THE COMPETENT COMPUTER

Roger T. Hartley
Department of Computer Science
Kansas State University
September 1981

Abstract

The emergence of expert system methodology as the cutting edge of the application of artificial intelligence techniques has tended to refocus the issues of the nature of intelligence. Expert systems are clearly not intelligent in the meaning espoused by early workers in AI. Indeed, they are intended to be transparent in operation to the user. However, they are competent in their domain of application. This notion of competence and of a competence model turns out to be an appropriate conceptual analysis for much of the early efforts in AI, as well as current work in the field.

1. Introduction

Artificial Intelligence has traditionally been subject to major dichotomies of interest, leading to opposing methodologies. Some of these are esoteric, such as the procedural-declarative controversy [11] knowledge versus power in representation [2]; algorithmic-heuristic differences and the age-old split between simulation and synthesis. However, it seems that the field has never before been so widely polarized as it is by the move, on the one hand, deeper into theory with cognitive science, and into practice with expert systems. It is, of course, in the very nature of the development of science that changes like this should occur, so our job is not to oppose them or deny their existence, but to attempt to understand why they are occurring and what the next stage of development might look like.

In this paper I attempt to explain the current changes by re-interpreting some early work in AI and show how minor differences occurring there are magnified now through increased knowledge and understanding. The approach is almost entirely due to Les Johnson 13,41, but the detailed thinking is all mine. The central issue is that the word "intelligent" is almost certainly the wrong word to focus on when considering the programming of computers to exhibit complex, sophisticated behaviors. Indeed the word is being abused by the computer industry with the advent of "intelligent" cables and similar semantic atrocities. However, we do have a word which, although used in a technical sense by some people (notably Chomsky) seems to be a good choice as an attribute of suitably programmed computers. The word is competent. This paper will use the word in a non-technical sense. If this usage turns out to be correct, then I have succeeded. Even a partial success can, I think, lead to greater insight into some very difficult problems.

2. Competence And Competence Modeling

2.1 If a person, having learned a skill, performs various tasks which utilize that skill regularly and satisfactorily then we say that person is competent. Repetition is an essential feature of competence, as is a reasonably wide application and a relatively high level of sophistication. (We all walk regularly, on different surfaces at different speeds, but we do not call ourselves competent walkers.) There is, however, a feature of competence which makes it a member of to avoid bad moves. The criteria for making a move will be based on heuristic considerations pulled from a variety of sources, both experimental (always place your Rook on an open file) and theoretical (King and Queen always beats King and Rook in an end-game). If we carry out this procedure, as has been done to great effect many times, what status does the end product have? Is it a model of a particular chess player's skills? Probably not. Is it a model of human chess skills in general? Certainly not. If it were, computers would now be beating individual grandmasters regularly. Is it a model of that part of human intelligence devoted to game playing or even to chess in particular? This is getting closer, but many features of chess programs are most definitely non-human; in particular, they typically evaluate hundreds of thousands of possible moves - a feat beyond all human cognitive capabilities. What then does the chess program represent? In our view, what it does do is present a model of chess competence. This is not necessarily human chess competence, but by making explicit (even transparent) the rules by which it operates, it says to the interested observer "If you want to play chess well here is one way to do it." Because the programmed computer is competely mechanistic in nature A= human beings have a firm belief that, given enough time and resources, they could take the place of the machine and run the program themselves, then this hypothetical statement by the machine is entirely meaningful to us. Notice that the model of competence presented might turn out to be a bad one - the criteria under which it operates might be badly chosen, usually through lack of understanding. However, the aim of the programmer was undeniably to play good chess, not to represent human frailties when playing chess (that would be simulation modeling, not AI).

2.3 If we attempt now to generalize the chess example, the same aim of competence modeling can be seen to apply to almost all areas of classical AI, whether it is problem solving, natural language, vision, robotics, knowledge representation, inference-making or learning. The next section deals with some of these other fields and gives examples from AI work in the past. I am not here attempting to claim that these researchers thought they were doing X but actually they were doing Y. The aim is to show how a re-interpretation of the end-products of their research can lead to an understanding of current AI, and in particular, expert systems. This re-interpretation relieves us of the seemingly impossible problem of whether these programs display intelligence or not. The issue of whether a methodology exists which aims at the intelligent computer is left to the end of the paper. The main aim is to show that a competent computer methodology is just around the corner.

3. Artificial Intelligence A= Competence Modeling

3.1 No apologies are offered for the catalogue of classic programs mentioned here, except that they are all rather overexposed by now. Examples are taken from problem solving, natural language, knowledge representation, and learning. Other sub-fields can be subjected to the same analysis, as can other examples within each sub-field.

3.2 Problem solving - STRIPS

3.2.1 STRIPS [5,6] is perhaps the classic example of problem-solving-as-theorem-proving. The main features are:

1. Use of predicate calculus as the universal representation of problem domain and solution set.

2. Use of the idea of the state of a problem and hence of operators defining transformations from one state to another.

3. Use of means-ends analysis implemented through heuristic search rules to guide the theorem prover.

3.2.2 The first point to note is that all three features are heavily theoretical in nature. Even means-ends analysis, demonstrated experimentally to be a common strategy used by human problem solvers is here constrained by the use of predicate calculus representations and the need to program the heuristic rules for the deterministic computer. There can be no question, therefore, that STRIPS is a simulation - it certainly is not. Is it then a presentation of a theory of human problem solving? This also is unlikely because most humans are inadequate at manipulating logical statements but yet can still solve problems. They can certainly solve the "Push Box A Next to Box B" sort of problem that STRIPS was set to do. We are thus left with the competence model. STRIPS tells us how we ought to solve such problems and demonstrates its criteria of competence by actually coming up with decent solutions to a wide enough range of problems.

3.2.3 Unfortunately, it is quite easy to demonstrate that STRIPS is not completely competent as a problem solver, especially as complexity of the domain increases. It makes silly mistakes, not all of them eliminated by the later modifications in ABSTRIPS 171. As a model of competence in problem-solving, we can thus say that STRIPS is inadequate. It also fails when presented with a new problem domain the setting up needed is lengthy and difficult.

3.3 Natural Language - SHRDLU

3.3.1 Terry Winograd's program [8], often cited as the classic AI program, can also be best viewed as a model of competence. Its main features are:

1. Interactive and conversational.

2. Can accept a wide variety of ordinary English commands, questions and statements, all of which provoke an appropriate response.

3. Can keep track of the dialogue in a limited sense, through backward reference.

4. Can solve problems using deductive inference methods.

5. Can explain its own behavior by keeping track of its working.

3.3.2 Again, as with STRIPS, SHRDLU operates through Winograd's thorough theoretical understanding of its domain (the so-called "blocks world") and of the language needed to talk about that domain. Again it is clearly not a simulation. Perhaps SHRDLU, then, gets close to being

• presentation of a theory of language understanding? This is undeniably

• crucial part of human intelligence - the capacity to generate and use

• symbol set through the abstract idea of a language. However, to me at least, it presents an extraordinarily constrained view of human understanding, if this is the case. Its heavy reliance on deductive inference-making, on an imcomplete system of semantics and an over-simplified method of problem-solving, seems to rule it out as a valid theory. There is too much experimental evidence which runs contrary to it, both in psychology and linguistics.

3.3.3 However, all of these objectives become virtues if SHRDLU is looked at as a competence model in the blocks world and any essentially similar domain, of which there are many. In fact, SHRDLU can interpret far more complicated sentences and solve more complicated problems than most humans can, because it is a blocks world expert. It really is competent at pushing blocks around! I feel that it is this aspect which has impressed many people, not that it presents a good theory. SHRDLU tells us how to be better at manipulating blocks on a table-top using only one hand.

3.4 Knowledge representation - FRL

3.4.1 Although FRL (9] is essentially an oversimplified account of how knowledge is organized (as the authors admit); it is interesting because it interfaces its AI techniques in knowledge representation to more standard computer science techniques in scheduling. As its name implies (Frame Representation Language) FRL is an implementation of Marvin Minsky's frame theory of knowledge structuring [10]. Minsky's ideas are not quite built up into a full-blown theory of knowledge (it is certainly not epistemology), yet when viewed as a competence model the ideas really come together as an important step forward. Minsky has basically presented a theory of how to represent knowledge in a machine. He is saying that knowledge representation is difficult and that to do it well, we had better organize knowledge into frames. His examples of the use of frames in vision work are his way of demonstrating how competent the machine can become if the theory is followed.

3.4.2 Turning to FRL, a simple implementation of the frame vision, we can see how accurate this is. The "language" (more like a data description formalism) is shown by Goldstein and Roberts to support the sorts of knowledge needed to schedule meetings between people. More importantly, it interfaces in a natural way with the user of the system, because its structures are transparent. The user is thus invited to acknowledge its competence in keeping track of times, dates, people and places. Here there is no suggestion that FRL simulates human knowledge representation and the structures and inferences that FRL supports are clearly inadequate to be called epistemology. FRL, however, d= tell us how to organize knowledge in order to do its job, and to do it well.

3.5 Learning - Pat Winston

3.5.1 Another classic of AI is Winston's program [11] which learns how to properly describe (i.e. choose appropriate predicates for) simple blocks world assemblies by being presented with examples and counter-examples of the structure. A typical assembly is an arch which consists of two uprights with an appropriately placed cross piece on top of them. By "knowing" simple relationships such as on-top-of, to-the-left-of, etc., the program manages to generate a semantic network which represents the essential features of an arch.

3.5.2 Here there is a strong feeling that Winston believed that his program could lend support to a larger, more general theory of human learning. However, I feel that apart from the bald statement of the theory itself - that we can learn proper descriptions of complex objects by inducing them from a large number of examples and counter-examples the whole project, including the program, says very little. The simple semantic structures used and the reliance again on theoretical ideas from predicate calculus tends to deny the possibility of the program presenting a genuine theory.

3.5.3 Of course I shall now say that Winston's work is an excellent example of competence modeling. His program really does tell us how we could best attack the problem of learning structures from examples. It tells us, for instance, that we need to know basic spatial relationships already. It also tells us that the example set should be well chosen -nothing too extreme, with a good balance between good structures and "near misses". Above all, it presents a good way (the semantic network) to record what we have learned and how to refer to it in the future.

3.6 Current work in AI

As stated before, I believe that much current work in AI is also best looked at as competence modeling. Good examples are in planning techniques, story understanding and automatic programming. No references are presented here, but the latest International Joint Conference proceedings are a good place to start [12]. Most of these programs do not really add to our understanding of human cognition, but they tell us an awful lot about how good our theoretical ideas are in coping with real and difficult problems. The vast majority of AI programs are designed to dg something - they are not passive vehicles. This is not an accident since the best way to demonstrate competence is to demonstrate it in a practical way!

3.7 Is Artificial Intelligence Therefore Impossible?

For me the only good theory in science is one which describes the phenomena under question. Thus the main traffic is from phenomena to formalism and back for checking - the classic scientific method. AI has seemingly redefined what a theory is by starting with formalism, or at least with essentially theoretical ideas and "discovered" phenomena which the formalism adequately covers. It is inevitable if this line is consistently taken (I am not here accusing the whole of the AI community) that some important phenomena will be overlooked, with disastrous results. It is a different matter if we choose to ignore phenomena - but then we do not end up with a theory, we end up with what I have here called a competence model. AI is thus not ruled out, it is just that most work done under that name has not met the strict, even unfriendly rules of scientific theorizing.

4. Towards the Competent Computer

4.1 The basic thrust behind the competence model is essentially the same as that behind much of decision analysis, operations research, mathematical modeling, and even logic. All of these fields present formalisms which can aid understanding and simplify practical problem solving. However, none of these formalisms is transparent, unless it is to an expert in the field. One reason for this is perhaps that none of the techniques utilized, with the possible exception of logic, find any correlation with experimental evidence of human cognition. If this point is accepted, it becomes obvious why AI techniques provide such good competence models. The theoretical ideas are chosen to mirror facts and intuitions about human thought processes, and thus the end product is transparent to even the naive user. Whereas an O.R. model of an airport, say, might be entirely accurate and produce good predictions we would, I think, be loathe to say that the model or program representing the model was competent at running an airport. If however, an AI approach were taken, the program could potentially be shown to "understand" how an airport should run and operate efficiently in so doing.

4.2 1 am here trying to make a subtle switch from a computer supporting a competence model to a competent computer. The first is interesting but the second is even more so. This is not the fabled robot of Asimov's and others' imagination, but simply a machine which, because it supports a good competence model, can be trusted to get on with the job - it can be given responsibility in that area, because it is an acknowledged expert in the field. It is not intelligent in any sense - just as a mathematical equation is not intelligent - it simply represents our theoretical understanding of the particular domain, And can act out this understanding through the dynamic nature of the machine which operates the model.

4.3 The word "expert" has reared its head again and it is appropriate to return to the abstract and explain its first sentence. The expert system is the competence model par excellence. Since its methodology aims at representing the knowledge of an expert and since an expert is acknowledged to be competent in his field (the two words are very close in meaning) the inference is an obvious one. Notice also that the main representation technique:. production rules, semantic networks, inexact reasoning formalisms are definitely not experimental in nature - they are heavily theoretical. This means that expert systems, above all other AI programs, are telling us how we ought to behave in these areas needing expertise. Interestingly enough, if the competence model is good enough, as it is in MYCIN 1131, DENDRAL (14] and to a lesser extent CRIB [15,16] then the experts will acknowledge the program's competence above their own, and adjust their usual methods of working. To me, this is complete vindication of AI and its techniques and argues well for the future of the subject.

4.4 The future

If we can generate competence models for such diverse fields as natural language, diagnosis, problem solving, speech recogniton (amongst many others) then it may not be beyond us to generate a competence model for acquiring competence. This is usually referred to in AI as the ultimate problem - how to make a computer learn. The lesson of this paper is, I hope, that this is the wrong way to pose the problem. A better way, and one I firmly believe will yield quicker results, is how to acquire sufficient theoretical understanding of the sort of processes involved in learning and acquiring skills such that a model of this understanding can be turned into a program. The understanding will tell us the criteria by which we can judge LE the learning is good or not. The program will therefore tell us how to learn but will not tell us how we do learn.

5. Conclusion

That AI is changing is undeniable. That AI is changing into hard -nosed competence modeling rather than staying with the romantic goal of building an intelligent machine is debatable. However, it is, for me, hard to avoid the feeling that we are emerging from the age of alchemy in AI into the age of enlightened chemistry. The "base metal" that is the everyday working computer may yet turn out to be "gold" in another form. We may riot yet b<e able to call our computers intelligent, but they can certainly do intelligent tasks for us - and do them competently.

REFERENCES

[1] Winograd, T. Frame Representations and the Declarative/Procedural Controversy in Representation and Understanding. Bobrow, D. G. and Collins, A. (Eds.). New York: Academic Press 1975.

[2] Feigenbaum, E. A. Themes and Case Studies in Knowledge Engineering In Michie, D. (Ed.) Expert Systems in the Microelectronic Age. Edinburgh, Edinburgh University Press, 1979.

[3] Johnson, L. Practical Reasoning. Man-Computer Studies Group, Brunel University, U.K. MCSG/TR1. 1979.

[4] Johnson, L. Human Reason. Unpublished PhD Thesis, Cybernetics Department, Brunel University, 1979.

[5] Fikes, R. E. and Nilsson, N. J. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. Artificial Intelligence 2(3/4), 189-208. 1971.

[6] Fikes, R. E., Hart, P. E., and Nilsson, N. J. Learning and Executing Generalized Robot Plans. Artificial Intelligence 3(4), 251-288. 1972.

[7] Sacerdoti, E. D., Planning in a Hierarchy of Abstraction Spaces. Artificial Intelligence, 5(2), 115-135. 1974.

[8] Winograd, T. Understanding Natural Language. New York: Academic Press. 1972.

[9] Goldstein, I. P., and Roberts, R. B. Using Frames in Scheduling. In Winston, P. H.P and Brown, R. H., (Eds.) Artificial Intelligence: An MIT Perspective - Vol. 1. Cambridge, Ma.: MIT Press. 1979.

[101 Minsky, M. A Framework for Representing Knowledge. in Winston, P. H. (Ed.) The Psychology of Computer Vision. New York: McGraw Hill. 1975.

[11] Winston, P. H. Learning Structural Descriptions from Examples. in Winston, P. H. (Ed.). The Psychology of Computer Vision. New York: McGraw Hill. 1975.

[12] Proceedings of the 7th International Joint Conference on Artificial Intelligence. University of British Columbia, Vancouver, Canada. 1981.

[13] Shortliffe, E. N. Computer-Based Medical Consultations: MYCIN. New York: American Elsevier. 1976.

[14] Buchanan, B. G., and Feigenbaum, E. A. Dendral and Meta-Dendral: Their Applications Dimension. Artificial Intelligence, 11(1,2), 5-24. 1978.

[15] Addis, T. R., Towards an Expert Diagnostic System. ICL Journal of Research. London: International Computers Limited. 1980.

[16] Hartley, R. T. How Expert Should an Expert System Be? Proceedings 7th IJCAI. University of British Columbia, Vancouver, Canada. 1981.