DOCUMENT RESUME

ED 295 673 IR 013 384

AUTHOR	Christensen, Dean L.; Tennyson, Robert D.
TITLE	Application of Heuristic Methods in the Design of Intelligent CAI.
PUB DATE	Jan 88
NOTE	11p.; In: Proceedings of Selected Research Papers presented at the Annual Meeting of the Association for Educational Communications and Technology (New Orleans, LA, January 14-19, 1988). An operating disk with sample lesson is available from the authors. For the complete proceedings, see IR 013 331.
PUB TYPE	Computer Programs (101) -- Reports - Descriptive (141) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Computer Software; Concept Teaching; *Expert Systems; *Heuristics; *Instructional Design; *Programing; Sequential Approach; Time Factors (Learning)
IDENTIFIERS	BASIC Programing Language; *Intelligent CAI Systems; *Learner Controlled Instruction

ABSTRACT

This paper describes the MAIS and presents the BASIC programming code for the heuristic employed in this expert tutor management system, which is based on the findings of investigations of the direct connections among such learning environment factors as individual differences, cognitive learning theory, instructional technology, subject matter structure, and delivery systems. The following computer based variables which the expert system adapts to individual learner differences during the course of the instruction are presented as subroutines: (1) beta value computation; (2) mastery check and advisement; (3) learning time interval; and (4) format of examples and sequence. A figure depicts the instructional variables monitored by the MAIS expert tutor. (11 references) (EW)

Application of Heuristic Methods in the Design of Intelligent CAI

Dean L. Christensen

Control Data Corporation

and

Robert D. Tennyson

University of Minnesota

January 1988

Paper presented in the symposium, "Expert systems and the learning environment," Mariana Rasch (Chair), at the annual meeting of the Association for Educational Communication and Technology, New Orleans.

Dean L. Christensen, Control Data Corporation, Old Shakopee Road, Minneapolis, Minnesota 55440.

Application of Heuristic Methods in the Design of Intelligent CAI

Intelligent computer-assisted instructional (ICAI) systems are characterized as holistic instructional inference-making systems that are iterative in nature such that with experience, they can continuously improve the learning of each individual learner (Tennyson & Park, 1987). This inference-making process is done by an intelligent expert tutor system which actively seeks to improve learning by (a) initially prescribing instruction that has a high probability of preventing learner error and/or misconceptions, (b) that continuously adapts the prescribed instruction according to moment-to-moment assessment and diagnosis, and (c) generatively improves its decision-making system. Figure 1 illustrates the various instructional variables that the MAIS expert tutor management system adapts to individual learner differences and needs during instruction (Tennyson & Christensen, 1988). These variables, termed computer-based enhancements, are managed by the expert tutor employing both formal and informal artificially intelligent (AI) heuristic programming methods (Dorner, 1983; Tikhomirov, 1983).

Insert Figure 1 about here

The MAIS

The intelligent learning system presented in this paper, the MAIS, is based on the findings of an extensive programmatic research effort investigating the direct connections among such learning environment factors as individual differences, cognitive learning theory, instructional technology, subject matter structure, and delivery systems (especially computer-assisted instruction). From the interaction of these factors, a MAIS-based ICAI program can be developed with reasonable success in reference to cost-effectiveness principles of improved learning within standard production costs. That is, unlike conventional ICAI demonstration (or prototype) programs that require costly dependence on powerful hardware and software systems (e.g., a LISP machine), a MAIS-based ICAI program can be developed within current microcomputer constraints of relatively limited memory.

Heuristic Programming

The purpose of this paper is to present the BASIC programming code for the heuristic employed in the MAIS. We have tried to make the following code as generic as possible so that anyone with at least a working knowledge of BASIC or some other language could easily design and program an ICAI. Remember that each of the variables of expert tutor is independent of the others, thus the selection of the individual variables is up to the designer. The only major dependent function needed to operate the MAIS expert tutor is the Bayesian conditional probability statistic. The Bayesian function sets the parameters of the mastery learning quality control of the MAIS. That is,

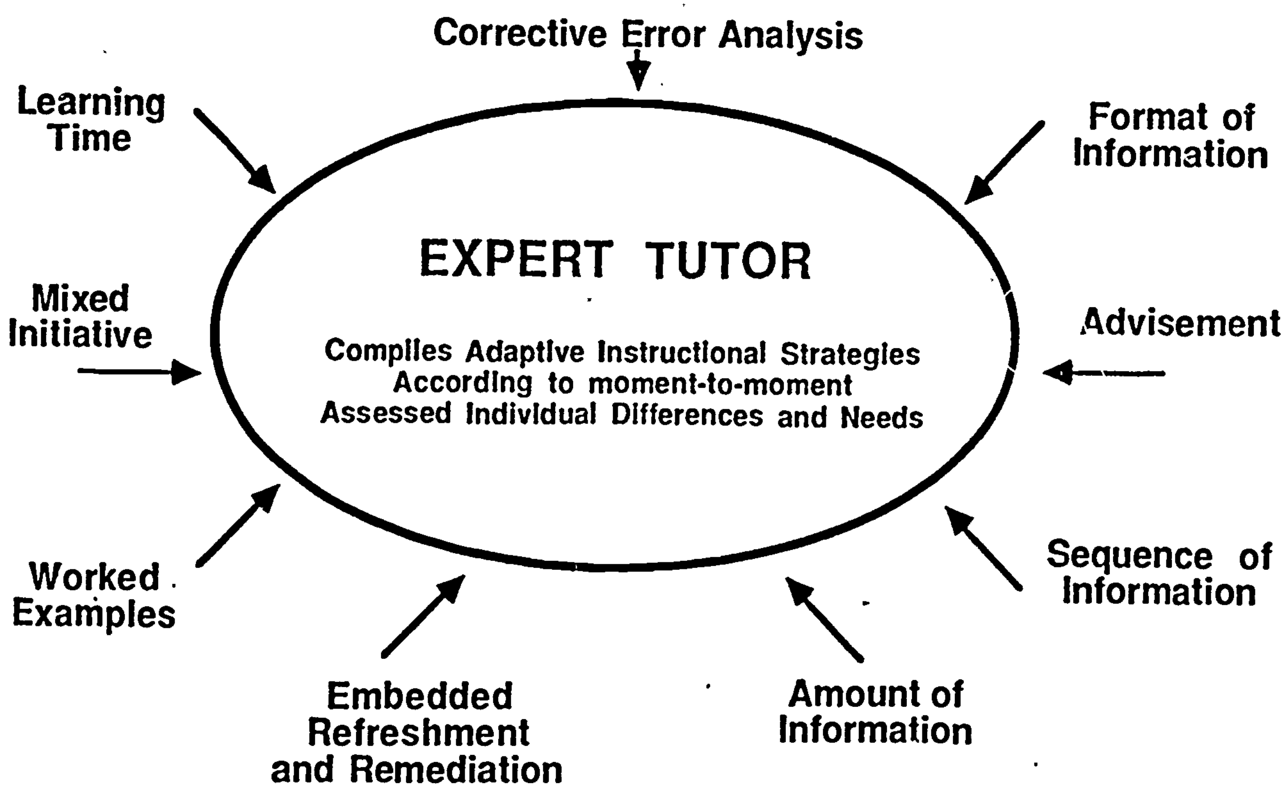

Figure 1. Illustration of instructional variables monitored by the MAIS expert tutor.

the Bayesian provides the information on the decision of whether to advance or retain the learner.

The statistical parameters in the Bayesian method allow the designer to determine the difficulty of the mastery learning decision. In our research we have established a standard format for the three parameters of the Bayesian statistic. Within this paper we will present only this standard format because it uses a heuristic that is very simple to program for use on a microcomputer. Advanced users may want to deal directly with the formula that calculates individual beta value tables. This information is in Tennyson, Christensen, and S. Park (1984).

The computer-based variables of the expert tutor are presented below as subroutines. Copies of an operating disk with sample lesson are available by writing directly to the authors.

Beta Value Computation

The Bayesian subroutine returns a two digit beta value for calculations that are needed in computing amount of information, advisement, and display time interval (Tennyson et al., 1984). The calculations in this standardized subroutine are an approximation of the incomplete beta function. The values from the incomplete beta function with a loss ratio of .3 (this figure is a statistical value in the Bayesian formula and ranges between values of .275 and .325, with the higher values resulting in increasingly conservative control over a false advance versus a false retain), a mastery criterion level of .75 (recall that this figure must include learning error, thus it may seem lower than usual levels for post test mastery learning objectives), and the number of interrogatory examples at 14 (this number could be increased, but should not really be decreased to maintain power of the statistic) is sent to a non-linear regression program that fits the best polynomial. The reason for the polynomial fit is to eliminate the need for calculating the beta value continuously throughout the program (this is certainly possible however on larger mini- and main-frame computers).

INPUT	The only input required is the number of examples correct and the number of examples presented.
OUTPUT	Two place beta value.

Variable List and explanation:

CORRECT	Number of examples that were correct.
PRESENT	Number of examples that were presented. (Note: The code PRESENT^2, means to the second power.)
BETA	Beta value.
C0, C1, C2	Variables used in polynomial

Code:

```
100 C0 = -.385747 + .0507146 * PRESENT - .00328486 * PRESENT^2 +
```

```
    .0000935574 * PRESENT^3
200 C1 + 1.37385 - .2958510 * PRESENT * .02450580 * PRESENT^2 -
    .0007305730 * PRESENT^3
300 C2 + -.273399 + .0767955 * PRESENT - .00725163 * PRESENT^2 +
    .00023077470 * PRESENT^3
400 BETA + C0 + C1 * CORRECT +C2 * CORRECT^2
500 BETA + INT [ABS(BETA * 100 + .5)] / 100
```

Mastery Check and Advisement

This subroutine uses the computed beta value to determine whether a given learner has mastered a given concept or rule (Tennyson & Buttrey, 1984). The mastery decision is used by the expert tutor to make a decision on when to terminate instruction. For learner control situations, the expert tutor advises the learner of his/her progress and recommends an appropriate decision, but allows the learner to decide when to terminate (Johansen & Tennyson, 1984). This subroutine reports to the learner, after each example, his/her current level of mastery regardless of learner control or program control. Note that multiple concepts and rules (coordinate) can also be used in this subroutine. In the following example code, the lesson has four concepts.

INPUT	Beta value Number of examples presented for each concept.
OUTPUT	Booleen statement--mastered or not mastered concept(s). BETA value PRINTED to advise learner of progress.

Variable List and explanation:

BETA(	Beta values from subroutine in array format
MASTERED(	Array to determine concept mastered (1) or not (0)
PRESENT(	Array for number of examples presented in each concept
CONCEPT	Number of concept
MAST	Accumulate mastery of all concepts
EX	Accumulate exhausted pool for concepts

Code:

```
XX10 REM Reset EX and MAST to 0
XX20 EX = 0:  MAST = 0
XX30 REM Mastery check
XX40 FOR CONCEPT = 1 TO 4
XX50 IF BETA(1) > 75 THEN MASTERED(I) = 1:  DONE = MAST = MAST + 1
XX60 IF PRESENT(CONCEPT) > 13 THEN EXHAUST = EX = EX + 1
XX70 NEXT CONCEPT
XX80 REM Determine if all concepts mastered or example pools exhausted
XX90 IF EX + MAST = 4 THEN (EXIT TO END OF PROGRAM)
X100 REM Print advisement
X110 FOR CONCEPT = 1 TO 4
X120 REM Format screen for your desired presentation
X130 PRINT BETA(CONCEPT)
```

X140 NEXT CONCEPT
X150 RETURN

Learning Time Interval

This subroutine monitors and updates the learning time of the interrogatory (practice) examples my increasing the amount of time for correct solutions (Tennyson & Park, 1984, 1985). This MAIS enhancement monitors learning time for two purposes: (a) to provide immediate instructional help if the learner has not yet developed sufficient procedural knowledge; and (b) to prevent the learner from being forced into making an incorrect response (Tennyson, Park, & Christensen, 1985). Monitoring the learning time is not a only a means to improve effectiveness of the instruction, but also to maintain efficiency of the learning environment. That is, time available for learning is a finite variable controlled by both external factors (e.g., school time periods, time of the day, excess to appropriate facilities, etc.) and internal factors (e.g., fatigue, attention, effort, etc.).

Because the parameters of this subroutine include statistical values concerning (a) difficulty of the concept, (b) difficulty of each example, and (c) update in learning progress, it is necessary to establish these values before using the learning time subroutine. In practice, we initially estimate these values and then collect actual times to precisely set the values. (A detailed discussion of these parameters is given in Tennyson, O. Park, & Christensen, 1984).

INPUT	Beta Value
	Example to presented next
	Example Difficulty Index (EDI) (mean time for experts to answer problem correctly)
	Concept Standard (statistical mean time of EDIs)
	Concept Difficulty Index (CDI) (statistical variance of EDIs)
	Lapsed time on example
OUTPUT	Learning time
	Concept Difficulty Index (value added to increase learning time)

Variable List and explanation:

LAPSE	Total elapsed time
LOCAL	Local current example's learning time
CDI(	Array for concept difficulty index
BETA	Beta value
EDI(	Array for concept difficulty index
RESPONSE	Last response 0 = Incorrect; 1 = Correct; 2 = Time elapsed
CONCEPT	Current concept being presented
STANDARD(	Array of concept standard
EXAMPLE(	The specific example to be presented next

Code:

```
XX10 REM Check learning time against elapsed time
XX20 IF LAPSE > LOCAL THEN RESPONSE = 2
XX30 REM Concept difficulty index subroutine
XX40 CDI(CONCEPT) = CDI(CONCEPT) + BETA * STANDARD(CONCEPT)
XX50 REM Learning time subroutine
XX60 LOCAL = EDI(EXAMPLE) + CDI(CONCEPT)
```

For example, if the total set of examples for a given concept has a mean value of 17 sec. (STANDARD), and a variance of 2.5 sec. (CDI), and a current beta value of .55, the calculation for the concept difficulty index would be the following:

CDI(Concept) = 2.5 + .55 * 17
CDI(Concept) = 11.85

For the next example (if current example correctly answered and an EDI value of 15 sec.), the learning time value would be increased as follows:

LOCAL = 15 + 11.85
LOCAL = 26.85

This heuristic allows for an iterative learning time increase with each succeeding correct response.

Formal of Examples and Sequence

This subroutine selects the format of the next example according to the response given to the current example, as follows: if correct or if time elapses, the next example will be in an interrogatory format; if incorrect, it will be presented as an expository example (Park & Tennyson, 1986). Also, this subroutine selects the sequence of the next example according to, first, the generalization rule (usually for the first four interrogatory examples) and, second, the discrimination rule (usually starting with the fifth example (Park & Tennyson, 1980). This subroutine also determines that no example is presented more than once and that no example is presented if the example pool is exhausted.

INPUT	Last response
	Concept presented and selected
	Number of examples for each concept presented
OUTPUT	Sequence of next concept and format of example

Variable List and explanation:

RESPONSE	Last response, 0 = Incorrect; 1 = Correct; 2 = Time elapsed
ANSWER	Last concept selected
EXAMPLE	Number of example selected
SEQUENCE	0 = Generalization; 1 = Discrimination

MASTERED(Array to determine concept mastered (1) or not (0)
PRESENT(Array for number of examples presented in each concept
CONCEPT Number of concept

```
XX10 REM Select example number from pool
XX20 EXAMPLE = INT(RND(1) * 14 + 1
XX30 REM First four examples presented in each concept are generalization
XX40 IF SEQUENCE = 1 AND PRESENT(CONCEPT) < 5 THEN SEQUENCE = 0
XX50 IF SEQUENCE = 1 AND PRESENT(CONCEPT) > 4 THEN SEQUENCE = 1
XX60 REM When response is incorrect or time elapsed and generalization is in
     effect then concept to be selected remains the same
XX70 IF RESPONSE = 0 OR (RESPONSE = 2 AND SEQUENCE = 0) THEN GOTO X120
XX80 REM If response is incorrect and discrimination then next concept to be
     selected is the learner's incorrect response
XX90 IF RESPONSE = 0 THEN CONCEPT = ANSWER: GOTO X120
X100 CONCEPT = INT(RND(1) * 4 + 1)
X110 REM Determine example pool exhausted / If exhausted then response must be
     changed to correct so random select of concept occurs
X120 IF PRESENT(CONCEPT) > 13 THEN RESPONSE = 1: GOTO XX10
X130 REM If response is correct and concept not mastered then start over
X140 IF RESPONSE = 1 AND MASTERED(CONCEPT) = 0 THEN GOTO XX10
X150 REM If example was used before start over
X160 IF SELECTED(CONCEPT,EXAMPLE) = 1 THEN GOTO XX10
X170 SELECTED(CONCEPT,EXAMPLE) = 1
X180 RETURN
```

The variables of corrective error analysis and embedded refreshment and remediation are task-specific enhancements that are designed at the point of individual lesson development. The important concept for the former variable is to consider the type of analysis as a function of the instructional strategy to be employed. For the latter variable, the design decision comes from the structure of the content to be learned. Both of these variables need attention so as to provide adequate instructional help, but not to the point of reducing efficiency of the learner. For example, too much interference from adjunct instruction can distract the learner and consequently use up valuable learning time.

Summary

Our purpose in this paper was to present the program code of the heuristics employed in the MAIS. The MAIS program is supported by both learning theory and instructional theory. Also, the instructional variables and conditions of the MAIS are supported by empirical verification; tested in a well-defined program of research, and evaluated by disciplined peer review. The value of the theory-based instructional design system supported by direct research findings is that it can be generalized to specific learning needs and conditions. And, for implementation purposes, the MAIS is readily transferable to most currently available hardware and software. And, as computer technology itself improves, it will be possible to both enhance the present variables and conditions yet to be discovered. Some of these new variable will come from research in such diverse areas as individual differences, human-machine interface design, neuropsychology, psychometrics, computer software, perception, and the continuing significant research and theory development in the field of instructional technology, curricular management as well as hardware and software developments.

References

Dorner, D. (1983). Heuristics and cognition in complex system. In R. Groner, M. Groner, & S. F. Bischof (Eds.), Methods of heuristics. Hillsdale, NJ: Lawrence Erlbaum Associates.

Johansen, K. J., & Tennyson, R. D. (1984). Effect of adaptive advisement on perception in learner controlled, computer-based instruction using a rule-learning task. Educational Communication and Technology Journal, 31, 226-236.

Park, O., & Tennyson, R. D. (1980). Adaptive design strategies for selecting number and presentation order of examples in coordinate concept acquisition. Journal of Educational Psychology, 72, 499-505.

Park, O., & Tennyson, R. D. (1986). Response-sensitive design strategies for sequence order of concepts and presentation form of examples using computer-based instruction. Journal of Educational Psychology, 78.

Tennyson, R. D., & Buttrey, T. (1984). Advisement and management strategies as design variables in computer-assisted instruction. In D. F. Walker & R. D. Hess (Eds.), Instructional software. Belmont, CA: Wadsworth.

Tennyson, R. D., Christensen, D. L., & Park, S. I. (1984). The Minnesota adaptive instructional system: An intelligent CBI system. Journal of Computer-Based Instruction, 11, 2-13.

Tennyson, R. D., & Park, O. (1987). Artificial intelligence and computer-assisted learning. In R. Gagne (Ed.), Instructional technology: Foundations. Hillsdale, NJ: Lawrence Erlbaum & Associates.

Tennyson, R. D., Park, O., & Christensen, D. L. (1985). Adaptive control of learning time and content sequence in concept-learning using computer-based instruction. Journal of Educational Psychology, 77, 481-491.

Tennyson, R. D., & Park, S. I. (1984). Process learning time as an adaptive design variable in concept learning using computer-based instruction. Journal of Educational Psychology, 76, 452-465.

Tennyson, R. D., & Park, S. I. (1985). Interactive effect of process learning time and ability level on concept learning using computer-based instruction. Journal of Structural Learning, 8, 241-260.

Tikhomirov, O. K. (1983). Informal heuristic principles of motivation and emotion in human problem solving. In R. Groner, M. Groner, & W. F. Bischof (Eds.), Methods of heuristics. Hillsdale, NJ: Lawrence Erlbaum Associates.