Finally, performance of the screen-only group was worse than that of the line-and-screen group on perverse EMACS, and the screen-only group continued to lag behind even after switching back to EMACS. In conclusion, the authors point out that "all sustained cases of negative transfer seem to be explainable in terms of the positive transfer of nonoptimal methods" (p. 137).

* * * *

Wallace, D.F., Anderson, N.S., & Shneiderman, B. (1993). Time stress effects on two menu selection systems. In B. Shneiderman (Ed.), Sparks of innovation in human-computer interaction (pp. 89-95). Norwood, NJ: Ablex Publishing Corporation.

The purpose of this experiment was to compare two types of menu structures: broad/shallow and narrow/deep. Broad/shallow menus have more options at each level and fewer levels. Narrow/deep menus have fewer options at each level and more levels. Most studies have found that performance increases with the use of broad/shallow menu structures. For this experiment, the researchers added another dimension: time pressure. Although no specific hypotheses were presented, the authors did assume that they would find an interaction effect between menu type and time pressure. Subjects searched for target items in a hierarchical menu; in the broad/shallow condition, there were three levels with four items each and in the narrow/deep condition, there were six levels with two items each. Subjects under the time constraint took longer to complete the tasks than subjects not under the time pressure. In addition, subjects in the broad/shallow menu condition took less time to complete the tasks than subjects in the narrow/deep menu condition. Finally, subjects under time pressure made more errors than subjects who were not under time constraints, and subjects in the broad/shallow menu condition made fewer errors than subjects in the narrow/deep menu condition. The authors suggest further research to examine the effect of time pressure more closely.

* * * *

Webster, J., & Martocchio, J.J. (1993). Turning work into play: Implications for microcomputer software training. Journal of Management, 19(1), 127-146.

This study examined the effects of labeling training as play or work and trainee age on the learning of computer skills. It was hypothesized that younger individuals would outperform older individuals on objective tests of software mastery and that participants in training sessions labeled as play would outperform participants in training sessions labeled as work, especially among younger participants. However, the authors do not mention the criteria used to determine younger and older ages. Participants were 68 employees of a university enrolled in a course to learn features of WordPerfect 5.0. Although the sample is the same as in the Martocchio and Webster (1992) study, the authors state that these data were collected at a different point in time. Training consisted of a one hour session on the use of the mail merge feature of WordPerfect. The work and play conditions received the same training, except the sessions were labeled differently. After completion of training, participants received an objective test on the material covered in training. The hypothesis proposing that younger individuals would outperform older individuals was not supported. However, the authors did find a strong negative correlation between age and training outcome in the play condition, suggesting that younger individuals outperformed older individuals in the play condition, but not in the work condition.

* * * *

Wilson, M.D., Barnard, P.J., & MacLean, A. (1985). Analysing the learning of command sequences in a menu system. In P. Johnson & S. Cook (Eds.), People and Computers: Designing the Interface. Proceedings of the Conference of the British Computer Society Human Computer Interaction Specialist Group (pp.63-75). University of East Anglia, Cambridge, UK: Cambridge University Press.

In this experiment, users were trained to use three programs: a word processor, a graphics package, and a spreadsheet. Over the course of eight sessions, subjects' progress was monitored with the use of three assessment tests which measured performance on 9 basic functions, 18 additional functions, and 3-6 novel functions. Performance on the 9 basic functions was measured at five different times throughout the course of the experiment. The results showed a decrease in the amount of time it took to complete tasks from the first test to the fifth test. In addition, subjects attempted and completed more tasks on the fifth test than on the first test. However, complexity of the task did not seem to determine the number of errors that subjects committed. From their results, the authors conclude that the acquisition of computer skills does not seem to be a single course of development "...from 'problem solving' through 'integration' to 'automated' responses" (p. 74). Not surprisingly, the acquisition of computer skills appears to be more complex, although its true nature is not yet known.

* * * *

Appendix B

Unobtained References

Barnard, P.J., Hammond, N.V., MacLean, A., & Morton, J. (1982). Learning and remembering interactive commands. Proceedings of the conference of human factors in computer systems (pp. 2-7). New York: Association for Computing Machinery.

Bikson, T.K., & Gutek, B.A. (1983). Training in automated offices: An empirical study of design and methods (Report No. WD-1904-RC). Santa Monica, CA: The Rand Corporation.

Black, J.B., & Moran, T.P. (1982). Learning and remembering command names. Proceedings on the CHI '82 Conference on Human Factors in Computer Systems, 8-11. New York: ACM.

Bovair, S., Kieras, D.E., & Polson, P.G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. Human-Computer Interaction, 5, 1-48.

Carroll, J.M., & Lasher, M. (1981). Getting to know a small computer. Manuscript, IBM Watson Research Center.

Carroll, J.M., Mack, R., Lewis, C., Grischkowsky, N., & Robertson, S. (1984). Learning to use a word processor by guided exploration. Research Report RC 10428. Yorktown Heights, NY: International Business Machines, Research Division.

Charney, D.H., & Reder, L.M. (1987). Initial skill learning: An analysis of how elaborations facilitate the three components. In P. Morris (Ed.), Modelling cognition. Chichester, England: John Wiley & Sons.

Compeau, D.R., & Higgins, C.A. (1995). Application of social cognitive theory to training for computer skills. Information Systems Research, 6(2), 118-143.

Eason, K.D. (1989). Patterns of usage of a flexible information system. In S.D.P. Harker and K.D. Eason (Eds.), The application of information technology. London: Taylor and Francis.

Erber, J., & Botwinick, J. (1983). Reward in the learning of older adults. Experimental Aging Research, 9, 43-44.

Foltz, P.W., Davies, S.E., Polson, P.G., & Kieras, D.E. (1988). Transfer between menu systems. Proceedings of the CHI '88 Conference on Human Factors in Computing Systems, 107-112.

Frese, M., Ulich, E., & Dzida, W. (Eds.). (1987). Psychological issues of human-computer interaction in the workplace. Amsterdam: North-Holland.

Harrap, K. (1990). CBT and classroom training: A winning combination. Computing Canada, 16, 41-44.

Hartley, A.A., Hartley, J.T., & Johnson, S.A. (1984). The older adult as computer user. In P.K. Robinson, J. Livingston, & J.E. Birren (Eds.), Aging and technological advances (pp. 347-348). New York: Plenum Press.

Hays, R.T., & Singer, M.J. (1989). Simulator fidelity in training system design. Bridging the gap between reality and training. New York: Springer-Verlag.

Hurlock, R., & Montague, W. (1982). Skill retention and its implications for Navy tasks: An analytical review. NPRDC SR 82-21. San Diego, CA: Navy Personnel Research and Development Center.

Johnston, B.R., & Fletcher, J.D. (1995). Effectiveness of Computer-Based Instruction in Military Training. Alexandria, VA: Institute for Defense Analysis.

Jones, M.H. (1983). Sex differences in achievement on cognitive dimensions of computer literacy. Unpublished Ph.D. dissertation, University of Virginia, Charlottesville, Virginia.

Kieras, D.E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. Cognitive Science, 8, 255-273.

Landauer, T.K., Galotti, K.M., & Hartwell, S. (1983). Natural command names and initial learning: A study of text editing terms. Communications of the ACM, 26, 495-503.

Muncher, L. (1988). The acquisition of spreadsheet skills. Unpublished Master's Thesis, Department of Psychology, University of Colorado, Boulder, Colorado.

Olfman, L., & Mandiviwalla, M. (1994). Conceptual versus procedural software training for graphical user interfaces: A longitudinal field experiment. MIS Quarterly, 18, 405-426.

Olfman, L., Sein, M.K., & Bostrom, R.P. (1986). Training for end-user computing: Are basic abilities enough for learning? Proceedings of the Twenty-Second Annual Computer Personnel Research Conference, 1-11. New York: ACM.

Poplin, M.S., Drew, D.E., & Gable, R.S. (1984). Computer aptitude, literacy, and interest profile. Austin, TX: Pro-Ed Publishers.

Rediffusion Simulation Ltd. (1986). Simulation in training--A guide for trainers and managers. Training Technology Section, Manpower Services Commission.

Sein, M.K. (1988). Conceptual models in training novice users of computer systems: Effectiveness of abstract vs. analogical models and influence of individual differences. Unpublished doctoral dissertation, Indiana University, Bloomington, Indiana.

Sein, M.K., Bostrom, R.P., & Olfman, L. (1987). Conceptual models in training novice users. Proceedings of INTERACT '87 IFIP Conference on Human-Computer Interaction, 861-867. New York: Elsevier North-Holland.

Singley, M.K., & Anderson, J.R. (1988). A keystroke analysis of learning and transfer in text-editing. Human-Computer Interaction, 3, 223-274.

Winter, S.J., Gutek, B.A., & Chudoba, K.M. (1992, December). Misplaced resources? Factors associated with computer literacy among end-users. Paper presented at the 13th Annual International Conference of Information Systems, Dallas, TX.

Appendix C

List of Acronyms

ARI .... U.S. Army Research Institute
AWE .... Advanced Warfighting Experiment

C^4I .... Command, Control, Communication, Computer and Intelligence
CAI .... Computer-Assisted Instruction
CBRD .... Chemical, Biological, and Radiological Defense
CFMS .... Critical Function Monitoring System

DTIC .... Defense Technical Information Center
DURESS II ..... Dual Reservoir System Simulation II

ERIC .... Educational Resources Information Center

FBC .... Future Battlefield Conditions

GIL .... Graphical Instruction in LISP

HALO .... Handling Alarms Using Logic

ISD .... Instructional Systems Design

LISP .... List Processing

MATRIS .... Manpower and Training Research Information System

SAT .... Scholastic Aptitude Test

TSP .... Training Support Package

USAARMC .... U.S. Army Armor Center