In: M. Bramer, editor, *Research and Development in Expert Systems*
Proceedings BCS Expert Systems Conf. 1984
Cambridge University Press 1985, pp 163-183

Why We Need Many Knowledge Representation Formalisms

Aaron Sloman

Cognitive Studies Programme
University of Sussex, Brighton, UK
Now at School of Computer Science, The University of Birmingham, UK
http://www.cs.bham.ac.uk/~axs/

Abstract

Against advocates of particular formalisms for representing *all* kinds of knowledge, this paper argues that different formalisms are useful for different purposes. Different formalisms imply different inference methods. The history of human science and culture illustrates the point that very often progress in some field depends on the creation of a specific new formalism, with the right epistemological and heuristic power. The same has to be said about formalisms for use in artificial intelligent systems. We need criteria for evaluating formalisms in the light of the uses to which they are to be put. The same subject matter may be best represented using different formalisms for different purposes, e.g. simulation vs explanation. If different notations and inference methods are good for different purposes, this has implications for the design of expert systems.

1. Introduction

It is sometimes a good strategy to adopt an extreme position and explore the ramifications, for instance choosing a particular language, or method, and acting as if it is best for everything. This can have two consequences. First, by striving to use only one approach one is forced to investigate ways in which that approach can be extended and applied to new problems. Secondly if the approach does have limitations we will be in a better position to know exactly what those limitations are and why they exist.

For example, it is a good thing that some people should think, rightly or wrongly, that the methods of AI can be used to simulate and explain every aspect of human mentality, and try to establish this by doing it. If they are wrong, their efforts will give us new insights into both why they are wrong, and what the AI methods can achieve.

Similarly, it is good that some people pin all their hopes on first order predicate logic (FOPL) as the language to be used for all purposes, as recommended by Bob Kowalski, in characteristically forceful style:

> ‘‘There is only one language for representing information -- whether declarative or procedural -- and that is first-order predicate logic. There is only one intelligent way to process information -- and that is by applying deductive inference methods.’’
>
> (Kowalski 1980)

Many people have been inspired by this idea, and as a result excellent work has been done, and will be done, exploring the power of logic-based programming languages. It is, perhaps, sad that

without the motivation provided by a mistaken ideal, such work might not be done.

Anyway, my present purpose is neither to praise nor to bury logic but to understand its limitations and assess some alternatives. Predicate logic needs no praise from me, for it is clear that there is no other formalism which is simultaneously so well understood, so widely applicable, and so clear in its semantics. It must therefore play an important role in theorising about intelligent systems, and perhaps also in modelling them.

Nevertheless, logic is not all-embracing. It may be best for the largest range of uses, without being best for everything. FOPL is an example of an 'applicative' formalism (defined below). I have previously argued (Sloman 1971, 1978) that for some purposes analogical representations may be better than applicative formalisms, including logic. Bobrow (1975) suggests, more generally, that there are several dimensions along which the utility of representations may be compared. Logic does not always win. Goodman (1969) has also compared different schemes of representation. (He makes more distinctions than I shall discuss here.)

It is evident that many different representational systems are used by ordinary people, by scientists and by engineers, including maps, models, diagrams, flow-charts, etc. And there is plenty of evidence that logical thinking is not always what people use most naturally or effectively (Johnson-Laird, 1983), though that in itself does not prove that logical thinking should not be used for all tasks.

2. Terminology

In this general survey of issues, I shall not be very precise in my terminology, switching between expressions like 'symbolism', 'language', 'notation', 'representational system' and the like. In all cases I am talking about a set of possible structures which can be used in a systematic way for one or more of the following: storing information, communicating information, comparing information, formulating questions or problems, making inferences, formulating plans, controlling actions, etc. It does not matter whether the structures are within the mind, brain, or computer, or external structures, e.g. marks on paper. Neither does it matter whether they are concrete physical structures or abstract ''virtual'' structures (explained below).

I shall say very little about notations used for communication between intelligent systems as I regard the ''internal'' functions as more basic (for reasons given in (Sloman 1979)). A more complete discussion would need to analyse the problems and constraints involved in various forms of communication, showing which features of a notation make it more or less useful. For instance, redundancy will assist the cognitive processes in a receiver. Even for internal reasoning, within a single intelligent system, we shall see that different notations will be best for different purposes. Compare Brown and Burton (1975).

3. Why notations are important

In designing or describing an intelligent system, we can consider its knowledge at varying depths.

(1) The actual content of the knowledge base will be unique to that individual and be able to explain only its actual behaviour.

(2) The system of concepts used explains how that knowledge can be grasped, and would account equally well for many alternative contents. It also explains the range of questions which can be asked, problems formulated, instruction understood, etc.

(3) The formalism or notation used, together with relevant procedures, explains how that set of concepts, and the knowledge or questions expressed with their aid, may be stored, communicated, manipulated in reasoning, etc. But the same notation might be used for a quite different range of concepts, and thus the notation explains a wider range of possible belief states.

So the notation used has a deep explanatory role. What gives a notation or formalism its power is not just the static structure of the various symbols or representations, but the *procedures* available to operate on it, e.g. matching, parsing, substituting, etc. Logical inference procedures are a special case.

4. Some alternative notations

A survey of notations, formalisms, symbolisms, representational systems, used by mathematicians, scientists, engineers, musicians, programmers, choreographers, cartographers, and even some logicians will show that there is a very wide variety of types. It is very unlikely that all of these, or even a majority, have grown out of arbitrary whims. Rather there are cultural and evolutionary pressures provided both by the nature of the domain of application and the purposes for which they are used, which have shaped their development. Some of the pressures include the perceptual and cognitive problems involved in parsing and interpreting structures. Some include the requirements of cognitive processes making use of the structures for such varied purposes as inference (e.g. calculation), planning, searching and problem solving, and detailed control of actions.

For example, it seems that for many purposes of control of actions, the two-dimensional structure of the visual field plays an important role, as it does in discerning what somebody is looking at, by following the line of sight across the scene.

Even within logic, geometrical representations and geometrical procedures play a role. For instance the unification algorithm used in logic programming is a 'geometrical' inference procedure, operating on the syntactic structure of symbols to infer whether one can be transformed into another by a suitable geometric substitution. If instead of representing logical formulas geometrically, and performing geometric operations on them, we tried to use only logical descriptions of their structure, and logical deduction, then we should have an infinite regress.

If it is true that different sorts of representations and inference procedures should be used for different sorts of tasks, then the implications for Expert System design are profound.

5. Applicative and other notations

What is special about logic? We shall answer this in terms of the type of syntax (or geometry) used, the semantic presuppositions and the inference procedures. FOPL uses an 'applicative' notation. This means:

(a) Applicative syntax.
All composite formulas are constructed from simpler formulas by a combination of function symbols and argument symbols, so as to represent the application of a function to its arguments.

(b) Applicative semantics.
The denotation of any complex symbol depends on the individuals denoted by its arguments and the function denoted by the function symbol. (How it depends on them needs to be defined more precisely.)

The second requirement, (b), is sometimes referred to as the principle of compositionality, probably first understood and analysed by Frege (which is why in (Sloman 1971, 1978) I previously

used the label 'Fregean' for applicative representations). Using an applicative representation presupposes that there are individuals and functions (mappings from individuals to individuals). We later show this to be a restrictive presupposition.

Together, (a) and (b) imply the *principle of substitutivity:* in an applicative formalism, symbols and well-formed combinations of symbols will be divided into syntactic types, such that in a complex well-formed formula, any well-formed sub-formula can be replaced by another (simple or complex) of the same type and the result will still be well-formed. Moreover this will not affect the significance of any other symbol in the formula. For example, in ''3 + 2'', ''2'' can be replaced by ''(5 x 99)'' and ''+'' can be replaced by ''-'', and in either case the remaining symbols and their denotations will not be altered, although the denotation of the whole expression is changed.

Thus, ''3 + 5'' uses an applicative notation, with ''+'' as function symbol, ''3'' and ''5'' as arguments. Similarly, in its obvious interpretation, ''between(3,2,4)'' is applicative, though here the function is from triples of numbers to truth-values.

Pictures generally violate the principle of substitutivity. E.g. in a picture of animals in a field, there is no way that a picture of an elephant can be replaced by a picture of a giraffe, without it also making a difference to how much of the background is depicted.

The substitution property is one of the features which gives logic its generality. Assertions made about one class of objects, or inference principles discovered in relation to one class of objects, may be sensibly transferred to others by substituting appropriate sub-expressions. This encourages the formulation of new conjectures and various kinds of analogical and metaphorical reasoning. So applicative notations underpin some of the most powerful and creative reasoning processes.

First order logic uses an applicative notation, since, as Frege noticed, predicates are functions from n-tuples of objects of any kind to truth-values. Moreover, quantifiers (e.g. ''for all x'', ''for some x'') can be construed as 'second level' functions from predicates to truth-values. (I.e. 'for all x P(x)' is true if every meaningful substitution of an argument in 'P( )' produces a true result.

(c) The *principle of extensionality,* is also a feature of applicative notations. It states that if F1 is a well-formed formula, and S1 is a subformula of F1, then if S1 is replaced by another formula S2 with the same denotation, transforming F1 into F2, then F2 will have the same denotation as F1. Extensionality is a feature of FOPL. (This definition needs to be relativised to a situation or possible world. See Sloman (1965))

As Frege first pointed out, it seems that natural languages do not satisfy this condition, and in particular that sentences about the mental state of an intelligent system will not always retain their truth-value if a component is replaced by another with the same denotation. E.g. if 'Bill Bloggs' and 'The mayor of Maresville' denote the same individual, then replacing the former with the latter will not alter the truth-value of an extensional assertion like:

'Bill Bloggs hit Harry Holmes'

whereas it may alter truth value in an intensional context, like:

'Fred Fikes believes Bill Bloggs hit Harry Holmes'

Various attempts have been made to show how such assertions can be translated into FOPL, preserving extensionality, e.g. using a metalinguistic extensional language. For instance,

'Fred believes the evening star is the morning star'

might translate into something like:

sentence(s1) & believes(Fred,s1) & identity(s1)
& arg(1, s1, ‘the evening star’)
& arg(2, s1, ‘the morning star’)

I.e. there is an identity statement which Fred believes whose arguments are: ‘the evening star’ and ‘the morning star’. More complex translations would be required for other sorts of beliefs.

There are problems with this sort of suggestion. An acceptable translation must not assume that Fred is an English speaker, for example. Attempting to get round this by avoiding literal English strings, and instead using a representation of the meaning common to them and their translations in other languages, might re-create intensional contexts. For an alternative attempt to rescue FOPL see McCarthy 1979. Frege’s own means of rescuing the principle of extensionality was to allow an embedded expression to denote an abstract entity called the ‘‘sense’’ or ‘‘intension’’ of the expression. The debate will no doubt continue for a long time.

For now we may merely note that it is not obvious that an extensional notation like FOPL is adequate for describing intelligent systems. Expert systems which reason about intelligent agents may therefore need a richer formalism.

6. The importance of truth-values

One of the reasons for the generality of logic is that it postulates in its ontology (i.e. its implicit theory about what exists) a class of entities called booleans, i.e. the truth-values TRUE and FALSE. What these entities actually are is irrelevant, since all that is important is their role in defining logical operations.

The same expression (e.g. ‘P or Q’) may evaluate to TRUE in a variety of ways. So simply indicating that the expression is true is a convenient way of conveying very non-specific information, which is often useful when further details are either irrelevant or unknown. For instance if the predicate ‘is red’ is defined suitably, then ‘the ball is red’ conveys information about the colour of the ball without being at all specific about the precise shade of red, etc. By contrast, a painting of the ball would not normally be able to do this. Similarly ‘There is a table in front of me’ is totally non-committal about which table it is, what sort of table, and how it is spatially related to the speaker. A painting cannot be so non-committal, though some styles attempt to overcome this.

Paradoxically, almost, we can say therefore that part of the power of logic is its ability to express various kinds of imprecise information, including negative, disjunctive and existentially quantified assertions. However, we shall see that it may nevertheless be limited by specific sorts of ontological commitments.

To sum up, FOPL uses applicative notations, with a compositional denotative (i.e. extensional) semantics. It uses the principle of substitutivity for its generality and the principle of extensionality to achieve its semantic simplicity. It presupposes the existence of individuals of some kind, including booleans, and (situation relative) mappings from individuals or sets of individuals to booleans -- i.e. properties and relations.

7. Logic presupposes a meta-ontology

We have seen that logic (or the use of logic) presupposes that the world can be construed as made up of:

(1) objects (including booleans)

(2) functions (which subsume properties and relations.)

It is arguable that this is too restrictive for some purposes. Consider a human body. We do have names for many parts, but there are few natural boundaries: the names refer to portions which are not necessarily precisely demarcated from the rest, causing difficulties in deciding whether a particular object has a property or stands in some relation to another object. Deciding whether a forefinger is or is not longer than a thumb depends on what the boundaries are: compare the views of a hand from the palm side and the knuckle side.

More importantly, a physical object appears to be a continuum, and at least at the resolution at which we can perceive or think about it, seems to be indefinitely divisible in many different ways, rather than made up of a fixed hierarchy of objects. Different kinds of properties and relationships will be relevant to different modes of subdivision. For some actions, such as touching an object, little or no decomposition may be required.

So, to represent the way we perceive a body, or even the way we think about what we perceive, or the action of running a finger smoothly along the surface of a torso, we require a representation which does not presuppose some decomposition into well-defined objects. Sculptures and paintings are examples of such representations. They are examples of what I called 'analogical' representations (in 1971). Instead of explicitly naming individuals, properties and relations, they represent complex wholes by allowing properties and relations to be represented implicitly by properties and relations, including shapes, colours, etc.

8. Analogical representations and continuity

The concept of an 'analogical representation' does not require the representation or what it denotes to be continuous or ontologically uncommitted. For example, a list of names of people might be a discrete analogical representation of the order in which the people were born, or the chain of command in a military unit. Similarly, in a Prolog program, the order of portions of the text analogically represents the order in which subgoals (at a certain level) are attempted. This is Prolog's procedural aspect.

An analogical notation may or may not be capable of representing some continuous reality. A discrete analogical or logical notation may be used to describe a continuous object to any desired degree of resolution if there is some means of decomposing that reality into small enough individuals. Does that presuppose the use of some *other* notation to represent the continuous reality prior to finding a good decomposition?

Computer vision systems use a quantized approximation to continuous representations e.g. 2-D arrays. These may be thought of as providing a *sample* of data in a continuous optic array. The same sample could be represented in a database of logical assertions, though, in the array, unlike a logical database, neighbourhood and other relationships are represented analogically, possibly at a 'virtual' level (explained below).

The array is committed to a particular ontology for the sample, i.e. a finite set of measurements, but not for the domain of structures represented. Procedures which search for evidence of edges may be thought of as helping the search for a good decomposition.

So a finite discrete machine can embody representations of a continuous, unarticulated, reality in an ontologically uncommitted fashion. (This requires further analysis.)

9. "Analogical" does not imply "similar" or "isomorphic".

People often fall into the trap of assuming that an analogical representation must be *isomorphic with* or *similar to* what it represents. But this is not necessary. For instance, in a flat picture of a

three dimensional scene, relations between things in the picture represent relations between things in the scene, yet picture and scene have quite different structures - one is two dimensional and the other three dimensional. Moreover the relation between what is represented and how it is represented may vary according to context. In a picture of a room, the relation 'higher' in the picture may represent any of 'higher', 'further', 'nearer', depending on which portion of the picture is involved (Sloman 1971, 1978). Similarly, a flow chart may represent a linear computer process in which *many* sub-processes are represented by *one* loop. Despite the lack of isomorphism between chart and process, this is an analogical representation, though, like a map, it may also include other notations. Finally, as we have seen, a discrete, finite, structure may be an analogical representation of a continuous structure.

10. Uncommitted ontologies

More importantly, an analogical representation need not be composed of parts with properties and relations in any determinate way. Like the portion of the world it depicts, a picture or sculpture or map may be decomposed into parts, with mutual relationships, in many different ways, which may be significant for different purposes. This can give such representations great flexibility and power. By contrast, a logical formula wears its syntactic decomposition on its face: you cannot understand it at all without knowing how it is to be parsed.

This need not be true of a very large collection of logical formulae, even though it is true of individual formulae. A massive database needs some organisation in order to be useful, and different organisations (at a high level) may be useful for different purposes, even though individual formulae may be uniquely parsed. The clean simplicity of logic may be irrelevant to such global complexity, just as knowing everything about the structure of individual circular dots may be irrelevant to making sense of a newspaper picture composed of dots.

One reason why this sort of ontologically (comparatively) uncommitted representation may be important is that it provides a framework in which learning can take place. A learning system not yet sure of the best way to decompose the world may need to have some way of representing it which is not yet committed to any particular decomposition into objects properties and relations.

'Low level' representations created by visual and other perceptual systems may have this important property. How exactly they are used, how the learning takes place, how a new ontology is formed and represented are all important unanswered questions. Recent work by G.L. Scott at Sussex University (unpublished apart from (Scott 1984)), involves attempting to discover how structure can be imposed on unarticulated data by very general processes which simply attempt to maximise aesthetic qualities, e.g. simplicity, symmetry, harmony. Goodman (1978) also discusses the construction of alternative world-views, suggesting that aesthetic criteria play a significant role. We have yet to understand the trade-offs between totally general ontologically uncommitted learning processes and various kinds of comparatively domain-specific, partly committed learning. Perhaps the former type, like biological evolution, requires millions of years to achieve what the latter can do quickly, at the cost of more stored prior information and restricted generality.

Learning systems and theories which assume the kind of decomposition required by an applicative representation cannot explain how that decomposition is learnt. This applies to many psychological learning theories, to philosophical theories about inductive inference, and to most AI learning theories. In many cases the 'learning' consists simply in trying to find a set of rules which best fits some data where the set of possible rules is constrained by a definite formalism

and ontology. For a survey see (Bundy Silver and Plummer 1983).

The crucial feature of an analogical representation is that instead of using explicit names (predicate symbols, relation symbols) to represent properties and relations of things it uses properties of and relations between parts of the representation itself. This does not require all analogical representations to be totally ontologically uncommitted. For instance, a London underground map is committed to the existence of railway routes, stations along those routes, and to relations of ordering and connectivity. The map also gives a very vague indication of other spatial relations. Maps with dots and other symbols representing named towns, roads, rivers, distances, etc. may also contain a mixture of ontological commitments and uncommitted representation. For instance, there need be no commitment to a particular decomposition of a winding river into parts. An aerial photograph of the same terrain would be even less committed.

Expert systems concerned with diagnosis and repair of equipment may need to use spatial representations which allow different decompositions to be explored in tracking down unusual problems. I once saw a mechanic attempting to divide the engine compartment into regions more and less likely to be affected by the temperature change as the engine warmed up, in tracking down an elusive fault which appeared only after running for a few minutes.

An important task for AI is to study such mixed representations, to understand how they are created, how they are used, and how a learning system can move between different levels of ontological awareness.

11. Perception and ontology

The same issues arise in perceptual systems. The physical world is not intrinsically articulated in any particular way. A perceptual system may have to deliver some sort of articulated, perhaps even logical, representation that can be used as a basis for planning, monitoring actions, forming generalisations, testing hypotheses, communicating with others, etc. As we have seen, in order to derive such cognitively useful representations the system has to have some way of representing and processing information about the unarticulated starting point.

Similar remarks may be made about the need for representations which can guide the behaviour of external objects - arms, legs, wheels, grippers, etc. Many actions, such as catching a ball, throwing a stone at a moving target, tracking an object with one’s eyes, drawing or painting a scene, dancing to music, require both perceptions and actions which involve continuous variation and are closely matched to each other. In simple devices this can easily be achieved by means of mechanical or electronic feedback loops. In humans and other animals there seem to be far more sophisticated and powerful processes, which can improve themselves qualitatively as well as quantitatively. I shall not speculate about the sorts of internal representations required, or the inference, retrieval and matching processes. We understand very little of these matters. Designers of expert systems for real-time control will have to address the problems.

All this raises the question whether there can also be more and less ontologically committed representations of more abstract domains, such as number theory, set theory, computing science, etc. If so, do we need a variety of types of representations to account for learning in these domains, or are the different kinds of knowledge all expressible in FOPL because of the structure of the domain?

Notice that even if everything can be expressed in FOPL, this may be of little use in relation to the task of imposing an organisation on the massive database of fragmentary information. For example, the power of logic would be of little use to a visual system which translated all its

image arrays into logical assertions. The major problems would remain unchanged.

Our discussion suggests that we need to qualify the claim in (Woods 1975) that we need a representation that will precisely, formally, and unambiguously represent any particular interpretation. It depends what you want to use the representation for, and how far your learning has progressed.

12. Explanation and representation

Usually philosophers of science who discuss explanation (e.g. (Nagel 1961)) assume that an explanation is composed of a series of assertions expressed in a verbal or logical formalism. Yet if we examine the cognitive function of explanations, namely their role in producing new insights, a deeper ability to make plans and predictions, to diagnose faults, to form new questions and hypotheses, we find that often an explanation is most usefully expressed in a non-verbal, non-logical form. For instance, seeing a diagram showing the workings of a mechanical clock, or even opening up the clock and looking directly at the cogs and levers, can yield a deeper understanding of how it works and the many ways it can go wrong, than a purely verbal description. Why and how this is so, and what abstract knowledge is presupposed, requires further analysis. I offer it now as yet another familiar example of the power of analogical representations, including the use of something to represent its own structure.

13. Numerical notation

Besides applicative and analogical notations, there are many special-purpose notations. To illustrate further the claim that different sorts of notations may be best for different purposes, we may consider how our ordinary arithmetical notation deviates from being purely applicative.

Ever since the heroic, but unsuccessful, attempts of Frege, Russell and Whitehead to reduce arithmetic to logic it has been clear that there are deep relationships between the two. So it may seem surprising that the notation we regularly use for arithmetic is not predicate logic, but a special purpose formalism, with features specifically designed for their heuristic power in this domain.

Numerals, like ‘‘999’’ are composite formulas whose denotation depends on the denotations of the parts in a systematic way. But instead of having one or more symbols to represent the functions being applied, we use relative positions of the digits. So being n steps from the right (or to the left of a decimal point) represents being multiplied by the (n-1)th power of 10 and added to the running total. This not an applicative notation: the principle of substitutivity is violated. In this notation, you cannot replace an arbitrary argument symbol (e.g. the middle ‘9’ in ‘999’ with any other arbitrary symbol denoting a number (e.g. ‘65’ or ‘3 + 77’), and leave the rest of the expression with the same function-argument relations as before. Moreover, the functions being applied to get the total are not explicitly represented by symbols which can be replaced by other symbols representing different functions. Hence this is not a pure applicative notation.

The invention of the place notation, including a special symbol for zero, was a major intellectual achievement. It enables the notation to do a lot of work for us, when we do additions, multiplications, and divisions. In particular it enables us to use *position* of a digit, at intermediate stages of a calculation, to carry useful information in a very economical form, and it enables us to get by with a small set of primitive numerals in representing all possible integers.

We could, of course, extend our notation using parentheses so that, for example ‘‘9(65)9’’ denoted the same number as

9 x 100 + 65 x 10 + 9

but that would lose some of the economy and power of our existing system, although it might

have other advantages. It would still not be a completely applicative system, insofar as some functions and relations were not represented by explicit names, but by syntactic relations. Our existing notation has 'heuristic power' because of the particular properties of its domain, and the operations we wish to perform in that domain. There are many different kinds of notations which have special features, tailored to the structure of a domain and our purposes.

Diagrams, models, simulations, etc. can play heuristic roles in controlling the search for a formal proof. E.g. don't try to prove subgoals false in the model.

Some of these representations can also be used more constructively -- suggesting a good strategy. For example, if you need the shortest route from A to B, then a good heuristic is to draw a straight line between A and B on a map of the available roads, and then investigate only nearby roads. The heuristic assumes that closeness along roads is represented by closeness on the map. This is not always true, when roads have to go round a large river, for instance.

This method works only because fragments of roads are implicitly indexed by their geometrical relationships, so that one can use geometrical relationships in the map to control the search for roads satisfying a geometrical relationship: being close to the shortest line joining start and end points. This is one of many ways in which a relation of 'nearness' in a representation can be used to represent nearness in the world, to great effect.

14. Intelligence and notations

Intelligence has different dimensions. One is the type or level of competence achieved. Another can be defined as *productive laziness.* It is not only *what* a system can do that determines whether we think of it as intelligent, but also *how* it is done. If methods of blind exhaustive search are used, for example, then that may be useful if the searching is fast enough, but it is not as intelligent as finding a way to avoid the search.

Sometimes finding the right representation for information and problems is a crucial first step -- for instance mapping the well-known chess-board and dominoes problem into a representation involving numbers, in order to prove that a set of dominoes covering adjacent squares cannot cover the board with a pair of diagonally opposite corners removed. Discovering the mapping is made much easier with the aid of a geometric *analogical* representation of the board in which neighbouring squares are given different colours. (Proving that this can always be done with a rectangular grid is not as easy as always seeing how to do it with a particular grid. The general proof requires something like an abstract logical representation.)

Whether it is intelligent to use a particular method may depend on context. For a simple problem with a small search space, the lazy, and therefore the intelligent, strategy may be blind exhaustive search, for instance selecting the right key in a bunch to fit a given lock. When there are many thousands of keys distributed over a variety of shops, it may be intelligent to do some preliminary detailed study of the lock and its properties in order to delimit the search space.

15. Subject matter does not determine best notation.

The *uses* are important, since a domain in itself may be usefully representable in many different ways. For example the London underground railway system may be represented for most users of the system in a fashion which indicates connectivity very accurately, but distances and directions only very loosely. But for the engineers working on the system, and time-table planners, it is important to have a representation which conveys more detailed information.

16. Illustrating the power of diagrams - internal and external

Because they help to control the search space, diagrams are often used in mathematical and logi-

cal reasoning, in planning, in design, etc.

Layout planning (Eastman 1971) often uses a map of the situation, to constrain the set of possibilities to be explored in searching for a configuration satisfying some constraints. Because the representational medium is closely related to what is represented all sorts of possibilities are pruned from the search space simply because they cannot be represented. If a logical or verbal representation were used, there would be nothing in the syntax of the formalism to prevent the impossible situations being described. We now know, from 2-D pictures of impossible 3-D scenes, that Wittgenstein was wrong when he claimed (1922) that it is impossible to represent geometrically the geometrically impossible.

Consider the following problem. In how many points will a 'perfect' circle and a 'perfect' triangle intersect if one corner of the triangle is inside the circle and two outside? How did you solve that problem? If the triangle is entirely inside or entirely outside the circle the number of intersection points is zero. If exactly one corner lies on the circle and the other two outside the circle, the number of intersections is one? (How can you be sure?). How many different numbers of intersection points are possible? Don't read on until you have worked out your answer, and then thought again about whether you have considered all possibilities.

Most people seem to explore a 'space' of possible configurations of the circle and triangle. But hardly anyone does so simply by manipulating verbal or logical descriptions of possible configurations and checking them against axioms for geometry. Instead they seem mentally to construct something which functions like a two dimensional diagram on which they impose geometrical transformations, like sliding the triangle around, making it larger or smaller, changing its shape, etc. It takes most people some time to do this exploration, and not all do it thoroughly enough to find all seven possible numbers of intersections.

Having used a (real or imagined) diagram to explore the problem and identify a likely solution, we may use logic to demonstrate its correctness. But it would not be intelligent to start with a logical representation alone. The burden of constantly referring to explicit geometrical axioms is removed by using a representational medium in which the axioms cannot be violated. This enormously constrains the search space, enabling us to be lazy and productive.

Of course, this does point to a problem of the sort of 'meta-level' representation required in order to infer that all combinations have been tried. I have no doubt that alongside the analogical representations there are very abstract descriptions. Exactly how these should interact is an important research topic. How can a machine be made to represent the process of sweeping through a range of possibilities, subject to constraints? (For some examples see (Funt 1977).)

Alan Bundy informs me me that his equation-solving system, PRESS, cannot find a solution for 'x' in

$$x = \sin(x)$$

though people can. E.g. starting from a trigonometrical definition of 'sin', we can derive the general shape of the graph of 'y=sin(x)', then superimpose the graph of 'y=x', notice that they intersect only near the origin, then argue that it must be exactly at the origin. Notice that visualising the *approximate* shape of the graph is not the same as having an exact image. Neither does it imply that the final result is approximate: a mixture of inference methods can be used to achieve exactness.

How can we be sure we have exhausted all possibilities? In general we can't. See Lakatos (1976) on the history of proofs of Euler's theorem relating the numbers of vertices, edges and faces of a polyhedron. But it is important not to confuse the demand for heuristic power, which is what I have been talking about, with the demand for rigour. Often rigour can come later, though

total rigour is unattainable except in the very simplest domains.

I have tried to indicate why it is not just because of psychological limitations of human problem solvers that it is sensible to use a variety of representational systems in expert activities. This counters the view of some philosophers and mathematicians that mathematics is essentially logic, and that any use of non-logical methods of reasoning by human mathematicians is simply due to their limited intelligence. In some cases the intelligent thing to do is find a special-purpose representation, tailored to the problem domain. Of course, it would be even more intelligent to have a deep understanding of the nature of the representation and the reasons why it is appropriate. It may prove best for this second-order reasoning to use a quite different type of representation, e.g. logic.

17. Using spatial structures in logical structures.

The fact that any visible notation has to be embedded in a medium with its own geometry can blur some of our distinctions. For example, spatial reasoning/perception can be used in analysing a set of axioms, prior to constructing a proof. E.g. seeing a set of implications as forming a sort of 'chain'. The axioms:

P -> Q
Q -> R
R -> S

could be embedded in a larger set of axioms. By taking the ends of the chain we get

P -> S

and similarly for other transitive relations. So what looks like logical reasoning using applicative representations, may in part be geometrical reasoning using analogical representations.

This is ultimately due to the fact that even an applicative logical notation must be embedded in a usable, manipulable, medium. In a structure like 'f(a,b)' geometrical relations between the components are used to indicate the relation of *applying* between the function and the arguments. If an explicit symbol were required for 'apply', as in 'apply(f,a,b)', then for consistency we should require this to be expanded as 'apply(apply,f,a,b)'. Aristotle discovered this infinite regress in connection with the relation between a predicate and its subject and decided this was not really a relation. Without taking sides on that issue we can see that at least the argument shows that if our notation is to be finite and usable there must be a level of representation which is analogical not applicative. The geometric/syntactic structure of a formula can represent analogically the application of a function to its arguments: the application is pictured, not described. If it were always described explicitly instead of being depicted then the unification algorithm used in logic programming languages would have to be quite different, and much less efficient.

18. Conjecture

Human spatial abilities underly many other more abstract abilities, like medical expertise, mechanical or electronic fault-finding, logical reasoning. For instance, the notion of a 'search space' uses physical space as an analogical representation of part of a process of solving a problem.

All known animals which are good at logic, planning, etc. have visual apparatus (even blind people still have the relevant part of the brain). But the converse isn't true. Will either be true of

intelligent machines?

19. Representing processes: simulation vs description

Simulations which run e.g. (Brown and Burton) form a special class of analogical and sometimes mixed representations. They should be contrasted with descriptions from which inferences are made. A collection of equations with algorithms for transforming some of the parameters can be an applicative implementation of a non-applicative, analogical virtual representation, i.e. the running simulation program, in which changes in datastructures or the values of variables represent changes in the thing represented.

Often a simulation of some kind gives the easiest and quickest means of providing an answer to a question about how a system would behave in certain conditions. But a simulation may not produce enough information to answer other questions, for instance about *why* it behaves like that, or what the preconditions are for its behaving like that, or what the range of variation of behaviours would be in a range of situations. For these purposes a more abstract, perhaps more logical, description may be helpful.

A fbw chart can be regarded as a sort of 'frozen' simulation of a certain class of processes: projected from a space/time domain into a two dimensional spatial domain. The relationships are really more complex than this, since the process simulated may have several sub-processes, corresponding to one loop in the fbw chart. A computer program will generally use a still more complex mode of representation, with a mixture of applicative and analogical representations together with a host of special-purposes notational conventions which may affect any of: (a) the process of reading in program text, (b) the process of compiling to a lower level language, (c) initialisation processes and (d) the process of running the program.

Natural languages use an even more complex mixture of representations, especially in spoken forms, where stress, intonation, volume and tempo may all interact with each other and with the words selected. Often mixed modes are used, e.g. sentences where time order or spatial order is represented analogically, along with explicitly named properties and relations.

20. Virtual machines and virtual representations

Often we seem to use objects in our minds which are like objects which exist in the physical world. A visualised map or diagram may be used for some of the same purposes as a real physical one. Sometimes an external physical map will be easier to use, because it is more detailed than an image, more stable, more easily traversed in all directions, and can have a different range of operations applied to it, for instance laying a ruler or other cut-out shape on it. Nevertheless, the status of mental maps, diagrams, models, is of considerable interest.

Introspective reports are not to be taken too seriously -- though they are often suggestive evidence. People often say they use a picture or image in performing some task. But they can't *just* use a picture or image. E.g. as they investigate relationships they must be making use of some specifi cation of constraints (e.g. it must remain a triangle even though its shape changes.) The constraints may exist only in a compiled form -- another type of representation.

Moreover, even the claim to be using a mental picture or diagram cannot be taken literally. A literal mental picture would require a mental eye to look at it, and it would presumably produce its own 'internal' mental picture which would require another mental eye to look at it.....

One answer to this is to acknowledge that one sort of representation may be *implemented* in terms of another quite different sort, just as a computer may be a 'virtual machine' implemented by software or microcode in terms of some lower level machine. For instance a lot of picture-like representations in computer programs use two-dimensional arrays, which are actually

represented at a lower level as a one-dimensional array or vector, and at still lower levels as complex patterns of switch states. What makes us justified in talking about a 2-D array is the availability of *procedures* which produce operations best interpreted in 2-D terms, such as scanning a row, or a column, of the array, or scanning all eight neighbours of an array element. In principle the array could even be implemented in terms of a logical database, and array operations implemented in terms of logical deductions. This would be quite acceptable as a lower level representation, if the logical virtual machine ran fast enough. (Very very fast!). Hayes (1974) made this point by referring to the need for an underlying medium in which a representation is embedded.

My point is that one can discuss the heuristic and other properties of a representation independently of how it is actually implemented -- and it may have totally different properties from those in an underlying virtual machine.

Of course, a poor implementation may have features (e.g. excessive space or time requirements) which undermine the advantages of the virtual representation -- a common trap for programmers unaware of the lower levels of the systems they use.

So when introspection suggests that we are using a certain sort of representation, the properties of that representation may be achieved by implementing it in terms of a quite different representation to which we may have no introspective access at all. Some of the differences can be brought out by simple experiments. For instance, a person who claims to be able to visualise written words will often find that he can read the letters off his image much faster from left to right than from right to left. This is not the case when the letters are in front of him on paper. Perhaps the discrepancy is due to the visual image being implemented in terms of list structures, or similar chains of binary associations between objects and the rest of the list. We have already noted that a list may function as an analogical representation of an ordered set of objects.

AI systems also seem to need a variety of layers of representations. Von Neumann computers seem to be well suited to this; will the same be true of other novel architectures, e.g. declarative machines?

21. Criteria for assessing a notation

Chomsky (e.g. 1965) distinguished several kinds of adequacy of grammars and grammatical theories. An *observationally* adequate grammar for a natural language generates all and only well formed strings of the language. A *descriptively* adequate grammar also assigns parse-trees which accord with the way users understand the language. *Explanatory* adequacy of a grammatical theory (for Chomsky) is concerned with the ability to account for how a language is learnt.

Chomsky's distinctions do not address the question whether one notation or grammar is more useful than another for the purposes of an intelligent language user, for he claimed not to be concerned with processing. But the attempt to formulate criteria for evaluating grammars was a precursor of the important task of formulating criteria for assessing formalisms for use in Artificial Intelligence. It is very important, however, that we distinguish two major roles, namely the use of a formalism in a working system and the use of a formalism for theorising about a working system and its task domain. Building explanatory theories requires a more abstract, more logical, language than building a working model or simulation. The requirements are quite different. For instance a working system has time constraints. Theoretical discussion may have quite different constraints, or none at all. A working system merely has to represent or replicate a class of behaviours. A theory has to say something about the relationship between those behaviours and others not produced, requiring a much higher level of abstraction.

Criteria for adequacy of representations used for a *visual* system were discussed by Marr and Nishihara (1978) and Marr (1982). They consider such things as whether the representation is

easily *accessible*, i.e. readily computed from available input, *general*, i.e. able to cope with a range of cases, *uniquely determined* by the input, *stable*, i.e. resistant to changes of view or lighting, *sensitive*, i.e. able to detect and indicate small differences between scenes. These criteria (especially the last two), may conflict, and any selection will generally involve a trade-off. They did not discuss many other relevant criteria. For instance, their hierarchically organised representation makes it hard to represent spatial relationships between arbitrary parts of a structured object. So one finger can be related to another on the same hand quite easily, but not so easily to a finger on the other hand, or to the nose it is scratching. Criteria relevant to choice of a representation used for recognising objects may not be relevant to the goal of avoiding collisions with them, or the goal of picking them up without damaging parts. Of two tasks, one may require the representation of far less detail, and quite different spatial relationships.

For instance, it is an error to suppose that all the uses of vision require a representation of 3-D structure. Much motion control, for example, can efficiently be based on the monitoring of 2-D image structure.

Woods' valuable essay on semantic nets (1975) discusses criteria for assessing them. However, he seems to assume throughout that what needs to be represented is what logic represents exactly. I have shown that this may not always be the case.

McCarthy and Hayes (1969) introduced three sorts of criteria more relevant to evaluation of general knowledge representations. Their criteria partly echo Chomsky's distinctions, perhaps unintentionally. Consider an agent A using a language L in a world W.

(a) Metaphysical/Ontological adequacy.
Can L express everything that can be the case in W?

(b) Epistemological adequacy (relative to agent A)
Can L express everything which A needs to know about W?

(c) Heuristic adequacy.
Does L facilitate the *Processing* that A requires, better than alternative languages L1, L2,..., for representing the same world, W.

These criteria were presented as if they might be absolute. That is, how the world actually is, and what A needs to know about it, and the purposes for which A needs the knowledge are assumed fixed, and then different languages are discussed. But we have seen that how A needs to construe the world may depend on how much A has already learnt, and what tasks or problems he has. So quite different representations may be needed at different times. This does not, however, rule out a general theory of what the relationships are between purposes, types of environment, and useful representations.

Readers interested in exploring these issues should try formulating criteria for selecting a notation for numbers. If the criteria include easy learnability, and other cognitive criteria, then the first few Roman numerals may do quite well. However, once there is a need to represent infinitely many integers or to multiply and divide large numbers; a different sort of notation becomes desirable.

A more detailed analysis of criteria for assessing representation schemas would have to be far more careful than anything I have seen so far in the AI literature. It would have to include a survey of the different purposes for which formalisms may be used. For instance we have already seen that the following two uses may be incompatible:

A. theorising about properties of a domain

B. programming something to act intelligently in the domain

These generate very different requirements. I suspect that further investigation will reveal a host of different sorts of criteria, and that there will often be conflicts to be resolved by a systematic analysis of trade-offs.

22. Some problems with FOPL

FOPL is rich, powerful, and the most general language we have. But it is far from unproblematic. There are many difficulties which I am not going to have time to go into in detail. Here are a few old problems.

(1) Is *first* order logic enough? E.g. ''Napoleon had all the qualities of a good general''

(2) Can actions/real relationships be adequately represented? This raises the problem of indefinite qualification. In English we seem to be able to say things which are indefinitely expandable in ways in which an assertion using logic would not be.

Bill hit Joe
with a fish
last Thursday
on the head
hard
to hurt him
at the market

One common answer is implicit in the use of case grammars, but was originally suggested by Donald Davidson, I believe. This is to extend the ontology, to include entities called actions with a variety of properties and relations to other entities.

act(a) & type(a, hit) & agent(a, Bill) & object(a, Joe)
& instrument(a, x) & fish(x) & time(a, Thursday 5th Sept),
& application_point(a, head(Bill)), & force(a, hard)....

Would a logical formalism allowing variadic predicates be better? How would its semantics be defined?

(3) Problems arising out of the restriction to denotational (extensional) semantics have been discussed above.

23. Conclusion

I have tried to indicate, though in a sketchy and incomplete fashion, some of the reasons why we need to explore the uses of different sorts of formalisms for different purposes. We need to understand how an intelligent system can choose between different formalisms, and how it can, on occasions, create new formalisms when doing so would give new insight or heuristic power of some kind. The discussion suggests that the design of really intelligent systems is going to be a very difficult and very complex task. If only Kowalski were right!

Acknowledgement

Part of the work reported here was supported by a fellowship from the GEC Research Laboratories. Despite our disagreements, I, like many others, have learnt a great deal from Bob Kowalski.

REFERENCES

Bobrow, D. (1975). 'Dimensions of Representation', in D.Bobrow and A.Collins (eds) *Representation and Understanding*, Academic Press.

Brown, J.S. and Burton, R.R. (1975). 'Multiple Representations of Knowledge for Tutorial Reasoning', in D.Bobrow and A.Collins (eds) *Representation and Understanding*, Academic Press.

Bundy, A, Silver B, and Plummer D. (1983). 'An analytical comparison of some rule learning programs' *Proceedings British Computer Society Expert Systems Group Conference*, Churchill College Cambridge.

Chomsky, N. (1965). *Aspects of the theory of Syntax.* MIT Press.

Eastman, C.M. (1971). 'Heuristic algorithms for automated space planning' in *Proc. 2nd IJCAI* British Computer Society.

Funt, B.V. (1977). 'WHISPER: a problem solving system utilizing diagrams and a parallel processing retina', in *Proceedings 5th IJCAI*, MIT.

Goodman, N. (1969). *Languages of Art* Oxford University Press.

Goodman, N. (1978). *Ways of worldmaking*, Harvester Press.

Hayes, P.J. (1974). 'Some problems and non-problems in representation theory', in *Proc. AISB Summer Conference* University of Sussex, 1974 (out of print).

Johnson-Laird, P.N. (1983). *Mental Models*, Cambridge University Press.

Kowalski, R.A. (1980). contribution to SIGART newsletter No 70, *Special Issue on Knowledge Representation*, Feb. 1980

Lakatos, I (1976).. *Proofs and Refutations*, Cambridge University Press.

Marr, D. (1982). *Vision* Freeman.

Marr, D and Nishihara H.K. (1978). 'Representation and recognition of the spatial organization of three-dimensional shapes. *Proc Royal Society B200*,

McCarthy, J. (1979). 'First-order theories of individual concepts and propositions', in D. Michie (ed) *Expert Systems in the Microelectronic Age* Edinburgh University Press.

McCarthy, J. and Hayes, P.J. (1979). 'Some philosophical problems from the standpoint of Artificial Intelligence', in *Machine Intelligence 4*, ed. B. Meltzer and D. Michie, Edinburgh University Press.

Nagel E. (1961). *The Structure of Science*, Routledge and Keegan Paul.

Scott, G.L. (1984). 'Obtaining the structure(s) of a non-rigid body from multiple views by maximising perceived rigidity' in *Proc European Conference on AI*, Pisa.

Sloman, A. (1965) 'Functions and rogators', in J.N. Crossley and M.A.E. Dummett (eds) *Formal Systems and Recursive Functions*, North Holland.

Sloman, A. (1971). 'Interactions between philosophy and artificial intelligence', in *Artificial Intelligence 2*,

Sloman, A. (1978). *The Computer Revolution in Philosophy: Philosophy Science and Models of Mind*, Harvester Press and Humanities Press.

Sloman, A. (1979). 'The primacy of non-communicative language' in *The Analysis of Meaning, Proceedings 5th ASLIB Informatics Conference*, ASLIB, London.

Wittgenstein, L. (1922). *Tractatus Logico Philosophicus*, Routledge and Kegan Paul.

Woods W.A. (1975). 'What's in a link: Foundations for semantic networks', in D.Bobrow and A.Collins (eds) *Representation and Understanding*, Academic Press.