DOCUMENT RESUME

ED 307 196 SO 019 901

AUTHOR Schaffer, John William
TITLE Microcomputer-Based Intelligent Tutoring Systems: An Assessment.
PUB DATE 88
NOTE 18p.
PUB TYPE Reports - Descriptive (141)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; *Expert Systems; Higher Education; *Instructional Innovation; *Microcomputers; *Music; *Music Education; Teaching Methods
IDENTIFIERS Harmony; Harmony Coach (Computer Software); *Intelligent Tutoring Systems

ABSTRACT

Computer-assisted instruction, while familiar to most teachers, has failed to become an effective self-motivating instructional tool. Developments in artificial intelligence, however, have provided new and better tools for exploring human knowledge acquisition and utilization. Expert system technology represents one of the most promising of these developments. Expert systems are computer programs that store human-like knowledge. Methods of expert system design are discussed, as are ways of computing propositional and predicate logic. "Harmony Coach" (written in Turbo Prolog) is an intelligent tutoring system that uses the microcomputer environment (IBM PC, XT, AT) to coach the specific musical tasks of writing tonal harmonic progressions, aiding the student in the creation of an acceptable harmonic sequence based on either an unfigured bass line or on a melody. It also coaches students in part-writing any previously harmonized exercises using the traditional four-part chorale model. The "Coach" is designed around the following premises, most of which are essential to any intelligent tutoring system: it must be based on sound musical precepts; it should operate in a passive role; hardware aspects should be transparent to the end user; it should use artificial intelligence whenever possible; it should embrace knowledge, explanative understanding, and problem-solving abilities; and it must run under normal restrictions of a high-end microcomputer learning environment. These precepts are explored in detail. Strengths and weaknesses of the program are discussed and a 7-item bibliography is included. (GEA)

Microcomputer-Based Intelligent Tutoring Systems: An Assessment

By

Dr. John Wm. Schaffer
University of Wisconsin-Madison

Introduction

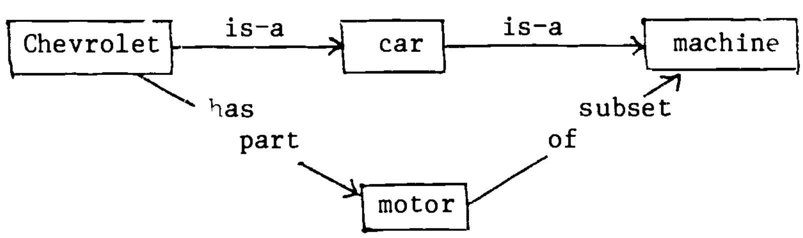
Since the first commercial microcomputers appeared in the late 1970s, pedagogues have utilized them in various aspects of teaching. Computer-assisted instruction has become a concept familiar to most, if not all, teachers in nearly every level of education ranging from nursery school to college. Unfortunately, CAI has failed to live up to its potential as an effective self-motivating tool. First, the limited computational power of early microcomputers with their restricted memory space and processing capabilities proved insufficient for developing instructional materials able successfully to transcend rudimentary 'right vs. wrong' pedagogical strategies. This concept worked well for assisting a student in developing and refining quantifiable skills. However, since straight drill-and-practice tended to aim at the refinement of previously learned skills and not the acquisition of new knowledge, it often lacked effective motivational incentives derived through self-exploration and guided learning. Second, the complexities necessary to emulate human thought processes and conceptual understanding in a computer program are only now beginning to be successfully unraveled.

Developments in the field of artificial intelligence, however, have given us new and better tools with which to explore human knowledge acquisition and utilization. Expert system technology, while no longer the 'cutting edge,' represents one of the most promising of these developments. Simply stated, an expert system is a computer program that stores human-like knowledge in a computer. As a point of clarification it will be helpful to discuss the main methods by which expert systems and intelligent tutoring systems express their knowledge.

The following list, drawn from the excellent works of William Gevarter and Donald Waterman,[1] identifies the most significant schemes of knowledge representation:

1. Semantic Networks are a method of representing knowledge by describing the properties and relationships of concepts, objects, events, etc., as nodes interconnected by lines, or links, that describe their relationships (Figure 1).

Figure 1



2. Procedural Representations (Production Rules) are small programming procedures that act out, or represent, a logic relationship. Figure 2a represents a production rule using predicate logic to define a relationship (in this case that John is the father of Allison). Figure 2b is a logic clause that infers a relationship, while figure 2c represents a production rule in which resultant action will depend on one or more conditions.

Figure 2

a) father(JOHN,ALLISON)

b) yesterday(Sunday) and tomorrow(Tuesday)
implies that today(Monday).

c) IF the chord belongs to a cadence
AND the chord is a dominant
AND it is the last chord in the phrase
THEN continue analyzing the composition.

3. Analogical or Direct Representations are simply arrays, or data bases, that contain complete images or ranges of variable values as a mechanism for making comparison references. For example, a particular scale might be used as a reference for all available pitch materials allowed in a given segment of music.

4. Property Lists represent a means of codifying all the characteristics of a particular object. Figure 3 presents a property list for a root-position major triad.

Figure 3

Root Position Major Triad:

-three different pitches
  -one pitch is the root
  -one pitch is the third
  -one pitch is the fifth
-the third is a Major 3rd above the root
-the fifth is a Perfect 5th above the root

5. Frames and Scripts are a more complex variation of the property list concept. A frame is a complex data structure that represents a stereotyped object, event, situation, etc. In a typical frame some of the 'slots' are filled (representing known facts) and some are empty (waiting to be filled as knowledge is acquired based upon preexisting bounds). Figure 4 represents a frame describing an automobile.

Figure 4

FRAME: CAR

type: range(sedan, station wagon, hatchback,...)
manufacturer: range(Chevrolet, Ford, Toyota,...)
weight: range(1,000 lbs. to 4,000 lbs.)
engine size: range(4, 6 or 8 cylinders)
fuel type: range(gasoline, diesel, propane)
etc...

Only two of the rather large number of methods (from which these five examples are drawn) have proven significant in expert system design. According to Peter and others these two are semantic networks and productions rules.[2]

Within each of these knowledge representation schemes there are several different methods for computing the logic. The first type, known as propositional logic, is based on simple true or false statements using standard logic operators such as AND, OR, NOT, IMPLICATION, and EQUIVALENCE. Propositional logic is somewhat limited in that it can only deduce true or false answers. The second scheme, based on predicate logic, overcomes this limitation by allowing the logician to make assertions about objects within a statement, and allows for the use of variable substitutions within clauses. For example, with the statement in figure 2a of 'father(JOHN,ALLISON)', propositional logic can give only a true or false answer based on whether John was or wasn't the father of Allison. On the other hand, predicate logic allows for the substitution of names with variables. The same clause written in predicate logic as 'father(x,ALLISON)' no longer requires a true or false answer. Instead, the clause is satisfied by finding the correct value for the variable 'x.' Thus, the solution to the clause is not true or false, but is 'JOHN.' Since most expert systems and intelligent tutoring systems use symbolic knowledge representation, predicate logic, with its inherent ability to manipulate symbols, is generally the representation of choice.

As an aside, any study of how the mind works must necessarily study the way in which the mind learns as well (i.e., attains and retains knowledge). In his discussion of artificial intelligence Patrick Henry Winston comments on an interesting theory, speculating that,

> . . . One result . . . may be new ideas about how to help people become more intelligent. Just as psychological knowledge about human information processing can help make computers intelligent, theories derived purely with computers in mind often suggest possibilities about methods to educate people better. Said another way, the methodology involved in making smart programs may transfer to making smart people.[3]

Since expert systems represent a potentially ideal means of storing and retrieving knowledge in an intelligent manner, and since knowledge acquisition and understanding are two of the most important aspects of education, it seems logical that pedagogues should take a keen interest in the development and utilization of this technology. While the desire to use and apply computers in teaching has long been felt and practiced, the previous lack of effective artificial intelligence techniques has severely limited developments in this area to a relatively small number of experimental models. With the advent of expert system technologies, however, the concept of an Intelligent Tutoring system (in other words, a system that tutors with the aid of an 'intelligence-based' knowledge structure) is rapidly reaching fruition.

Much has been written on the subject of CAI and many, if not most, recent writers call for the need for more intelligent computer teaching approaches. However, with the exception of a small handful of scholars, very little of any specific research has been undertaken with regard to the application of artificial intelligence techniques to the design of music-oriented instructional software.

The Programs:

The understanding of many music theoretical premises can often be quite elusive. The rules we create to explain music often appear to a student as representing quantifiable information. Therefore these rules might appear to be ideally suited to a simple drill-and-practice lesson. The ways in which these skills are applied, however, often appear to be much more elusive, requiring varying levels of insight, reason, logic, and intuition. For this reason alone an Intelligent Tutorial has the potential to aid significantly in bridging the gap between simply drilling a student in the 'how' of an objective skill and intelligently tutoring the student in the more abstract 'why' of the application of that knowledge.

HARMONY COACH is an amalgamation of three interrelated coaching modules originally developed to demonstrating the feasibility of creating such a system, not within the confines of a powerful limited-access mainframe or minicomputer, but within a widely-accessible and relatively inexpensive microcomputer environment. Two of the three program modules coach the specific tasks of writing tonal harmonic progressions, aiding the student in the creation of an acceptable harmonic sequence based on either an unfigured bass line or melody.[4] The third program module coaches the student in part-writing any of their previously harmonized exercises using the traditional four-part chorale model.

Although the initial experimental nature of the these modules, together with the inherent size and power limitations of microcomputer-based implementations in general, placed numerous restraints on their overall scope, the tutorials do contain a sufficiently large knowledge base capable of dealing adequately with all basic formulations relating to non-modulatory common-practice harmony. In this case, the scope of the program modules are purposely constra ed within a well defined domain in order to maximize their coaching effectiveness without losing the inherent flexibilities of a microcomputer environment. For similar reasons the use of passive coaching strategies, as opposed to more active tutorial methods, conserves memory resources, which, in turn, permits greater flexibility in the development of the actual knowledge base.

The actual Tutorial (hereafter referred to as "the Coach") is designed around six specific premises, most of which are essential to nearly any Intelligent Tutoring System:

1. The Coach must be based on sound musical precepts.
2. The Coach should operate in a passive role, in which the student is given the flexibility to learn through creative self-exploration.
3. Hardware aspects of the Coach should be as transparent as possible to the end user.
4. The Coach should use Artificial Intelligence techniques wherever appropriate to attain as intelligent and flexible an environment as possible.
5. The Coach should embrace the three elements primary to any good Intelligent Tutorial: knowledge, explanative understanding, and problem solving ability.
6. The Coach must be able to run under the normal restrictions of a typical high-end microcomputer learning environment.

At this point it essential to examine each point in greater detail.

1. The Coach must be based on sound musical precepts. The best an expert system can presently achieve is to recreate human logic and understanding in a manner that is 'perceived' to be true. In other words, since an expert system is not capable of creating its own truth without prior knowledge of a domain it cannot effectively judge the rightness of the information imparted to it. Therefore, an expert system can be no smarter than the logic and knowledge that is given to it. Its ability, therefore, must be judged on the basis of how well it "emulates" our human logic, no matter how fallible that may be. Human fallibility being a given, however, it is important that a knowledge base represent the collective knowledge of more than one expert, taking into account both the cumulative strength of mutual agreement, as well as the weakened state of discrepancy. As a point of departure, the knowledge structure in the Coach is based on information derived from a representative body of human tutors--in this case Theorists and the textbooks they write and teach with.

2. The Coach should operate in a passive role, in which the student is given the flexibility to learn through creative self-exploration. Sleeman and Brown, in the introduction to their book of collected essays on the subject of intelligent tutoring systems, discuss this idea and speculate on some guidelines for successful systems.

> In the last five years researchers have focussed on supportive learning environments intended to facilitate learning-by-doing: transforming factual knowledge into experiential knowledge. These systems attempt to combine the problem-solving experience and motivation of 'discovery' learning with the effective guidance of tutorial interactions. These two objectives are often in conflict since, to tutor well, the system must constrain the student's instructional paths and exercises to

those whose answers and likely mistakes can be completely specified ahead of time. To overcome these limitations, the system must have its own problem-solving expertise, its own diagnostic or student modelling capabilities and its own explanatory capabilities. In order to orchestrate these reasoning capabilities it must also have explicit control or tutorial strategies specifying when to interrupt a student's problem-solving activity, what to say and how best to say it; all in order to provide the student with instructionally effective advice. . . . by augmenting open-ended, problem solving environments with the above sort of tutorial intelligence, it becomes possible to transform a student's conceptual flounderings and misconceptions into profound and efficient learning experiences -- ones rooted in his own actions and hypotheses. In brief, the augmentation of environments with intelligent tutoring enables more students' misconceptions to be transformed into constructive experiences.[5]

The design of the Coach aims at stressing precisely this type of interaction. It would be misleading, however, to imply that the Coach achieves such ideal goals. Current microcomputer systems have neither the memory nor the processing power to endow the Coach with all the necessary tools. Essentially, the Coach allows the student to complete a task by supplying help and guidance when requested but imposes no particular strategy or impetus. Thus, the student is free to approach any task with or without being coerced into any preconceived notions about how to achieve the desired goal, while simultaneously forcing students to undertake a greater self-role in their own learning process.

3. Hardware aspects of the Coach should be as transparent as possible to the end user. For a tutorial to attempt to emulate a human tutor it must strive to divorce itself from those things that are distinctly non-human, i.e., the computer and its associated use and paraphernalia. For this reason the Coach is designed in a perceptually simple, straight-forward manner with a minimum of cosmetic 'bells-and whistles.' Most interaction adopts the now relatively standard 'point-and-choose' menus and all music is displayed in graphics using traditional notation. While the Coach may lack certain initial 'glitz,' it tends to focus the student's attention more on the task than on the means.

4. The Coach should use Artificial Intelligence techniques wherever appropriate to attain as intelligent and flexible an environment as possible. The use of expert system technology is central to the design of the Coach, since an expert system offers a relatively established method of coding a domain of human knowledge, while simultaneously imposing no strict sequencing of logic and understanding. It is specifically these properties of a suitably designed expert system that allow students to tackle problems in their own manner. Every time the student proposes a choice, the rule base examines its facts and relationships and draws a conclusion based on what it knows in relation to both the information contained in the knowledge base as well as information previously imparted through the student's previously acceptable choices. In other words, each time the student asserts a correct idea, that information is inserted in the knowledge base, thus potentially affecting later decisions.

The knowledge base itself is designed around a three-tiered heuristic model. The highest level defines ten general rule categories relating to the normative conventions of harmonic progressions. Categories covered include:

1. Large-scale harmonic motion,

2. general chord classifications,

3. root progressions,

4. first inversion chords,

5. the cadential I6/4,

6. general seventh chord basics,

7. dominant sevenths,

8. other sevenths,

9. harmonic rhythm, and

10. harmonic successions.

This highest level of rules in most cases attempts to be non-chord specific, instead generalizing on classes of chords. Figure 5 shows a typical subset of rules extracted from the general category dealing with first-inversion chords.

Figure 5

RULE 4a: a progression from any root-position chord to the same chord in first inversion is <u>acceptable</u>.

RULE 4b: a progression from any first-inversion chord to the same chord in root position is acceptable, but not <u>preferred</u>.

RULE 4c: a progression from any first inversion chord to a chord whose bass note lies stepwise above or below the current bass note is acceptable and <u>desirable</u>.

Once a specific chord is assigned to a chord class, the primary function of the middle level of the knowledge base is to coordinate which general rules should be applied and tested. The Coach uses the four class categories determined by Allen Winold in his recent text <u>Harmony: Patterns and Principles</u>.[6] Specifically, these classes are Tonic, Dominant, Subdominant, and Linear. Figure 6 shows an example of a middle-level rule:

Figure 6

RULE 11.Ta: A root-position Tonic-Class triad may harmonize the given base note...

IF a chord moves to another chord with the same root AND rules...
(2.1a [Tonic-class chords may move to chords of any class.] AND 4a [a progression from any root-position chord to the same chord in first inversion is acceptable.] AND 9 [two adjacent chords sharing the same root must not be separated by a bar line]. AND 1a [the last chord of a progression must be a tonic, dominant, or submediant.] are observed...)
OR two chords are not the same AND rules...
(2.1a [Tonic-class chords may move to chords of any class.] AND 1a [the last chord of a progression must be a tonic, dominant, or submediant.] are observed.)

[(same chord AND 2.1a AND 4a AND 9 AND 1a)
OR (NOT(same chord) AND 2.1a AND 1a)]

Finally, the lowest level heuristics deal with consultation decisions based on the location of a specific chord choice within the context of a given exercise. For example, regardless of chord classification (i.e., tonic class, dominant class), the positioning of a particular chord within a cadence, as opposed to elsewhere in the progression, will significantly effect which rules need be consulted. Additionally, the heuristics at this level must take into account previous successful chord choices. Obviously, the selection of a particular chord at a given position within a progression will be more restrictive if an adjacent position is harmonized, as opposed to one that is not. Since the Coach allows for harmonization in any order, it cannot assume that each chord choice will have an immediate successor or predecessor present in the progression. The Coach deals with this in a retrospective manner. Whenever a new chord is inserted into the exercise, existing chords on either side of the entry are rechecked, now taking into consideration the newest entry, thus insuring that all previous choices are updated in relation to any new information in the knowledge base. If a previously acceptable choice now becomes problematic in relation to the new choice the student is informed of the problem and has the opportunity to correct it by changing either or both of the offending harmonies.

5. The Coach should embrace the three elements primary to any good Intelligent Tutorial: knowledge, explanative understanding, and problem solving ability. First, an intelligent tutorial must embody a sufficient amount of knowledge to accomplish acceptably, by itself, any domain-specific task that might be posed by a user. It is precisely this ability that equips the tutorial to respond to any methodological strategy, regardless of the present state of the tutorial "universe," at least as long as students remain within the domain of the knowledge base. Since the memory constraints of current microcomputers pose limitations on the size of an intelligent tutorial, the knowledge domain of the Coach is specifically confined to allow it to remain comprehensive within its domain. Second, a tutorial must be able to explain the logic behind any response to a student's query. Actually, this information is inherent in any well designed knowledge base. Although the Coach has the technical capability to implement this aspect, the overall effectiveness is restricted by the lack of intensive interactive language capabilities. At present, creating and interpreting the English language represents an immense task. An average microcomputer would be incapable of supporting a tutorial based on anything more than a prohibitively small knowledge domain if it needed to be coupled with the ability to communicate even moderately effectively through comprehensible English.

Finally, an intelligent tutorial must be able to utilize the domain knowledge to work through any relevant problem solving process in order to evaluate effectively and coach a student. The Coach is able to use its predefined knowledge base, together with a student's prior input, to determine all available options, as well as to explain the logic behind any given response.

6. The Coach must be able to run under the normal restrictions of a typical high-end microcomputer learning environment. While it is certainly desirous to aim at achieving state-of-the-art implementations with the most powerful and flexible equipment possible, the reality of attaining these lofty goals is very unlikely in the foreseeable future. With respect to hardware considerations, this project was designed with three specific criteria in mind: 1) that it be implement on a relatively standardized, commercially available, and accessibly priced machine, 2) that it use a language that is both compilable and similarly accessible, and 3) that it utilize expert system strategies as much as possible within the previous constraints.[7] Fortunately, given the tutorial and domain constraints discussed earlier, these three criteria have been met. The Coach is currently implemented on an IBM-AT clone with EGA graphics, a hard disk, and 640k of main memory. The actual program, based around a knowledge base structured with predicate logic and utilizing backwards chaining and depth-first search heuristics, was completely written and compiled to machine code in Borlands Turbo Prolog 2.0.[8] It is hoped that an Apple Macintosh version of the Coach will be available within a year.

Conclusions:

Since matters of program design are highly personal, often subjective, and in this case represent new technology requiring new design criteria, it is difficult to assess the Coach from a purely objective point of view. With varying levels of success the program embodies the foundations and mechanisms for all the conditions set forth above. It also demonstrates the feasibility of successfully applying, within a traditionally limited microcomputer environment, those elements of an intelligent tutorial system presently deemed desirable. Clearly, since the Coach was originally conceived as a test vehicle. future changes and improvements are inevitable. the most notable being the Coach's response mechanisms. Most problematic is the ability to respond intelligently to specific student queries in grammatically correct English. Currently. the Coach can present rules and relationships in order to show the logic behind program judgments and indicate various generic responses designed to cover most normal situations. However, it cannot always respond in an intelligent manner to unusual situations, nor can it enter into a meaningful dialogue in an effort to understand the nature of a specific problem a student may be struggling with. Since the Coach's knowledge base is designed primarily to teach the norms of harmonic progressions and simultaneously is dealing with student's whose level of understanding is usually significantly below that represented in the Coach, its inability to deal with the few rare anomalous situations does not represent much of a weakness. For the most part the knowledge base designed into the Coach is actually quite comprehensive. In fact. one common pastime of the more advanced students is to actually go head-to-head in challenging the Coach's wisdom by purposely proposing anomalous. or even incorrect. information. Usually the Coach catches them.

Because of the new and experimental nature of Intelligent Tutoring Systems, much work still needs to be done. It is essential to explore how well this technology can represent and convey domain-specific knowledge. Of equal importance (but unfortunately beyond the scope of this report) is the exploration of just how effective these intelligent tutoring machines will prove to be. If one conclusion can be drawn from this work it must be this:

that the technology to build intelligent mini-tutorials for use on microcomputers is ready and waiting to be applied. Unfortunately, the knowledge of how to utilize it is, for the most part, time consuming and difficult to obtain. Programming complex intelligent tutorials is simply beyond the capabilities of the average weekend programmer. If our teaching is to advance with the aid of these programs then it is essential that support be given to allow our graduate students and scholars to pursue this research in a productive and profession manner.

Endnotes

1.Gevarter, William B. Artificial Intelligence, Expert Systems, Computer Vision and Natural Language Processing. Park Ridge, NJ: Noyes Publications, 1984, pp. 17-20; and Waterman, Donald. A Guide to Expert Systems. Reading, MA: Addison-Wesley, 1986, pp. 63-79.

2.Sell, Peter S. Expert Systems: A Practical Introduction. New York: John Wiley & Sons, 1985, p. 33.

3.Patrick Henry Winston, Artificial Intelligence, 2nd ed. (Reading, MA: Addison-Wesley Publishing Company, 1984), p. 2.

4.The eight levels of graded bass and soprano lines utilized in the programs are derived from chapters 8-17 of Allen Winold's Tonal Harmony (Prentice Hall, 1987).

5.D. Sleeman and J. S. Brown, eds., Intelligent Tutoring Systems (New York: Academic Press, 1982), pp. 1-2.

6.Allen Winold, Harmony: Patterns and Principles, Vol. 1, Englewood Cliffs: Prentice-Hall, 1986.

7.The three modules are written in Turbo Prolog and are designed to run on any IBM PC, XT, AT or similar clone with a minimum of 512K of RAM memory, two disk drives (or one hard disk), and a CGA and/or EGA graphics adaptor.

8.Turbo Prolog is a registered trademark of Borland International.