DOCUMENT RESUME

ED 354 872 IR 015 962

AUTHOR	Russell, Daniel M.; Pirolli, Peter
TITLE	Computer Assisted Instructional Design for Computer-Based Instruction. Final Report. Working Papers.
INSTITUTION	National Center for Research in Vocational Education, Berkeley, CA.
SPONS AGENCY	Office of Vocational and Adult Education (ED), Washington, DC.
REPORT NO	NCRVE-MDS-149
PUB DATE	Dec 92
CONTRACT	V051A80004-90A
NOTE	11p.
AVAILABLE FROM	Materials Distribution Service, Western Illinois University, 46 Horrabin Hall, Macomb, IL 61455 ($2).
PUB TYPE	Reports - Descriptive (141)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Artificial Intelligence; Business Education; Cognitive Processes; *Computer Assisted Design; *Computer Assisted Instruction; Computer Software; *Epistemology; Information Dissemination; *Instructional Design; Instructional Development; Postsecondary Education; Problem Solving; Teacher Education; Vocational Education
IDENTIFIERS	Cognitive Sciences

ABSTRACT

Recent advances in artificial intelligence and the cognitive sciences have made it possible to develop successful intelligent computer-aided instructional systems for technical and scientific training. In addition, computer-aided design (CAD) environments that support the rapid development of such computer-based instruction have also been recently developed. The researchers tailored a particular CAD system for instruction, called the Instructional Design Environment (IDE), for use in vocational training. This project brought together a consulting team that included a successful instructor in business education, cognitive scientists, workers in teacher education, and the IDE development team. The main goals of the project were to develop an instructional design methodology that teaches software use in the context of solving realistic problems and to extend the IDE to support this methodology, which is grounded in cognitive science research and is called example-based minimalist design (EBMD). It was found that the use of IDE has several side effects: (1) IDE encourages a greater depth of analysis and planning; (2) the semiformal representation language used in IDE shapes the design process and the manner in which the designer thinks about instruction; (3) the analyses and specifications developed in IDE provide an explicit design rationale for each product; and (4) designs and design rationales developed in IDE can be easily modified and reused thus standardizing instructional development and promoting dissemination of successful design methodologies. (Contains 11 references.) (ALF)

NCRVE

National Center for Research in Vocational Education

University of California, Berkeley

Working Papers

COMPUTER ASSISTED INSTRUCTIONAL DESIGN FOR COMPUTER-BASED INSTRUCTION

Daniel M. Russell
(Xerox Palo Alto Research Center and the Institute for Research on Learning)

Peter Pirolli
(Institute for Research on Learning)

University of California at Berkeley

National Center for Research in Vocational Education
University of California at Berkeley
1995 University Avenue, Suite 375
Berkeley, CA 94704

Supported by
The Office of Vocational and Adult Education,
U.S. Department of Education

December, 1992 MDS-149

This document is one in a continuing series of Working Papers. It has not been reviewed by NCRVE; therefore, this paper represents the views of its authors and not necessarily those of the Center or the U.S. Department of Education. Other documents cited within this document are not available from NCRVE. NCRVE makes these Working Papers available upon request for informational purposes.

FUNDING INFORMATION

Project Title:	National Center for Research in Vocational Education
Grant Number:	V051A80004-90A
Act under which Funds Administered:	Carl D. Perkins Vocational Education Act P.L. 98-524
Source of Grant:	Office of Vocational and Adult Education U.S. Department of Education Washington, DC 20202
Grantee:	The Regents of the University of California National Center for Research in Vocational Education 1995 University Avenue, Suite 375 Berkeley, CA 94704
Director:	Charles S. Benson
Percent of Total Grant Financed by Federal Money:	100%
Dollar Amount of Federal Funds for Grant:	$5,675,000
Disclaimer:	This publication was prepared pursuant to a grant with the Office of Vocational and Adult Education, U.S. Department of Education. Grantees undertaking such projects under government sponsorship are encouraged to express freely their judgement in professional and technical matters. Points of view or opinions do not, therefore, necessarily represent official U.S. Department of Education position or policy.
Discrimination:	Title VI of the Civil Rights Act of 1964 states: "No person in the United States shall, on the grounds of race, color, or national origin, be excluded from participation in, be denied the benefits of, or be subjected to discrimination under any program or activity receiving federal financial assistance." Title IX of the Education Amendments of 1972 states: "No person in the United States shall, on the basis of sex, be excluded from participation in, be denied the benefits of, or be subjected to discrimination under any education program or activity receiving federal financial assistance." Therefore, the National Center for Research in Vocational Education project, like every program or activity receiving financial assistance from the U.S. Department of Education, must be operated in compliance with these laws.

Computer Assisted Instructional Design for Computer-based Instruction

Final Report to the
National Center for Research on Vocational Education

Daniel M. Russell

Xerox Palo Alto Research Center
and the
Institute for Research on Learning

and

Peter Pirolli

University of California, Berkeley
and the
Institute for Research on Learning

Project Abstract

Recent advances in artificial intelligence (AI) and the cognitive sciences have made it possible to develop successful intelligent computer-aided instructional (ICAI) systems for technical and scientific training. In addition, computer-aided design (CAD) environments that support the rapid development of such computer-based instruction have also been recently developed. We tailored a particular CAD system for instruction, called IDE (the Instructional Design Environment) for use in vocational training. The project brought together a consulting team that included a successful instructor in business education, cognitive scientists, workers in teacher education, and the IDE development team. This group, working at the Institute for Research on Learning developed an IDE prototype for the development of computer-based training in business education.

Specific Objectives

Many areas of instruction and training are changing in reaction to the integration of computers into the workplace. Unfortunately, prevailing methods of instruction tend to dissociate teaching about subject-matter areas, such as accounting, from teaching about software, such as spreadsheets. The main goals of our project were to (a) develop an instructional design methodology that teaches software use in the context of solving realistic problems, and (b) to extend the Instructional Design Environment (IDE, Pirolli & Russell, 1990) to support this methodology. The design methodology we have developed is grounded in cognitive science research on the acquisition of problem-solving skills, and the analysis and training of software use. We claim that this design methodology provides a more detailed and accurate analyses of tasks, cognition, and conceptual knowledge than other methodologies (Gagne & Briggs, 1979; Reigeluth, 1983) that de-emphasize the importance of acquiring knowledge in the context of its typical use.

This work involved two substantial ancillary projects. The first involved porting IDE from expensive Xerox Lisp Machines to the more accessible MacIntosh platform. The new version of IDE runs in the FoxBase database system. The second ancillary project was essentially a test of the usability of IDE by a students in the Stanford Teacher Education Program: IDE was used by students in a curriculum and instruction course (lead by Lorretta Kelly) to design an entire high school algebra curriculum. A report on that effort is presented in Russell and Kelly (1991).

The Instructional Domain and its Problems

We chose business instruction incorporating the use of spreadsheets as our application domain for developing our instructional methodology and IDE. In part, this was motivated by the availability of Business Magnet School in a nearby metropolitan area that was facing the task of updating its curriculum to incorporate the use of spreadsheet instruction into its curriculum. The school had several well-equiped classrooms containing Apple II, MacIntosh, and IBM computers, its teachers had several years of experience in developing courses on the use of other application software such as word processors, and the Business Department Chair and its teachers volunteered to provide advice and to allow observation of the classrooms.

As part of our background research, we examined both state curriculum guidelines for vocational education and training packages developed by leading software vendors that are used both in office training and high schools. We found both to be seriously flawed and this opinion was shared by our contacts in the Business Magnet School. California state curriculum standards on spreadsheet use list objectives that consist mainly of naming or defining various concepts and exhibiting basic skills. The training packages adopt the paradigm of teaching elemental concepts and skills before teaching more complex elements. Both the curriculum standards and training packages miss the point that the main purpose of spreadsheets is serve as an aid to perform accounting, make projections, analyze different business scenarios, prepare reports, and so on.

Framework

The design methodology we have developed is called *example-based minimalist design* (or EBMD in short). The methodology integrates aspects of three lines of research. The first (Card, Moran, & Newell, 1983) is research in human-computer interaction on generic engineering models for the analysis of tasks and user cognition and performance. The second (Singley & Anderson, 1989) is research on the acquisition of expertise in highly cognitive domains. The third (Carroll, Smith-Kerker, Ford, & Mazur-Rimetz, 1987-1988) is research on novel training methods tailored for human-computer interaction tasks.

Te purpose of the Card et al. (1983) model is to provide an integrated, albeit approximate, summary of a scientific base--in this case, cognitive psychology. The Card et al. model includes a means of representing the cognitive structures and processes involved in specific tasks. This is the GOMS analysis, which requires specification of (a) the *goals* involved in a task, (b) the *operators* or basic actions available in the task, (c) the set of *methods* for achieving goals, and (d) the *selection* rules for choosing among competing methods. This way of analyzing cognition is, in general, consistent with many more specific theories of cognitive performance.

The EBMD methodology augments GOMS analysis with additional assumptions incorporated from recent information-processing theories of learning by doing. More specifically, our assumptions derive from production system theories of learning and transfer (Singley & Anderson, 1989). EBMD also stresses the role of example solutions in learning in problem-solving domains. When faced with novel problems, the preferred method of problem solving involves the use of example solutions (LeFevre & Dixon, 1986; Pirolli, 1991). The effectiveness of the examples increases with the degree of correspondence between example and target problem solutions (Pirolli, 1991). Using a variant on GOMS analyses (Pirolli, 1991) one can accurately predict the degree to which particular example solutions will facilitate learning. Variants of GOMS analyses have also been used to predict transfer of cognitive skill across problem solving tasks (Singley & Anderson, 1989). Guided by these findings, we can assess the expected instructional effectiveness of example solutions, when new cognitive skills will be acquired in problems solving, and how cognitive skills will transfer to novel problems.

EBMD is also grounded in research on *minimalist instruction* (Carroll, et al., 1987-1988), which is intentionally designed to be briefer than traditional instructional designs. Many methodologies (Gagne & Briggs, 1979; Mager, 1988) dictate bottom-up instructional progressions through hierarchically decomposed learning objectives. Such presentation strategies typically result in instruction that does not immediately address the tasks that learners want to perform, and typically teach many things that are not of any immediate use. One strategy adopted in minimalist instruction is to focus the instruction on real tasks and activities, getting learners to perform real tasks as quickly as possible without full coverage of all that there is to know about an application program. Researchers have also noted that learners typically ignore or skip overly verbose training material (Carroll, et al., 1987-1988), and that shorter texts can actually yield better learning than more elaborate texts (Reder & Anderson, 1980). Minimalist instruction attempts to slash excessive verbiage, in part by assuming that the learner can infer a great deal of information. Minimalist instruction also stresses guided exploration and support of error recovery. Again this tends to reduce instructional material by offloading learning to activities and exercises.

Example-Based Minimalist Design in IDE

The current version of IDE is implemented on a MacIntosh and provides a variety of hypermedia, database, and artificial intelligence functionalities. IDE is loosely based on the NoteCards (Halasz, 1988) metaphor: elements of an analysis or specification are entered on structured notecards and links of various types capture relations among these elements. The cards and links come in various types for representing different kinds of information and relations. These cards and links can also be dynamically generated. To support particular design methodologies one defines a semiformal representation language by specifying how information will be represented in card types and link types. In addition, one typically creates computational tools associated with this representation language that improve or automate aspects of the design process.

IDE was modified to support EBMD for teaching spreadsheet use in business tasks. IDE was integrated with screen recording software, which provides recordings of expert solutions to be used in GOMS tasks analyses and as material for example presentations. IDE was also integrated with HyperCard, so that instructional design specifications could be compiled directly into instructional software. Using this configuration, we screen-record expert solutions, develop GOMS analyses of these solutions, and specify presentations and activities that are compiled into HyperCard instruction.

Screen recording software is used to generate "tapes" of expert problem solutions using spreadsheets. These tapes serve as raw data for GOMS analyses. The tapes are segmented and coded to identify the goals, methods, operators, and selection rules that the experts appear to be using. Typically, this GOMS analysis has to be augmented by inferring additional analyses of problem solving that is not directly manifest in spreadsheet activity. These analyses are specified as hypermedia structures in IDE, that capture the cognitive structures involved in the expert solutions.

Based on these analyses, instruction is specified by organizing modules of instruction that predominantly include example presentations and problem solving activities. The example presentations are also provided by segments of text-augmented screen recordings. Activities usually involve working directly with spreadsheets to solve some particular problem. The specifications of these instructional elements contain a variety of links that indicate, for instance, the flow of control among instructional units, the knowledge needed to perform an activity, and the knowledge to be taught by an instructional unit. These IDE specifications can be passed to a compiler that generates HyperCard stacks that actually conduct the instructional interactions. Individual HyperCard frames may, for instance, present some text, run a screen tape of an example solution, or activate a spreadsheet for an exercise activity.

General Discussion

Clearly, part of the purpose of IDE is to accelerate instructional design and to reduce the cognitive load imposed on designers. However, our development of IDE has also been motivated to alter the design process and the designer in several ways. Instructional analyses and specifications must be described in a semiformal representation language in IDE. We claim that such activity and its products have several side-effects. First, the environment encourages a greater depth of analysis and planning, just as computer programming forces one into more rigorous analyses of problems and solutions. Second, the particular semiformal representation languages used in IDE shape both the particular design process and the manner in which the designer comes to think about and reflect upon instruction. Third, the analyses and specifications developed in IDE provide an explicit design rationale for each particular product. Such rationales provide a framework for structuring instructional evaluations, as well as systematizing new design abstractions and theories over particular design cases. Fourth, designs and design rationales developed in IDE can be easily modified and reused, thereby standardizing instructional development and promoting the dissemination of successful design methodologies. The EBMD methodology is grounded in successful theories of cognition, learning, and its integration into IDE could possibly shape a successful theory of instruction in software-intensive domains.

A Note on the Availability of IDE

IDE 2.0, manuals, and papers describing IDE are available from the Institute for Research on Learning (IRL), 2550 Hanover Street, Palo Alto, CA, 94304 (415) 496-7900. IRL -- a non-profit research organization studying learning and education practices -- makes IDE 2.0 available for a moderate fee to interested parties. Keep in mind that IDE 2.0 is implemented as a FoxBase application, and requires Fox as a separate purchase. Fox is available through your local Macintosh supplier.

Training and technical support for IDE 2.0 is available from Windrose Consulting, 390 Palo Alto Ave., Palo Alto, CA 94301 (415) 327-2543, fax (415) 327-7307.

Project Report Bibliography

Russell, D. M. (October, 1991). Semiformal representations for the analysis of complex domains. Paper presented at the NATO Workshop on Advanced Technologies for Training, Venice, Italy. (Also published in *Advanced Technologies Applied to Training Design*, Chatelier, P., Seidel, R. J. (Eds), 1992, Plenum Books.)

Russell, D.M. & Kelly, K. (1991, April). Using IDE in instructional design: Encouraging reflective instruction design through automated design tools. Paper presented at the Annual Meeting of the American Educational Research Association, Chicago, IL..

Russell, D.M., Kelley, L., & Miner, S. (1990, June). *IDE 2.0 tutorial for C&I algebra project.* [Computer program tutorial] Palo Alto, CA: Institute for Research on Learning.

Russell, D.M. & Miner, S. (1992). *IDE 2.0 reference manual.* Palo Alto, CA: Institute for Research on Learning.

Russell, D.M. & Pirolli, P. (1992, April). *The Instructional Design Environment (IDE): A tool for designing and creating instructional materials*. [Briefing document].

Russell, D.M. & Pirolli, P. (1992, April). *An instructional design environment for cognitive tasks involving software use*. Poster presented at the Annual Meeting of the American Educational Research Association, San Francisco, CS.

References

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction* . Hillsdale, NJ: Lawrence Erlbaum Associates.

Carroll, J. M., Smith-Kerker, P. L., Ford, J. R., & Mazur-Rimetz, S. A. (1987-1988). The minimal manual. *Human-Computer Interaction, 3*, 123-153.

Gagne, R. M., & Briggs, L. J. (1979). *Principles of instructional design* (2nd edition ed.). New York: Holt, Rhinehart & Winston.

Halasz, F. G. (1988). Reflections on notecards: Seven issues for the next generation of hypermedia systems. *Communications of the ACM, 31*, 836-852.

LeFevre, J., & Dixon, P. (1986). Do written instructions need examples? *Cognition and Instruction, 3*, 1-30.

Mager, R. F. (1988). *Making Instruction Work* . Belmont, CA: Lake.

Pirolli, P. (1991). Effects of examples and their explanations in a lesson on recursion: A production system analysis. *Cognition and Instruction, 8*, 207-259.

Pirolli, P., & Russell, D. (1990). The instructional design environment: Technology to support design problem solving. *Instructional Science, 19*, 121-144.

Reder, L. M., & Anderson, J. R. (1980). A comparison of texts and their summaries: Memorial consequences. *Journal of Verbal Learning and Verbal Behavior, 19*, 121-134.

Reigeluth, C. M. (1983). Instructional design: what is it and why is it? In C. M. Reigeluth (Ed.), *Instructional-design theories and models* (pp. 3-36). Hillsdale, NJ: Lawrence Erlbaum.

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