AART BIJL

EdCAAD, University of Edinburgh

Ai in architectural CAD

ABSTRACT

Nous explorons quelques conceptions dans l'intelligence artificielle et CAO ; les suppositions de la connaissance overte, des croyances et de l'intuition - illustrés par l'expérience du CAO. En changant notre suppositions, il est suggéré que nous peut-être achievons une compatibilité meilleure entre les ordinateurs et les designateurs.

We explore some concepts in artificial intelligence and computer aided design; the assumptions of overt knowledge, beliefs and intuition - illustrated by experience of CAD. By changing our assumptions, it is suggested that we might achieve better compatibility between computers and designers.

AI IN ARCHITECTURAL CAD

Computer aided design expresses an intention, a goal. Can computers aid designers; can they provide assistance to people when they design things, buildings, architecture? Are the assumptions that underpin computer technology compatible with the activity that we recognise in people when they design? These are the questions that will be explored in this paper. If we then also recognise that design activity typifies much of human participation in many non-design fields, these questions become vitally important to the future of computers and all people.

Both computers and design refer to kinds of mental activity, but with important differences. Computers can be seen as part of a long tradition of human endeavour to externalise thoughts, so that thoughts can be passed between people in the form of overt knowledge. This tradition embraces the objectivity of natural sciences, the detachment of mathematics and the truths of formal logic. It deals in the kind of knowledge that can be conveyed entirely by means of abstract symbolic expressions, such as words and numbers. It has now come to be accepted as setting the ground-rules for our rationalisations and our justifications for all things we *do*. Computers are the latest development in this tradition, promising environments in which knowledge can be represented independently of people, and which can operate upon knowledge autonomously on behalf of, or in place of people.

Design, on the other hand, differs in the way that it employs people's intuitive knowledge, embracing all kinds of human experience, calling on holistic involvement of people beyond their exercise of formal, overt knowledge. Design relies on individual judgment, to tell us what we *want* to do. We can emphasise this difference by adding that design activity calls on human sensitivities, on powers of perception and interpretation, that are not bounded by conventions governing the correctness of symbolic expressions. Design tends to employ analogic forms of expression, involving interpretation of people's concerns directly into material artefacts or, more generally, in decisions that affect our exploitation of our material world. Correct or, more precisely, good designs are decided, never proved.

We need to consider the nature and the limitations of overt knowledge, and the effects of encapsulating such knowledge within computers, with respect to people's intuitive abilities.

THINKING IS BELIEVING

Already, the previous paragraphs raise contentious issues. Our perception of the difference between overt and intuitive knowledge is fundamentally important to any view of computer aided design. Can we be sure about any distinction that we draw between these two forms of knowledge? Here we are faced with a philosophical question and we have to declare our belief. To say that there is no difference and that all human mental activity is capable of being represented as overt knowledge, so that it is potentially capable of being reproduced wholly within computers, seems a highly implausible assumption. To say that people, as beings, possess certain abilities that are inherently not reproducible within machines, is a more plausible and more interesting assumption. Accepting the latter, we might agree that computers can be made to exhibit behaviour that looks like the behaviour of people, but the two would not be the same if we also consider deep motivations and the role of judgment in validating the behaviour.

Intuition embraces that part of knowledge that we are unable to express wholly in an abstract world of symbols, and we assume that it cannot ever be so expressed. This assumption presents us with some difficulties, not least being the strongly established position of the opposite assumption. An inherent difficulty is, somewhat paradoxically, that the rules which govern the acceptability of written argument conform to the tradition of overt knowledge, producing a bias against reasoned explanations of intuition - it is difficult to talk about intuition. Intuition might be equated with belief, but this carries the implication (correctly) that it is unreasonable and therefore (wrongly) that its products are unjustifiable. We need to add that intuition can also be equated with feeling, employing all stimuli received through all the human senses, producing a holistic sense of knowing what is right and good. Unavoidably, any discussion of intuition has to embrace personal interpretation of personal experience - in my case, my years as a designer and as a researcher in computer aided design - and interpretations have then to be related across a broad spread of established overt knowledge.

The difference between intuition and overt knowledge need not be viewed as mutually exclusive. It can be argued (Bijl 1986) that overt knowledge is built upon people's collective intuitions, providing a basis for concerted mental activity that does not need to be checked out by individual intuitions for each new instance of application. Overt knowledge, seen in this way, has provided the foundation for our dramatic technological advances. It does not, however, tell us how we should exploit these advances. To inform ourselves of what we want to do, we have to supplement overt knowledge with intuition. Here is where design comes into play - design is about deciding what you want to do, rather than doing what you already know you have to do. In fields of design that demand innovation, we find that overt knowledge does need to be checked against individual intuitions - design activity is characterised by its overt dependence on intuitive knowledge, and especially so in the field of architecture.

To express this point succinctly and more generally, we can say that all overt knowledge is founded on assumptions and that assumptions are expressions of belief. Overt knowledge becomes established when many people share the same beliefs, and knowledge becomes undone when beliefs are challenged. We might go further and say that there is no intrinsic difference between overt knowledge and any other kind of expression of belief - the present dominance of overt knowledge would then have to be viewed as reflecting our present adherence to a certain kind of belief. We might be impressed by the ways in which we are able to build upon assumptions to arrive at predictions that conform to our perceptions of world phenomena - our familiar scientific method - but this ability remains dependent upon people's beliefs.[1]

The mechanics of thinking

Overt knowledge, as already noted, is used to pass information between people. We should add that such information has to be capable of being represented externally to people and has to pass between people so that, ideally, one person's knowledge can be recreated in another person. A test of the overtness of knowledge is that its existence can be recognised and validated by other people, without the author of the knowledge being present. A test of the correctness of transferred knowledge is that

[1] For a useful and easily understood presentation of modern physics, which discusses its uncertain foundations and philosophical implications, see part 1 and the afterword of Capra (1983). If a more authoritative reference is needed, consider the words of Einstein (quoted in Johnson-Laird 1983):
"Physical concepts are free creations of the mind, and are not, however they may seem, uniquely determined by the external world."

the author recognises that the recipient has come to share the same knowledge. In both cases, the tests have to focus on people's interactions with overt expressions of knowledge.

These conditions on overt knowledge are problematical, so much so that it is a bit surprising that so many people do actually believe that all the problems are potentially resolvable. Can we ever be sure that any two people share exactly the same knowledge? Can we ever be sure that any overt expression of knowledge exactly represents knowledge possessed by any one person? Even if we resort to notions of partial knowledge, are exactly the same boundaries to parts recognised by different people? Yet, the problems posed by these questions are the kind of problems that have to be resolved if computers are to play an active part in representing and manipulating people's knowledge.

Development of overt knowledge is entirely dependent upon our definition of symbols and permissible relationships that can exist between symbols, plus operations for transforming relationships. Symbols are simply abstract things that we agree to recognise as standing for other things that we (variously) perceive in our world - predominant examples of symbols are words and numbers. Symbolic environments are used to represent what we know about the world, and these representational environments or abstract *formalisms* provide the fundamental means by which we can manipulate overt knowledge. Thus we have variants of symbolic logic, and mathematics. The intention is that formalisms should achieve generality so that they can be used to build systems that correspond to various things that people have in mind. The systems should support the development of models that describe particular things, objects, actions and events, that can be appreciated as useful applications in human worlds.

At this point we should note that artificial intelligence seeks to encapsulate knowledge and human intelligence wholly in machines - an apparently logical extension of the development of overt knowledge. The implications will be discussed later.

Within any one formalism, we are able to do things and we can assess what we do - we may be able to establish, for instance, whether a proposition is true or false within the bounds of the formalism. We can regard formalisms as mental mechanics for thinking, enabling us to express what we think overtly. Like physical mechanics, such mental mechanics conditions how we can do things, and what we can do. In neither a physical or mental sense does the mechanics itself justify what we actually do. That actuality is governed by human considerations outside the mechanics, which prompts our development of new mechanics. We should expect formalisms to evolve in response to our changing perceptions of our world.

An appreciation of formalisms, the mechanics of thinking, has general importance. How we think, conditions what we can do - our understanding of our world, our place within that world, and what we do with it.[1] More specifically, we need to focus our attention on the nature of formalisms because of their relevance to design and to computer aided design. We have to answer the crucial question: what interaction can exist between formalisms operating within computers and the informal behaviour of designers who are invited to use computers?

[1] Here I refer to the kind of questions posed by Schumacher on technological know-how, what he calls bad metaphysics, and the need to know-what (Schumacher 1974).

We do as we think

Now let us consider design. We can choose to look upon design as something people do. Designed artefacts, the products of designing, are interesting only insofar as they tell us something about design. An extreme expression of this position is to say that the world of design is the thoughts in the heads of designers, plus the skills of designers in externalising their thoughts; designed artefacts, once perceived and accepted into the worlds of other people, are no longer part of the world of design.

We can describe design, briefly, as a process of synthesis. Design has to achieve a fusion between parts to create new parts, so that the products are recognised as having a right and proper place in the world of people. Parts should be understood as referring to anything, physical objects, abstract ideas, aspirations. These parts occur in some design environment from which parts are extracted, designed upon, and results replaced; in the example of buildings, the environment is people and the results have to be judged by reference to that environment. It is characteristic of design that both the process and its products are not subject to explicit and complete overt criteria.

This view of design differs sharply from the more orthodox understanding of scientific and technological endeavours which rely predominantly on a process of analysis. In this latter case, the approach is to decompose a problem into parts until individual parts are recognised as being amenable to known operations, and results are reassembled into a solution. This process has a peripheral role in design, when evaluating selected aspects of tentative design proposals, but the absence of well-defined and widely recognised criteria for design excludes it from the main-stream of analytical developments.

We can identify the key characteristics of design (figure 1) as:

a) design objects - subject to diversity of expression; different perceptions of things; lack of agreed abstract definitions;

b) design processes - not orthodox problem solving; conflicting criteria for validating results; many solutions - more aptly described as event exploration;

c) design knowledge - no formal and shared knowledge base; relies on integration of overt knowledge and intuition; necessarily manifest in idiosyncratic design practices.

Knowledge of designers, what each knows, can be viewed only in the external manifestations of their thoughts, in the drawings and words of designers. Intelligent behaviour is then perceived essentially as the ability to externalise thought, to give expression to both overt and intuitive knowledge (figure 2). Good design occurs in the eyes of other people who see a designers' externalised expression of thought which, in turn, is built on the designer's perception and interpretation of the values of those other people. This understanding of design places responsibility within people and focuses research on the formal expressions that pass between people.

This view of design is not intended to support the notion that design is somehow mystical. Instead, and quite simply, it recognises a distinction between abilities of people that can be represented as overt knowledge, and the different abilities that remain within individuals as intuition. Design objects include overt expressions of the products or conclusions of individual designers' intuitions. If we accept this view of design, we then have to face important consequences for computer aided design. We have to envisage systems that can accommodate individual expressions of intuition.

PAST AND PRESENT CAD SYSTEMS

The following paragraphs will outline experience of three kinds of computer application: integrated design systems, function orientated systems and drawing systems. This will not be a catalogue of available systems. Instead, we will identify characteristics that tend to be common across different systems, giving emphasis to relationships between the users' perception of their world, and the representation of that world within a computer application.

Integrated design systems

An *integrated design system* is one which employs a single model that can accommodate all information describing a design object, corresponding to different knowledge supplied by different people. The model then has to be capable of supporting a range of operations on the same description, to advance people's different interests during the course of designing the object. In addition, the system has to be capable of interpreting information from people's overt expressions of their thoughts about a design object, from text and drawings, to supply that information to the model. This goal focuses attention on the possible ways of organising data within computers, a data-orientated approach. During the 60s and early 70s this goal was recognised as obviously relevant to design and, with the courage of ignorance about the fundamental questions it posed, we tried to achieve integrated design systems.

The model of design used in the integrated systems approach may be described as:

a) a design is a single coherent description (of a building) that can supply information for many varied design tasks;

b) any part of a description may be defined by any other parts;

c) any part may serve more than one task.

One early example was the SSHA system for housing design (Bijl 1986 and 1979), developed at Edinburgh, and used by the Scottish Special Housing Association for more than ten years. The system employed graphics to describe the shape properties of a design object, and text expressions to describe other properties of the graphically depicted parts. The system saw buildings as composed of components which, in turn, were described as compositions of stretchable "thick slabs" - each component being described in situ, in the context of each design. The user would assign general materials specifications, excluding boundary conditions. The system would then look at spatial relationships between component parts to identify the necessary junction details, to complete each description. The faces of components that were not junctions were recognised as surfaces to room spaces that were bounded by components, as shown in figure 3. Essentially, the model recognised three kinds of entity: junctions that equated with construction details, components that equated with construction components; spaces that equated with room spaces.

Component entities were described by name (wall, partition, etc.) plus attributes; primary material; internal cladding; external cladding (for external components). These entities were related to junctions; construction details could have the effect of modifying component material quantities. Spaces were described by name (living room, kitchen, etc.) indicating function, plus attributes; furniture content; surface finishes. Spaces were related to components through surfaces: By describing instances of components in situ, changes to a component could propagate consequential changes to adjoining components. This system did not rely on the notion of discrete components. Thus, for example, room spaces that were described by boundary components were recognised as further components, and the "junctions" between a room space and a boundary component could be modified from the room. The surfaces to some or all the components bounding the room could be altered in this way, in turn propagating changes to the material descriptions of the boundary components.

By focusing library information on construction details of junctions between components, this system could operate with a very much smaller library of pre-defined information than was usual. Each detail could be used to describe many different components. Coupled with the non-discrete nature of components, these advantages were gained at the expense of more complex system design and implementation.

Despite the success of this system in practical use, it did not mark the beginning of a new era in CAD. The most serious problem that was presented by early integrated design systems, which persists to the present day, is that they relied on a very close correspondence between a user's design practices and the modelling procedures built into the system, and those procedures had to be defined at the time of developing the system. Predefined computer models had to anticipate precisely how buildings would be perceived in a user-world, down to a level of fine detail. Users had then to conform to these anticipations in order to use the system. We were then, and commonly we still are, exploiting a prescriptive computer technology. The explanation of why this is unacceptable, was formulated by the authors of the SSHA system in the mid 70s (Bijl et al 1979):

> "In general, our view of design practices suggests that design is not a knowledge based discipline, but evolves from the experience of many individual practitioners. By this we mean that the knowledge used by designers cannot, on its own, be constituted as a formal model which will then be recognised by all designers as representing their own individual practices. The role of experience is to build up the responses of designers to the tasks which other people present to them, and experience results in unique combinations of formal knowledge and subjective understanding within individual designers. This conclusion has profound implications for future, generalised CAD systems."

Function orientated systems

A *function orientated system* is one in which an anticipation of some specific task in a user-world forms the primary motivation for developing the computer system. An example might be calculations required for environmental appraisals, to assess performance of a proposed design for a building. Anticipation of function is paramount, providing a specification of required system operations, and organisation of data is subservient to the function. Typically, these systems perform discrete functions, each requiring separate input and providing output that users have to translate into their own perceptions of design.

The model of design used in a function orientated approach may be described as:

a) designing is a process that consists of tasks which people apply to things;

b) any whole design can be analysed into discrete parts that take their definitions from bounded tasks;

c) decomposed parts can be categorised as quantitative, amenable to computer processes, and qualitative, subject to human judgment and decision;

d) design products are some kind of summation of the results of tasks applied to parts.

The philosophy behind this approach to systems is illustrated by the work of ABACUS at Strathclyde (figure 4). Design is seen as an iterative process going through cycles of designing (hypothesising), evaluation and appraisal, looking at the results, scratching your head and deciding what to do next. This process is essentially sequential. The figure includes an example of this approach applied to a design of a small building, with results appearing as a graph indicating numeric values for total floor area, percentage educational space, planning efficiency and so on. Each of these results refer to separate criteria, and the nearer they get to zero or plus values so it is believed that the design is a better design. These systems can use sophisticated simulations and analytic functions, so that computers increasingly may be regarded as behaving like the consultants that architects might work with, looking at specific aspects of a building design. The idea is that the knowledge that such people have should be encapsulated as formulations of tasks inside a computer.

Problems are presented by the boundedness of discrete tasks, with consequent notions of classification and typing, that have to correspond to different users' perceptions of design. The boundaries between one task and another will vary for different users and, more seriously, the boundaries between quantitative and qualitative functions cannot be stabilised for different instances of design.

A possibly more serious problem is presented by the intention that function orientated systems should encapsulate specialist knowledge inside a computer program so that a designer can apply such knowledge without requiring the presence of the human specialist. The general position can be typified. In normal practice, a designer consults a specialist in order to gain access to that person's specialist knowledge. A response will refer to the specialist's bounded discipline and it may be expressed in numbers. The designer then has to receive the answer and make it fit in with all other considerations that bear on the design of a building. The building is not going to be defined in terms of a single criterion, such as environmental behaviour, it also has to do many other things. The answer may not fit. The designer then argues with the specialist, calling on the specialist to break the rules of his formal discipline and refer to his trained intuition. Essentially, the designer negotiates with the specialist for a wrong but acceptable answer. This bartering process is typical in design but it cannot be accommodated in orthodox function orientated systems.

Drawing systems

Following on the experience of integrated design systems during the 70s, and the limited success of function orientated systems, new CAD developments of the late 70s and 80s became far less ambitious. This trend was marked by the emergence and the popularisation of *drawing systems*. These systems enable people to use computer graphics to produce drawings, so that the drawings are the end product of using computers. The key difference to earlier integrated design systems is that, in the case of drawing systems, the computer knows nothing about what is being depicted in drawings; the computer knows only about lines and points, and edits and transformations that can be applied to collections of lines and points. The goal of these systems is to produce drawings faster and of better quality than might otherwise be produced by hand.

Lest this characterisation of drawing systems is challenged as being too exclusive, it should be added that some drawing systems variously do have limited abilities to handle other information associated with drawings. A typical example is the use of solid geometry modelling to support drawing projections of three-dimension objects, and another example is the provision of schedules by means of quantifying instances of parts of drawings. Where such "intelligence" is included, the problems described previously for integrated and function orientated systems begin to re-emerge. In this discussion we will limit ourselves to the more strict characterisation of "dumb" drawing systems that know nothing about what they might be depicting (figure 5).

The model of design that is implicit in current drawing systems may be described as:

a) design is manifest through drawings;

b) thinking about things during the course of designing can be separated from the act of drawing those things;

c) systems for producing and editing drawings can achieve the generality of word-processors.

The close parallel between dumb drawing systems and word-processors is significant. Apart from specialist fields of computer application, word-processors are the one big success story of computer technology. This success rests on the fact that word-processors are dumb; since they do not know the meaning of any words which they process, that knowledge cannot conflict with the different knowledge intended by any users. However, since there are similarities in the ways that all people operate with words (or so we believe), irrespectively of the different things that people intend to say with words, word-processors can exploit these similarities and thus they can be useful to many people. Coupled with the trend towards lower-cost/higher-power computers, this characterisation of word-processors explains why they can be targeted at mass markets. The idea for dumb drawing systems, then, is that they should follow in the path of word-processors. To do so, there have to be equivalent similarities in the ways that all people who draw, operate with drawings.

To expand on this point briefly, let us consider written words as symbolic objects and line drawings as analogic objects. Written words, made up of characters, can be thought of as graphical objects. However, the symbolic role of words has the effect that it does not matter too much how they are drawn - the depiction of a symbol does not have to look like its meaning. Quite a lot of distortion is permitted before the symbol that a written word depicts is changed or lost. Note that this tolerance is allowed mainly for individual characters, and less so for arrangements of characters within words and arrangements of words within word-strings. This tolerance is evident in the range of different type fonts used for printing, and in the variety of people's handwriting - and our ability to read will also tolerate some misspelling and even incorrect grammar. Indeed, such variations may be interpreted as intended expressions of intonation or emphasis that might add to the symbolic content of otherwise regularly formed (or drawn) words. The more we attach significance to the actual depiction of word symbols, so the word objects become pictorial objects.

This transition from symbolic object to pictorial object can be observed through a range of drawing objects from word characters, through idiographs and diagrams to verisimilar (or pictorial) drawings. Figure 6 provides an informal presentation of such a range. What distinguishes a pictorial object from a symbolic object? A verisimilar drawing is one in which the properties of shapes contained in the drawing correspond to properties of some other object which it depicts; the drawing provides a visual semblance of some perception of actuality. We can say, therefore, that a verisimilar drawing serves as an analogue for the thing it depicts. To put this point more strongly, from the point of view of the person who makes the drawing, a verisimilar drawing is the thing it depicts in the sense that it is what the person knows about the depicted thing. In this sense, a verisimilar drawing is the opposite of a symbolic object.

Equating pictorial objects with verisimilar drawings, our definition has implications on the ways in which pictures can be produced. The form of a drawing, its composition of shapes and the details of each shape, become vitally important in expressing what its author has in mind and, therefore, to the content that can be read from the drawing. We should expect that the process of drawing is idiosyncratic to the author, calling on both overt and intuitive knowledge that he may possess, to get at all he knows of the thing he is depicting. We should also expect that the process of reading the drawing will not result in the same knowledge as that possessed by the author; the reader will add his own intuition to what he sees.

A problem presented by all drawing systems arises from the deliberate separation of the act of drawing from thinking about things, buildings, that are depicted in drawings. Learning to use a system, becoming fully familiar with its user-interface, requires substantial dedicated effort - it is not unusual for an architects' office to spend a year learning how to exploit the system, before the office is able to produce practical working drawings. More critically, systems that are powerful enough to produce fully detailed working drawings of all kinds of building design have tended to be very expensive - offices have spent in the region of £200,000 in acquiring their systems.

This combination of difficulty and high cost has resulted in a trend, within the UK, of offices which form companies to operate their drawing systems and sell drawing services to other offices. Returning to our earlier consideration of design activity, if it is accepted that the contribution which architects make to designs is made visible by their drawing activity, that drawings show what architects know about buildings, then this trend towards specialised drawing services offered from outside an architects' office presents a very strange situation. If drawing operations come to be regarded as a kind of consultant activity, what then remains of the role of architects?

NEW GOALS AND TECHNIQUES

Experience of computer aided design points to inadequacies in prescriptive computer technology. There are fundamental incompatibilities between the demands that people make on designers, and the expectation that designers' work-practices should conform to predefined procedures set up within computer systems. Formal representational schemes that depend on prescribed models of user-worlds, of buildings as perceived by architects, are inherently prone to conflict with the varied and valid perceptions of different architects. Prescribed forms of expression, particular modes of text and graphics, that are defined by the need to access predefined organisations of knowledge held within computers, are incompatible with the ways in which architects have to think and express their thoughts, when designing.

These issues cannot be resolved simply by staying within the same technology and writing better programs. Even the change in technology that is marked by the shift from deterministic algorithmic processes to rule based systems is not likely to prove sufficient. This latter development is evident in Expert Systems and is identified with the field of artificial intelligence. Can we identify the more fundamental concerns that motivate AI, and are they relevant to our ambitions in CAD? Should we expect most help from those efforts in AI that are focused on natural language.

Help from AI

The field of AI covers a broad spectrum of interests loosely linked by a shared ambition to represent more of human intelligence in machines, or to achieve machines that exhibit intelligent behaviour. This ambition spans a range from hard AI, aimed at machines that can act autonomously and in place of people, to soft AI which expects machines to be responsive and supportive of people, to aid the actions of people. The assumption underlying hard AI is that all human intelligence potentially can be represented by computational processes operating in symbolic environments - soft AI is less sure.

This position is interesting because of its similarity to the dichotomy in CAD in the late 60s and early 70s, between those who believed that design generation could be achieved wholly by quantitative processes, and others who believed in a separate and necessary human responsibility for qualitative aspects of design. The important observation is that this division did not lead to a stable definition of a quantitative role for computers - the demarcation between quantity and quality could vary for each instance of design.

The equivalent problem for soft AI is the distinction between computational processes and other abilities that might contribute towards human intelligence, and the variability of demarcations that might be employed by different people. This problem will be revealed when AI systems are tested on practical applications - appreciation of this problem is already implicit in the tendency for AI researchers to deny the definition of design that is indicated in this paper.

Formal approaches to natural language seek ways to represent things that people have in mind, from their use of language (verbal, by voice or text), and to match this with some representation of a domain in a person's world. From a design point of view, our interest in natural language stems from that field's attempts to represent what it is that people have in mind, however they wish to express it and with little anticipation of what it might be. This ambition is appealing and the problems it presents are immense. Progress is by no means straight-forward and the chosen underlying strategies remain subject to destructive questioning. For example, these formal approaches rely on the real world assumption, but little is said about how the status of reality is perceived or comes about and how it gets to be represented. Yet the role of formal representations of things in a real world is usually regarded as central to the act of "understanding". Verification of language expressions against a world domain gets increasingly more complicated when meaning of expressions has to reconcile states of different worlds in the mind of the person uttering expressions and in the minds of other people receiving the expressions, given their various perceptions of some external domain. Still more critically, it is also argued that people do not necessarily work with representations of the world and that, instead, their knowledge is more structurally determined, like mental structures that produce reflex responses. This latter view suggests an alternative strategy that should focus more on interactions between language events in any dialogue, providing mechanisms for exposing the working assumptions of participants and aimed at achieving convergence of states of mind, without relying on some correct representation of a real world.[1]

The point in sketching out this divergence of views in the field of natural language is to show once again that knowledge about language, in common with knowledge more generally, is founded on belief. One starts with belief and builds up knowledge as though the belief is true, to discover what the belief enables us to do. In the relatively new field of natural language, relationships between beliefs, knowledge and achievements are still visible near the surface of the work by formal linguists. Shifts in beliefs can have profound effects on their formulations of overt knowledge. The object, then, is not to prove that any one form of knowledge is right and others wrong, but to explore the enabling power of different formalisms in terms of what they help people to do (not easy to evaluate). The kind of arguments that come to us from the work of linguists in AI, as outlined above, appears highly relevant to any formal approaches to design.

In looking to AI for guidance and help, we should resist the temptation to grab at new techniques in the desperate hope that they will somehow automatically solve our problems in CAD. The present enthusiasm for Expert Systems may prove to be a manifestation of such desperation. AI itself, at the present time, is also in danger of making the same error. Instead, we should keep in mind the

[1] For a fuller and better outline of this argument, see review by Winograd (1980) of his own evolutionary development in the field of natural language, and his reference to the work of Maturana.

fundamental issues that motivate developments in AI and recognise their equivalence with issues in CAD. Unavoidably, we have to recognise philosophical issues and discover how they condition our development and exploitation of new techniques.

Revising our assumptions

Returning to our earlier discussion of overt knowledge, formalisms, models and computer applications, can we conceive of new definitions that offer a better prospect of compatibility between computers and designers? An assumption for overt knowledge is that all overt expressions of knowledge, in the various modes of words, numbers and drawings, should ultimately be interpretable into some coherent and complete logical representation of all knowledge. In effect, the modes of expression should themselves then be understood as formalisms providing modelling environments, for purposes of communicating with this single all-embracing logic system. This assumption gives logic some superior existence outside of people, and lets it condition what can happen within people. Described in this extreme form, this assumption is uncomfortable - people make logic, so why should we let some people's creation dictate what is permissible for other people?

Can we conceive of another plausible but contrary assumption? Our earlier differentiation between overt and intuitive knowledge does imply a different assumption. This says that overt knowledge can only ever partially embrace all knowledge of people, and that logic systems have to be viewed as a multiplicity of discrete systems. This discrete nature of logic is permanent - we should expect logic systems to be redefined, over time, and we should expect links between systems to be illogical or intuitive. Discrete logic systems are defined and redefined by people, and interrelationships between systems are dependent upon mediation by people.[1]

This assumption of discrete logic is already evident, as when historians and sociologists refer to rolling rationality, meaning that rational perceptions at different times can be inconsistent without invalidating the separate rationalities.[2] More immediately, this assumption is evident in our established political institutions, in the role of our legislature and the manner in which its members are elected. People employ their intuitive judgment when legislating for connections between different sectional interests, where some of these interests may be rationally determined, and other people freely elect these legislators (attempts to let overt deterministic methods decide the formulation of legislation, present a contradiction and might be fundamentally unsound). The effect of this assumption is to place responsibility on people, expecting people to employ their holistic sense of being amongst other people, in the context of this world, to exercise their powers of temporal judgment. Here we have a characterisation of logic that appears to be compatible with the role of designers.

What are the implications of this assumption of discrete logic on the use of computers in AI and CAD? How can we expect representations of partial knowledge in computers to be useful to people? Can we expect boundaries to be drawn around parts of knowledge so that people can recognise the boundaries and decide on interrelationships between parts? Can such boundaries be changed during the course of using and extending knowledge? Here we have questions that provide scope for many years of research, and everybody should be interested in the results.

[1] To those who believe in a sharp distinction between logic and knowledge, my use of these terms may appear confusing; logic here refers to deeper levels of abstraction but not some bottom or absolute level.

[2] As expounded, for example, by Antony Giddens (in Ignatieff 1986).

Trying again

At Edinburgh we are exploring an approach to CAD which we believe has some relevance to these questions.[1] We now accept that knowledge can be revealed in different ways and that different logical constructs do not have to be consistent with each other - there is no outside arbiter of the correctness of logic. Instead, logic appears in ways that reflect people's various perceptions of their worlds, and connections between various perceptions are dependent on interactions between people outside logic. Our goal, therefore, is to devise a formal environment within computers that can *reflect* people's individual perceptions (or shared collective perceptions, allowing scope for bounded bodies of overt knowledge), and can serve as a medium for dialogue between people.

We accept that no part of the design of a system should be dependent upon any anticipation of knowledge that refers to anything that might be perceived in a user-world. The system should provide a general modelling environment in which any user can develop models of his or her own individual perception of the user's world. Users should then be able to use the system to communicate with each other, by referring to their separate knowledge represented in the same system. We accept the probability of conflicting knowledge and we do not regard the system as being responsible for resolving such conflicts - such resolution should be the result of dialogue between people, expressed in words and drawings, in the medium provided by the system.

This work is materialising in the form of our MOLE logic modelling system, developed in Prolog (outlined in figure 7 and presented by Krishnamurti in another paper at this conference, and in Krishnamurti 1985). We intend to give users full control of formal descriptions of things, where things can embrace any kind of object, event or operation, and descriptions may be subjected to any arbitrary modification. MOLE has to maintain logical coherence of descriptions, within the bounds of what it has been told by users, and it does so without relying on stored structures of whole representations or on any prescription of objects. To achieve these aims, we are forced back to a fundamental reconsideration of representational formalisms. The resulting system strategy has affinities with frame systems (with no prior organisation of slots) and semantic nets (with no significance given to names attached to arcs), using a "kinds/slots/fillers" convention that employs relative naming and inheritance.

As part of this work, we are devising a generalised syntax for line drawings. The syntax employs lines as primitives and line relationships, to describe segments and shapes - coordinate point values are not used. Generalised transformations of line intersections, segments and shapes are being defined for purposes of manipulating compositions of shapes in drawings. The syntax has a logical description within MOLE, so that instances of drawings can be automatically represented as MOLE descriptions. Drawings then form partial descriptions of other things. The intention is that a user should be able to use drawing as a means for communicating with MOLE, and that MOLE should be able to draw back to the user.

We make no claims for the practical utility of this system. It does not, as yet, have an acceptable user-interface and it is intolerably slow. Our purpose is not to develop a marketable CAD application, but to explore the questions that have been highlighted in this paper. We are confident that it will be feasible to devise new ways in which people can tell computers what they want them to know and do, describing things and the operations that are to be performed on things. We believe that such

[1] We, in this part of the paper, refer specifically to those of us at EdCAAD who are engaged in the work that is described.

descriptions can be interpreted by computers into models that reflect what individual users know. If we succeed, then we will have demonstrated the feasibility of working without predefined domain knowledge, and of using the mental mechanics provided by a formalism to support people's integration of overt and intuitive knowledge. These ambitions are extravagant but they are vitally important to people's acceptance of computers.

ACKNOWLEDGMENTS

The work on logic modelling under development at EdCAAD, in Edinburgh, is being supported by the UK Science and Engineering Research Council and the collaborative European Esprit Programme.

REFERENCES

Bijl, A., Stone, D., Rosenthal, D.; "Integrated CAAD Systems", EdCAAD Report to DoE (DGR 470/12), 1979.

Bijl, A.; "Computer Aided Housing and Site Layout Design", proc. PARC79, Berlin, West Germany, 1979.

Bijl, A.; "The Selfish Machine: computer discipline and design practice", book in preparation, 1986.

Capra, F.; "The Tao of Physics", Fontana/Flamingo, 1983.

Ignatieff, M. (presenter); "Modernity and its Discontents", TV Channel 4 and The Listener, March/April 1986.

Johnson-Laird, P.N.; "Mental Models - Towards a Cognitive Science of Language, Inference and Consciousness", Cambridge University Press, 1983.

Krishnamurti, R.; "Representing Design Knowledge", submitted to Environment and Planning B: Planning and Design (1986).

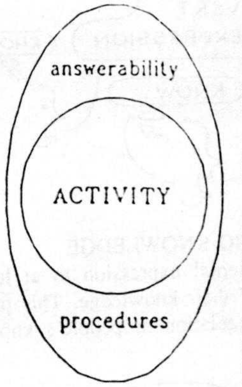
Schumacher, E.F.; "Small is Beautiful; A Study of Economics as if People Mattered",ABACUS, Sphere Books, 1974.

Winograd, T.; "What Does It Mean to Understand Language", Cognitive Science 4, pp.209-241, 1980.

Author: Aart Bijl

Address: EdCAAD, University of Edinburgh
20 Chambers Street
Edinburgh EH1 1JZ
Scotland

Phone: (031) 667 1011 x 4598 Telex: 727 442 UNIVED G



KNOWLEDGE BASED DISCIPLINE

ACTIVITY: explicit knowledge forms the primery means for progressing from problems to solutions

ANSWERABILITY: to peers who share a common knowledge base

DISCIPLINE: formal and detached, a shared basis for assessing new developments

PROCEDURES: methodical and stable, overt exclusion of intuitive judgement

EXPERIENCE BASED PRACTICE

ACTIVITY: relies on explicit knowledge plus intuitive judgement of practitioners

ANSWERABILITY: to public who do not share a common knowledge base

DISCIPLINE: informal and inconsistent, practitioners must be responsive to a volatile world

PROCEDURES: idiosyncratic, ill-defined integration of knowledge and experience

Figure 1: ANSWERABILITY AND PROCEDURES
In all situations in which people are answerable to other dissimilar people, where participants do not share a common knowledge base, intuitive knowledge within individuals is decisive in determining actions - intuitive knowledge here refers to that knowledge which is acquired through individual experience and which cannot be externalised so that overt expressions alone provide justification of the knowledge.

Lack of shared overt knowledge among designers and other people explains the lack of authority attached to design procedures and dependence on idiosyncratic design practices.

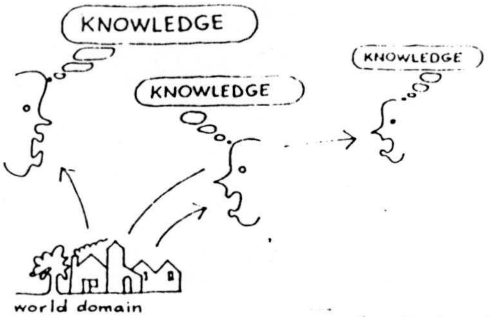


Figure 2a: INTUITIVE KNOWLEDGE
People's perception of their world may be thought of as intelligent, but if they had no means of exhibiting their knowledge we would not know. Knowledge wholly contained within individuals would leave us with the kind of intelligence that we might associate with dogs and cats.

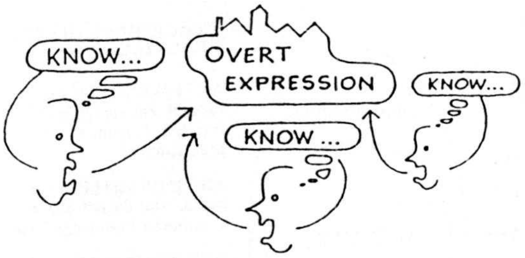


Figure 2b: EXTERNALISING KNOWLEDGE
People's ability to give external expression to at least part of their perception of their world, makes visible part of their knowledge. This physical manifestation of their perception is essential to our appreciation of people's knowledge and our ability to share and develop that knowledge.

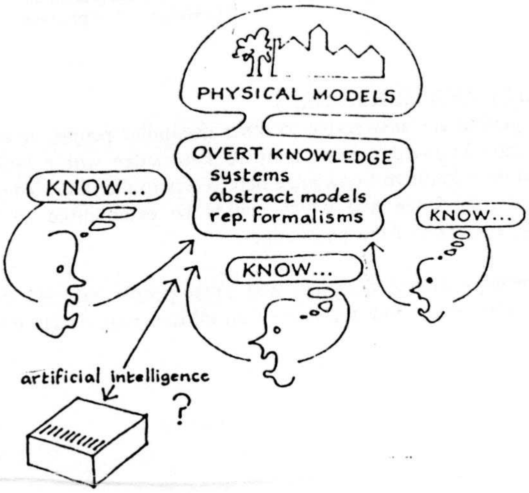


Figure 2c: OVERT KNOWLEDGE
Overt expressions employ formalisms, models and systems, as in the case of written language and line-drawings, that are used to describe further systems, like the construction of buildings - these overt expressions exist as physical models (e.g. printed words and drawn lines) that are interpreted symbolically or analogically into models that describe aspects of a user-world - any model's representation of some world phenomena is incomplete and may be partially informal, more so in the arts, less so in the sciences.

ARTIFICIAL INTELLIGENCE is aimed at increasing the power of people to externalise their knowledge; operates on overt expressions of people's knowledge and requires complete abstract formalisms that are not dependent on particular things any person might know or do.

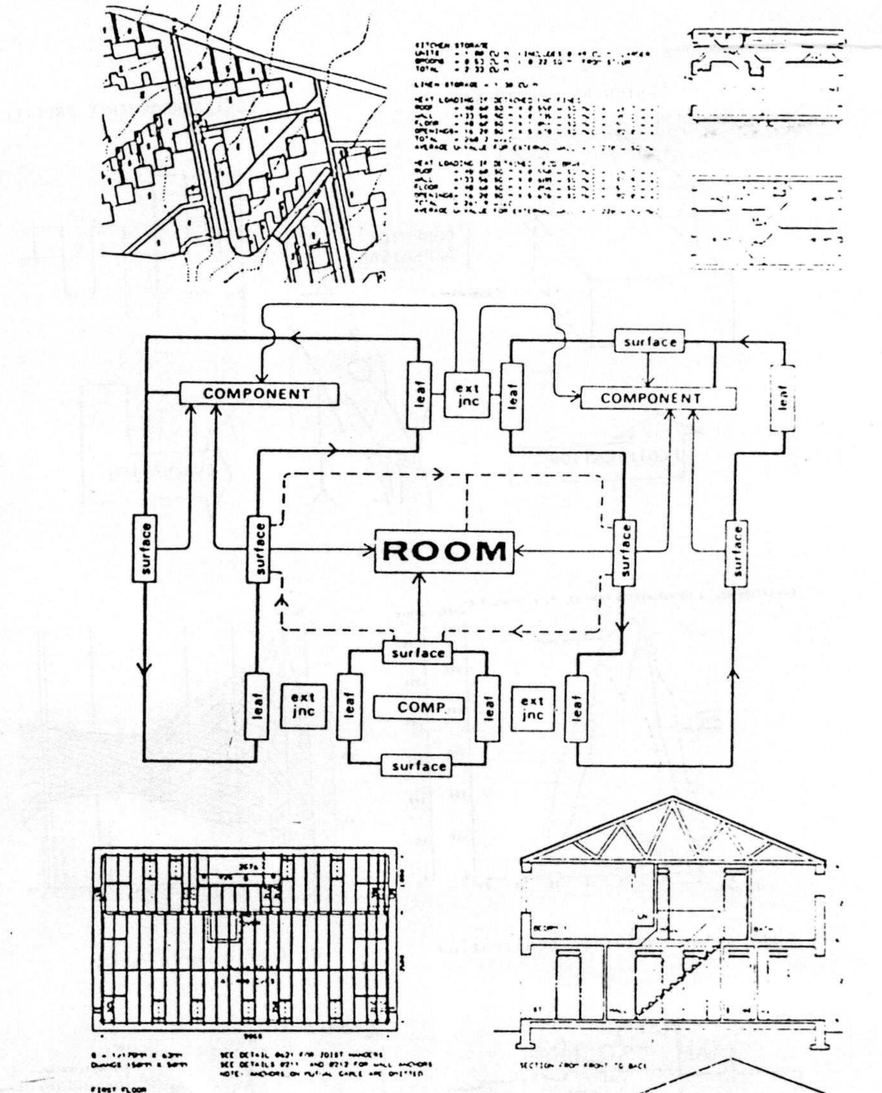
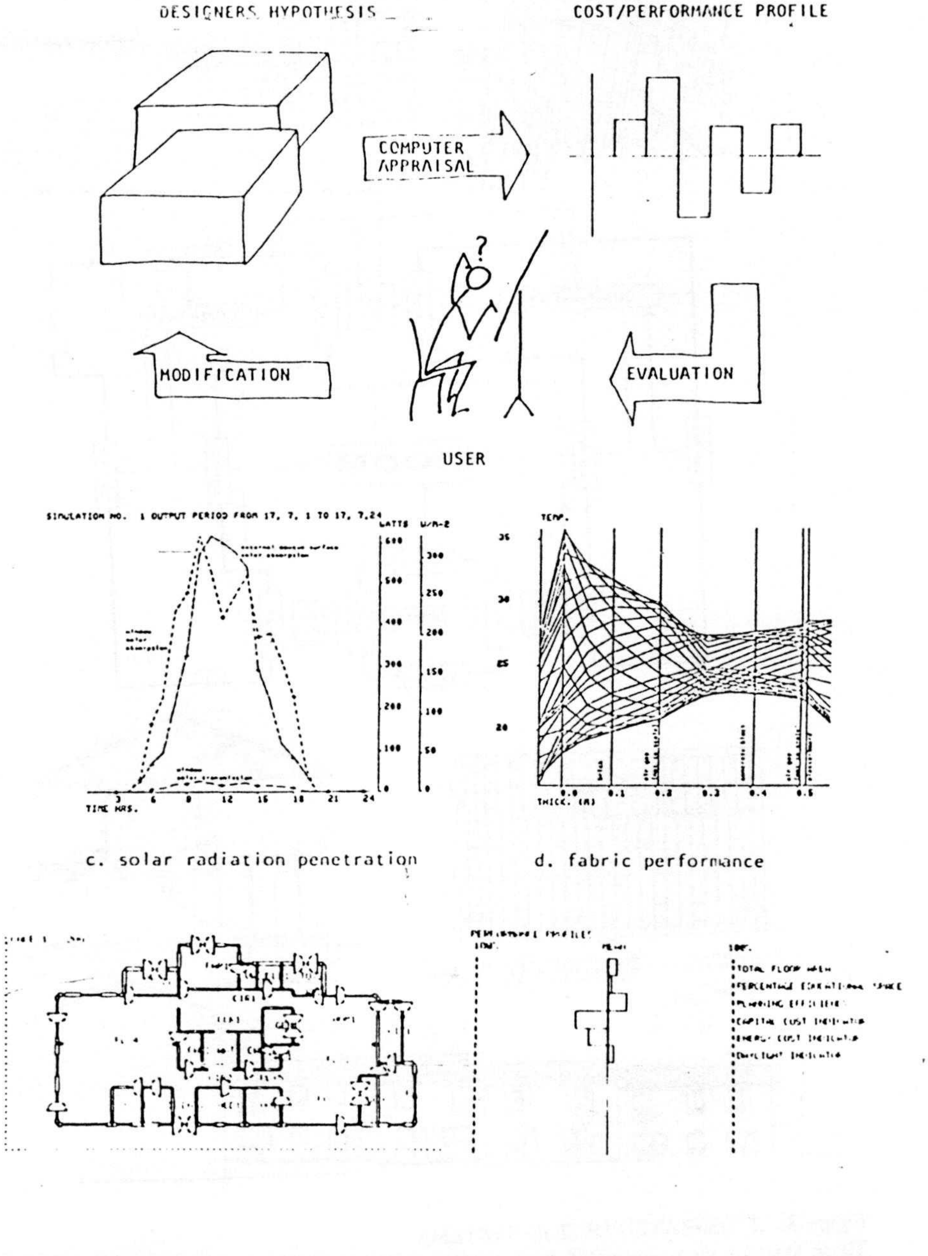


Figure 3: INTEGRATED DESIGN SYSTEMS
These systems were motivated by general considerations of data organisation, to provide single models of all that a user knows about each design, relying on the system to identify data relevant to each task across a broad range of tasks; this example is the SSHA housing design system, used for design appraisal and generation of production information, for more than ten years - eventually, the problem of maintaining equivalence between the user's work-practices and the representation of those practices within the system became too great - system by EdCAAD, Edinburgh.



c. solar radiation penetration

d. fabric performance

Figure 4: TASK SPECIFIC, FUNCTION ORIENTATED SYSTEMS
These systems are defined by an anticipation and formulation of a task (or a related set of tasks), with general consideration of input, data organisation and output subservient to the task - presenting a model of design as a sequential iterative process in which designers supply input in the form required by the task and assimilate output in terms of their own holistic perception of design goals - illustrations from ABACUS, Strathclyde.

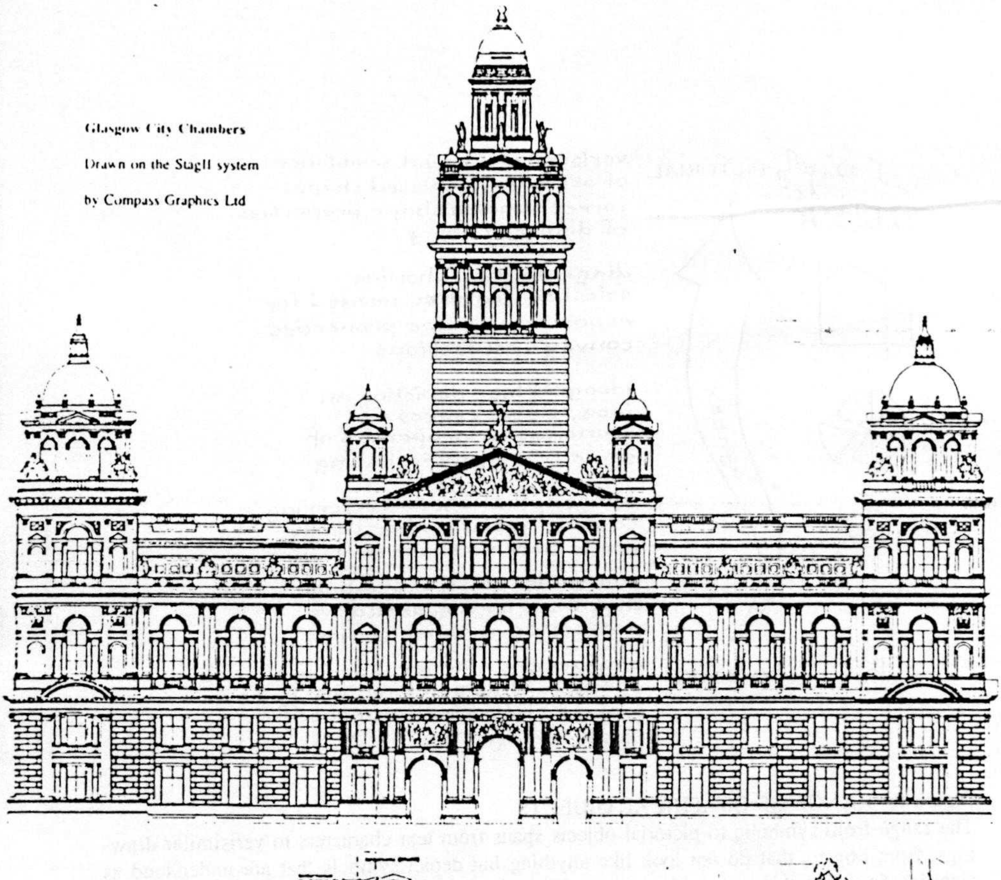


Figure 5: DRAWING SYSTEMS
An example of a "dumb" drawing system, STAG, which offers a congenial environment for producing any kind of line drawing, rather like word-processors with respect to text - the goal is to focus on drawings as objects independently of other things that they might depict, avoiding conflict between a machine's "intelligence" and other knowledge that users may variously associate with drawings - this system exploits AT&T's Unix, C and raster-ops, and is modestly priced, by DeCAL.

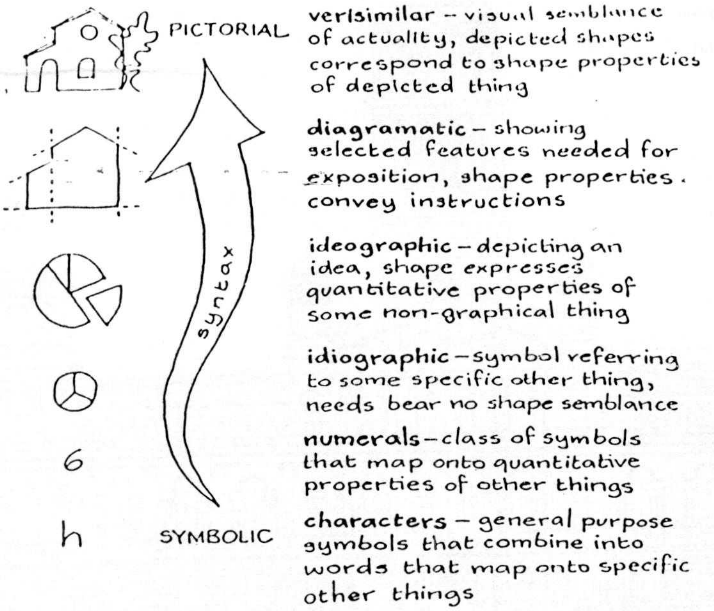

PICTORIAL
verisimilar – visual semblance of actuality, depicted shapes correspond to shape properties of depicted thing
diagramatic – showing selected features needed for exposition, shape properties. convey instructions
ideographic – depicting an idea, shape expresses quantitative properties of some non-graphical thing
idiographic – symbol referring to some specific other thing, needs bear no shape semblance
numerals – class of symbols that map onto quantitative properties of other things
characters – general purpose symbols that combine into words that map onto specific other things
syntax
6
h
SYMBOLIC

Figure 6: RANGE OF GRAPHICAL OBJECTS
The range from symbolic to pictorial objects spans from text characters to verisimilar drawings, from objects that do not look like anything but depict symbols that are understood as standing for other things, to objects that exhibit shape properties of other things, serving as analogues of shapes of the other things - if knowledge of other things is to be used by a drawing system to produce pictorial objects, verisimilar drawings, then the system must be capable of dealing with knowledge so as to reflect what each user knows.

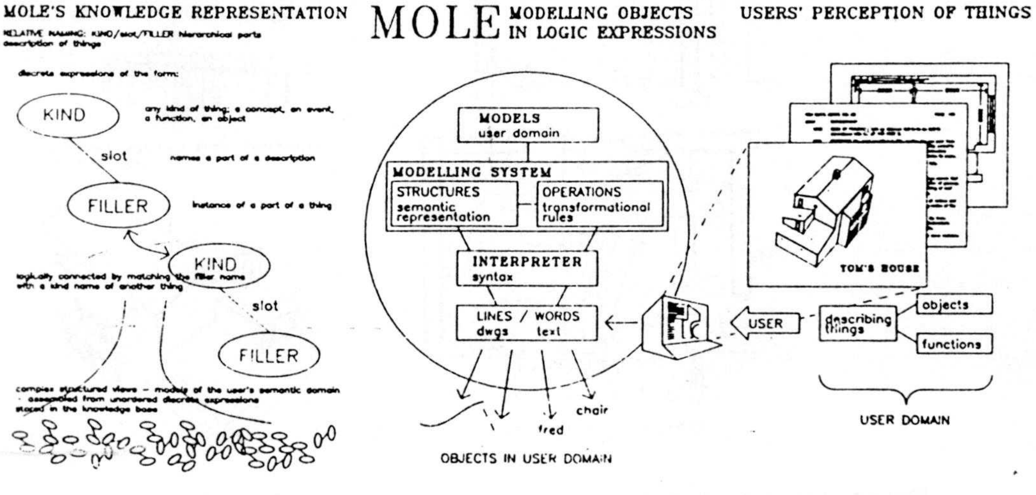

MOLE'S KNOWLEDGE REPRESENTATION
RELATIVE NAMING: KIND/slot/FILLER hierarchical parts description of things
discrete expressions of the form:
KIND
any kind of thing; a concept, an event, a function, an object
slot
names a part of a description
FILLER
instance of a part of a thing
KIND
logically connected by matching the filler name with a kind name of another thing
slot
FILLER
complex structured views – models of the user's semantic domain assembled from unordered discrete expressions stored in the knowledge base
MOLE
MODELLING OBJECTS IN LOGIC EXPRESSIONS
MODELS
user domain
MODELLING SYSTEM
STRUCTURES
semantic representation
OPERATIONS
transformational rules
INTERPRETER
syntax
LINES / WORDS
dwgs text
chair
fred
OBJECTS IN USER DOMAIN
USERS' PERCEPTION OF THINGS
TOM'S HOUSE
USER
describing things
objects
functions
USER DOMAIN

Figure 7: THE MOLE LOGIC MODELLING SYSTEM
Knowledge possessed by users, expressed in words and drawings (right), interpreted by MOLE into abstract models corresponding to users' descriptions of their own worlds (centre), represented in the Kinds/slots/Fillers formalism, employing relative naming and inheritance to reassemble structured views on demand from users (left) - system is intended to reflect what it has been told, to reflect what the user knows, with no prescription of what knowledge might occur in a user-world.