4
Human experts and expert systems

P. N. Johnson-Laird

What are expert systems? A brief introduction

Expert systems are computer programs that embody human expertise and that are used as consultants in making decisions about problematical cases. Existing systems identify organic compounds on the basis of spectrographic data (DENDRAL developed by Feigenbaum *et al.* 1971), diagnose infectious diseases (MYCIN developed by Shortliffe 1976), advise geologists where to prospect for minerals (PROSPECTOR developed by Duda *et al.* 1979), and carry out many other such tasks that call for expert knowledge (see Michie 1979). There are even expert systems for setting up expert systems.

Most current systems consist of a large body of conditional rules of heuristic value (extracted by reaping a human expert's knowledge), and an inferential system which, by interrogating a user about a particular problem case, navigates its way through the rules to yield a conclusion. The conditional rules may be definitive or have probabilities attached to them. The inferential system may depend on working backwards from the consequents of the rules, for example given a rule of the form:

if A & B then C

the system in trying to establish C attempts to satisfy A and B (for example MYCIN.) It may work forwards from the antecedents of rules, for example given the data A and B, it deduces C (for example METADENDRAL which discovers rules to be used by DENDRAL). Or it may use both methods, which are indeed familiar from the development of the PLANNER language and its descendants (see, for example, Winograd 1972). The choice of strategy often depends on whether the system is to work 'top down' from hypotheses or 'bottom up' from data. The inferential system may have an estimate of the *a priori* probability of a hypothesis, and estimates of the conditional probabilities of the evidence, both given the truth of the hypothesis and given its falsity, in order to use Bayes's law to compute the probability of the hypothesis in relation to the given evidence (for example PROSPECTOR). It may involve in effect a many-valued or 'fuzzy' logic, for example the assignment of probabilities to each rule and the use of some system of allowing these probabilities and the probability values of data to be jointly computed and propagated along the chain of inference (for example PROSPECTOR, MYCIN). Some expert systems allow the user to ask why a particular datum is required, or how a particular diagnosis was arrived at, and allow an expert user to change the rules in the knowledge base (for example DENDRAL). Finally, expert systems are often programmed using 'production systems' (see Young 1979) or related implementations. They have also been developed in PROLOG; this language is closely related to the predicate calculus (see Warren 1979) and allows programs to be written that simulate non-deterministic search by backtracking (for the computational notion of non-determinism, see Hopcroft and Ullman 1979).

What are human experts?

There are gross differences between human experts and expert systems that are obvious to the cognitive scientist if not to the 'knowledge engineer'. It is easy to draw these contrasts, but rather harder to formulate a working definition of what constitutes human expertise. I am happy with the following tendentious definition: *an expert is someone who can make good excuses for being wrong*. This definition is at least extensionally correct: it applies to human experts but to no current expert system. In fact, however, there are five major distinctions between human and computer expertise.

First, and most obvious, humans can treat errors not merely as cause for excuse but also as the occasion for a revision in their knowledge or theories. Expert systems are not normally confronted with their errors.

Second, humans can do much more with their knowledge than can systems. They can use their knowledge in a flexible way to answer many different sorts of problem.

Third, human experts have a large amount of tacit knowledge that they cannot readily articulate in words. This knowledge is important in science (Polyani 1958), and it is crucial in those areas where science has yet to tread. Try asking a poet how to write a poem, and you will discover that poets have different working methods—though most of them still rely on the 'muse'. Try asking a wine taster how to distinguish one claret from another, and you are likely to be given a rule that is impossible to objectify, for example one tastes like pencil shavings, the other like wet slate. Try asking anyone for the grammatical rules of natural language, and you will learn hardly anything of value. Linguists have been working on this problem for two thousand years or more, and they have yet to formulate a complete grammar for any natural language or even to decide how powerful, computationally speaking, such grammars should be.

Fourth, there is an important but subtle difference in semantics. Humans have knowledge *about* things, whereas expert systems merely have sets of conditional rules which are denotationally interpretable by the human user, but not by the program. Indeed, the program has generally no way of relating its representations to the world; it has no way of determining the truth or falsity of the contingent assertions with which it deals. Since nearly all computer programs are in this predicament—as to reality, one might say, they leave that to the programmer—it is seldom commented upon. Certain 'knowledge enginers' may be inclined to regard this point as a philosophical scruple comparable to the objection that thermostats do not have intentions, but the lack of a semantics relating expressions to representations of the world is, as we shall see, an important obstacle to the solution of certain problems.

Fifth, humans and systems make inferences using rather different mechanisms. This point will emerge more clearly later in the chapter.

If you grasp these five points, you may well wonder how it is that some expert systems function so well. The answer is evident if you consider how you would develop a new expert system that, say, helped taxonomists in the identification of different species of coelenterates. Human experts have a coherent body of knowledge about the members of this phyllum: they have mental models that integrate information about their anatomy, physiology, ecology, and behaviour (for the notion of mental models, see Gentner and Stevens 1982; Johnson-Laird 1980, 1983a). With the assistance of a knowledge engineer, the expert may be able to translate certain features of these models into a set of recognition procedures couched in the form of conditional rules. These rules may suffice for recognition, but they will be useless for other tasks calling for expertise, for example how to cope with a plague of jellyfish. In short, the knowledge engineer reaps the fruit of the expert's knowledge, not the knowledge itself. Fruit make an excellent dessert, but experts do not live by fruit alone.

How are we to improve existing systems and to make them more expert? There are two obvious methods. The first is to devise some new and cunning techniques in artificial intelligence. The second is to try to close the gap between human and programmed expertise. Both methods have a contribution to make. Yet, a truly expert 'expert system' will surely have the following capabilities: it will be able to evaluate evidence without the mediation of a human user since it will contain its own semantics; it will be able to direct its expertise to a variety of sorts of problem in a flexible way; and, of course, it will make good excuses for its mistakes and learn from them. My aim in what follows is to offer some suggestions about how we might advance towards this goal. These suggestions will be motivated by considering one crucial problem: the maintenance of consistency.

How to maintain the consistency of a knowledge base

An expert system has a knowledge base of rules, and obviously its improvement may depend on modifying or replacing some of them. There are several possible approaches to such changes. The standard approach *finesses* the real problem—the maintenance of consistency—since the knowledge engineers themselves are responsible for any modifications to the system. Users can interrogate the knowledge base but they cannot alter what is there. This procedure may work well, but obviously it is more efficient if the maintenance of the knowledge base can be mechanized (cf. METADENDRAL). Of course, one can make an heroic denial of the problem of consistency. To paraphrase Walt Whitman, one can say: 'Does my knowledge base contain contradictions? Very well, it is big, it contains multitudes.' No one, perhaps, deliberately adopts this attitude, but if rules are to be routinely added to the knowledge base, then the possibility of inconsistencies is an ever-present danger. A much more sensible approach is to rely on logic. The knowledge base is, in effect, treated as a set of expressions in the first-order predicate calculus, and whenever a new rule is to be added, a theorem-prover is used to check the consistency of the modified system. The promise of PROLOG rests in part on the attraction of this approach. There are, however, at least three problems with it.

The first difficulty is well known: there are algorithms, such as resolution theorem-provers, that will determine in a finite number of steps that an assertion p is inconsistent with a set of assertions D (equivalently, the negation of p follows logically from D), but there can be no such general algorithms that determine that p is consistent with D. For any system with the power of the first-order predicate calculus (i.e. with quantification ranging over individuals), we can therefore have only half of a decision procedure, and so there will be no way of telling whether a program for testing the proposition is going to yield an inconsistency or continue to run *ad infinitum*. Although this problem is unsolvable, there are occasions, as we shall see, where it can be avoided in the maintenance of data bases.

The second difficulty arises from the meaning of conditionals, or rather from the lack of a semantic theory for them. Conditionals in computer programs have a simple semantics, but conditionals in daily life have a meaning that is so complicated that no one has yet succeeded in formulating a satisfactory theory of their semantics (see Johnson-Laird 1983b). Suppose, for example, an expert formulates a rule of the form:

If p, then r

then in all orthodox systems of logic no matter what is conjoined to the antecedent the resulting rule will still be true:

> If p & q, then r

In reality, however, this 'strengthening' of the antecedent may be unacceptable. For instance, one might accept the rule:

> If infection 1 is cystitis and infection 2 is bacteremia, then administer penicillin.

but totally reject the rule:

> If infection 1 is cystitis and infection 2 is bacteremia and the patient is allergic to penicillin, then administer penicillin.

The trouble is that in orthodox systems of logic the two rules are not merely consistent with one another, but whenever the first rule is true, the second rule is true, too. The second rule follows logically from the first!

This problem is a special case of a general phenomenon. Orthodox logics are *monotonic*; that is, given a set of premises D that validly imply a proposition p, then the addition of any further premise q to D cannot affect the validity of the inference to p:

> If $D \vdash p$, then $D \;\&\; q \vdash p$.

Various attempts have been made to formalize rules of inference that are non-monotonic (for example McDermott and Doyle 1980), but they have not yet succeeded (Davis 1980). The nature of the difficulty can be illustrated by returning to the example. Suppose that the conditional:

> If infection 1 is cystitis, then administer penicillin

is correct. We may also want the conditional:

> If infection 1 is cystitis and infection 2 is bacteremia, then administer penicillin

to be correct, but the conditional:

> If infection 1 is cystitis and the patient is allergic to penicillin, then administer penicillin

to be incorrect. There is, however, no difference in the logical form of these two conditionals. Their inconsistency arises at a deeper level, since it depends on our knowledge of the effects of penicillin on those who are allergic to it. The moral is plain: current systems of formal logic and their implementations in theorem-provers cannot cope with inconsistencies between ordinary conditional rules.

The third difficulty in detecting inconsistencies also concerns the use of formal techniques, and again I shall illustrate it by way of an example. Suppose there are two rules of the form:

If more than half the *A*s are *B*s then *X* is the case
If more than half the *A*s are *C*s then *X* is the case

and you wish to add the following rule to the knowledge base:

If at least one *B* is a *C*, then *X* is *not* the case.

Plainly, your new rule is inconsistent with your old rules, since the following inference is valid:

More than half the *A*s are *B*s
More than half the *A*s are *C*s
Therefore, at least one *B* is a *C*.

There is no way, however, in which this inconsistency can be detected in the first-order predicate calculus or a computer implementation of it. Why? Because the quantifier 'more than half', like many others, cannot be captured within any system in which existential and universal quantification occurs over individuals (Barwise and Cooper 1981). One needs quantification over sets, i.e. the second-order predicate calculus. But this calculus is incomplete in that there can be no set of formal rules of inference that capture all valid second-order deductions.

Doubtless, there are *ad hoc* solutions to the last two of these problems, but their existence should at least make us pause to think before we adopt systems based on formal rules of inference to check the consistency of a knowledge base. There is indeed an alternative approach that derives from the study of human cognition.

Mental models and the knowledge base

The problem of maintaining the consistency of a knowledge base (or indeed of any sort of data base) can be approached obliquely by considering first an apparently rather different problem: how people understand discourse and build up a mental representation of it. There are grounds for supposing that human beings have the ability to construct a mental model of the state of affairs described by the discourse (Johnson-Laird 1983a). This theory of mental models assumes that comprehension is a two-stage process: first, a superficial linguistic representation of an utterance is set up; second, this representation is used in the construction of a mental model of the state of affairs that the discourse describes. In general, a discourse is represented by just a single mental model, which is based on the meaning of the utterances in the discourse, relevant information about the context, and inferences from general knowledge. The content of a particular utterance is merely a clue to what has to be constructed: inferences 'flesh out' the bones of the discourse.

If one builds only a single model, then since nearly all discourse is radically indeterminate, the problem arises as to how to construct a model in the absence of a complete description. One method is to make an arbitrary choice, and, if the choice turns out to be wrong in the light of subsequent information, then to revise the model. The significance of the model therefore depends on the existence of procedures that can revise it so as to undo mistaken assumptions or inappropriate interpretations of indeterminate descriptions. The system therefore simulates a non-deterministic automaton that always constructs the correct model of the discourse (given that the discourse describes just one state of affairs): it converges on the correct model by modifying the existing model so as to render it compatible with the discourse as a whole (provided that the discourse is consistent).

At the heart of the theory lies the following idea: mental models represent the extensions of assertions, i.e. the situations they describe, whereas the superficial linguistic representations, together with the machinery for constructing and revising models, represent the intensions of assertions, i.e. the sets of all possible situations that the assertions could describe. In effect, a mental model is a fragment of many possible worlds: all those possible worlds within which the model can be embedded because the discourse it represents is true in them. Thus, the structure of mental models corresponds to the structure of a state of affairs. A mental model is therefore different in structure from a semantic network, or a representation in the form of a syntactically structured string of symbols. Both these types of representation have structures quite remote from the states of affairs that they designate. For example, a semantic network representing the assertion:

Every man owns one car

contains a node representing men, a node representing ownings, a node representing cars, a node representing implications, and a number of other nodes to capture the appropriate logical structure of the assertion (see Hendrix 1979). A mental model merely contains a set of mental tokens corresponding to the set of men, a set of mental tokens corresponding to the set of cars, and a set of mental relations between these tokens representing the relation of ownership:

man → car
man → car
      (car).

The token in parentheses represents the possibility of a car that is not owned by a man. This model is a fragment of all those possible worlds in which there are two men and three cars, and every man owns one car. The significance of the model is not limited to sets with these cardinalities. The numbers of tokens can be chosen arbitrarily if there is no information to the contrary, and they can be revised in the light of subsequent information. Obviously, a plausible psychological theory must make further assumptions about how to deal with large numbers: the model is not constructed in complete detail.

The way in which to maintain consistency can be illustrated by considering in more detail the process of representing discourse in a mental model—a process that has been modelled in a program that makes spatial inferences on the basis of verbal descriptions (see Johnson-Laird 1983a). The program requires seven general procedures:

1. A procedure that begins the construction of a new mental model whenever an assertion makes no reference, either explicitly or implicitly, to any entity in the current model of the discourse.
2. A procedure which, if at least one entity referred to in the assertion is represented in a current model, adds the other entities, properties, or relations, to the model in an appropriate way.
3. A procedure that integrates two or more hitherto separate models if an assertion interrelates entities in them.
4. A procedure which, if all the entities referred to in an assertion are represented in a current model, verifies whether the asserted properties or relations hold in the model.

The verification procedure may be unable to establish the truth value of an assertion for a current model: there may be no information in the model about the relevant property or relation. A model of John standing next to June, for instance, does not yield a truth value for 'John is taller than June'. In such cases, there is:

5. A procedure that adds the property or relation (ascribed in the relation) to the model in the appropriate way. (This procedure is not implemented in the program.)

Since only a single model is constructed for a discourse with referents in common, the procedures that construct the model will inevitably be forced to make arbitrary assumptions since ordinary assertions are almost always consistent with more than one state of affairs. Such a decision may be wrong in that it leads to a conflict with subsequent information in the discourse. Superficial conflicts will be revealed by the verification procedure. Two recursive procedures are needed, however, in order to cope with the simulation of the non-deterministic device that always constructs the right model. These procedures are able to detect genuine inconsistencies between the model and a new assertion:

6. If an assertion is found to be true of the current model (by the verification procedure), then there is a procedure that checks whether the model can be modified in a way that is consistent with the previous discourse but

that renders the current assertion *false*. Where no such modification is possible without doing violence to the previous discourse, the current assertion adds no new semantic content: it is a valid deduction from the previous assertions.

7. If an assertion is found to be false of the current model (by the verification procedure), then there is a procedure that checks whether the model can be modified in a way that is consistent with the previous discourse but that renders the current assertion *true*. When no such modification is possible without doing violence to the previous discourse, the current assertion is inconsistent with the previous discourse. If such a modification is possible, however, then the inconsistency is thereby resolved.

The first of these procedures embodies the fundamental semantic principle of validity: an inference is valid if and only if there is no model of the premises in which the conclusion is false. The second of these procedures embodies what might be termed the fundamental semantic principle of non-monotonic inference: any arbitrary assumption consistent with the premises can always be made in building a model, since it can always be revised if it is inconsistent with subsequent discourse.

The reason that these two procedures must be recursive can be illustrated by considering the operation of the program for spatial inferences. If the program is given the radically indeterminate description:

Z is on the right of *Y*
X is on the left of *Z*

it constructs the following spatial model:

X Y Z

If it is subsequently told:

X is on the right of *Y*

then the verification procedure (4) initially returns the value false. This value elicts the procedure (7) for determining whether there is any way in which the model could be rearranged so as to render the assertion true. Obviously, what the procedure must do is to check whether, if it switches round the positions of *X* and *Y*, the result is still compatible with the earlier discourse. In fact, there is no hindrance to the switch's taking place. But, suppose instead that the previous discourse had also contained an assertion to the effect that *W* is in front of *X*, and that the model therefore was of the form:

X Y Z
W

In this case, before X can be switched with Y, it is necessary to check whether the position of W can be similarly shifted. There is nothing in the discourse to prevent such a switch and the program will construct the revised model:

$$\begin{array}{ccc} Y & X & Z \\ & W & \end{array}$$

Let us consider one further example, however, in which other, earlier, assertions have established that V is in front of Y, and that V is on the right of W, yielding the model:

$$\begin{array}{ccc} X & Y & Z \\ W & V & \end{array}$$

At this point, the program is told as before that X is on the right of Y. It must now first attempt to switch X and Y, but it discovers that W is in front of X and that V is in front of Y. Hence, it must check the earlier discourse to determine whether W and V can be switched round; in fact, the switch is impossible because of the assertion that V is on the right of W. Only at this point is it plain that X and Y cannot be switched, and thus the program announces that the new assertion is inconsistent with the previous description.

There is no limit to the number of dependents of items in a model that might have to be followed up in order to determine whether the model can be reorganized. That is why the process has to be handled by a procedure that can call itself recursively.

The reader may by now be beginning to wonder what all this discussion of mental models and discourse has to do with the problem in hand: the maintenance of consistency in a knowledge base. The answer in a nutshell is that a knowledge base can be treated as a set of models of discourse. The machinery that has been described can be used *mutatis mutandis* to maintain consistency.

Conclusions

The use of models in a knowledge base depends, of course, on formulating truth-conditions for the relevant linguistic descriptions that will at the very least relate them appropriately to the models. The technique of searching for inconsistencies by relating a new rule to what is in the knowledge base stands or falls by the existence of such a semantics. It is a simple matter to formulate the appropriate truth-conditions for spatial expressions; it is much harder to formulate them for such terms as ‘allergic’, ‘cystitis’, etc. This problem certainly calls for expert knowledge, but it is an epistemological problem that confronts any theorist wishing to develop an adequate semantic theory and is in no way peculiar to the present approach. However, it is important to distinguish two levels of a model-based semantic system: the first level relates expressions to internal models, and the second level relates internal models to the world. The use of models as a means to consistency depends only on formulating a semantics at the first level. The general strategy that is employed nevertheless has the advantage of solving the problems that confront systems based on formal rules of inference.

First, if the models in a data base are finite and the recursive processes for revising them are restricted to finite possibilities, then the system will constitute an effective decision procedure for the consistency of any assertion with the set of assertions embodied in the models. The domains of most existing expert systems appear to be appropriately modelled in such a finite way; that is, unlike a mathematical calculus, one can always posit a maximum number of individuals of any sort without doing violence to the rules. Hence, for such domains, it is possible to obviate the lack of a general decision procedure for the first-order predicate calculus.

Second, the tactic of making an assumption that can then be revised in the light of subsequent information is indeed what is needed in order to accommodate the non-monotonic aspects of ordinary reasoning. Conditional rules can thus be treated as having the following force:

> If *p* is the case and there is nothing to the contrary, then *q* is the case

Third, once the truth-conditions for such quantifiers as 'more than half' have been formulated (in relation to models), there is no need to specify formal rules of inference since the recursive procedures work without them.

The use of models in a knowledge base is not a new idea (cf. DENDRAL), but what perhaps has not been appreciated is how they open up the way to more effective procedures for maintaining consistency. They also offer a potential means to improve other aspects of expert systems. A model can form the basis for different uses of expertise. Just as a human expert may depend on models for carrying out a variety of tasks, so too models in a computer system are likely to provide a more flexible representation of knowledge than explicit conditional rules. Moreover, if the models are isomorphic to the state of affairs they represent, then a major step has been taken towards the development of a semantics that relates a program's representations to the world.

References

Barwise, J. and Cooper, R. (1980). Generalized quantifiers and natural languages. *Linguist. Phil.* **4,** 159–219.

Davis, D. (1981). The mathematics of non-monotonic reasoning. *Artif. Intell.* **13,** 73–80.

Duda, R., Gaschnig, J., and Hart, P. (1979). Model design in the Prospector consultant system for mineral exploration. In Michie, D. (ed.) *Expert systems in the micro-electronic age*. Edinburgh University Press, Edinburgh, 153–67.

Feigenbaum, E. A., Buchanan, B. G., and Lederberg, J. (1971). On generality and problem solving: a case study using the DENDRAL program. In Meltzer, B. and Michie, D. (eds), *Machine Intelligence 6*. Edinburgh University Press, Edinburgh, 165–90.

Gentner, D. and Stevens, A. L. (eds) (1982). *Mental models*. Erlbaum, Hillsdale, NJ.

Hendrix, G. G. (1979). Encoding knowledge in partitioned networks. In Findler, N. V. (ed.), *Associative networks: representation and use of knowledge by computers*. Academic Press, New York. 51–92.

Hopcroft, J. E. and Ullman, J. D. (1979). *Formal languages and their relation to automata*. Addison-Wesley, Reading, Mass.

Johnson-Laird, P. N. (1980). Mental models in cognitive science. *Cognit. Sci.* **4,** 71–115.

Johnson-Laird, P. N. (1983a). *Mental models: towards a cognitive science of language, inference, and consciousness*. Cambridge University Press, Cambridge; Harvard University Press, Cambridge, Mass.

Johnson-Laird, P. N. (1983b). Models of conditionals. Mimeo, MRC Applied Psychology Unit, Cambridge.

McDermott, D. and Doyle, J. (1980). Non-monotonic logic I. *Artif. Intell.* **13,** 41–72.

Michie, D. (ed.) (1979). *Expert systems in the micro-electronic age*. Edinburgh University Press, Edinburgh.

Polyani, M. (1958). *Personal Knowledge*. Routledge and Kegan Paul, London.

Shortliffe, E. (1976). *Computer-based medical consultations: MYCIN*. Elsevier, New York.

Warren, D. (1979). PROLOG on the DEC system-10. In Michie, D. (ed.), *Expert systems in the micro-electronic age*. Edinburgh University Press, Edinburgh, 112–21.

Winograd, T. (1972). *Understanding natural language*. Academic Press, New York.

Young, R. M. (1979). Production systems for modelling human cognition. In Michie, D. (ed.), *Expert systems in the micro-electronic age*. Edinburgh University Press, Edinburgh, 35–45.

Intelligent Systems in a Human Context

Development, Implications, and Applications

Edited by

LINDA A. MURRAY and JOHN T. E. RICHARDSON

Lecturer in Psychology
Brunel University

Reader in Psychology
Brunel University

Oxford New York Tokyo
OXFORD UNIVERSITY PRESS
1989

Oxford University Press, Walton Street, Oxford OX2 6DP
Oxford New York Toronto
Delhi Bombay Calcutta Madras Karachi
Petaling Jaya Singapore Hong Kong Tokyo
Nairobi Dar es Salaam Cape Town
Melbourne Auckland
and associated companies in
Berlin Ibadan

Oxford is a trade mark of Oxford University Press

Published in the United States
by Oxford University Press, New York

British Library Cataloguing in Publication Data
Intelligent systems in a human context:
development, implications, and applications.
1. Artificial intelligence. Social aspects
I. Murray, Linda A. II. Richardson, John
T.E. (John Thomas Edwin), 1948–
306'.46
ISBN 0–19–853736–0

Library of Congress Cataloging in Publication Data
Intelligent systems in a human context, development, implications, and applications/[edited by Linda A. Murray and John T.E. Richardson].
p. cm. Bibliography: p. Includes index.
1. Artificial intelligence. 2. Expert systems (Computer science)
3. Artificial intelligence—Social aspects. I. Murray, Linda A.
II. Richardson, John T. E. III. Oxford University Press.
Q335.155 1988 006.3—dc 19 88–19546

ISBN 0–19–853736–0

Set and printed by Butler & Tanner Ltd, Frome and London