Sādhanā, Vol. 19, Part 1, February 1994, pp. 5–21. © Printed in India.

Understanding design: Artificial intelligence as an explanatory paradigm

SUBRATA DASGUPTA

Department of Computation, University of Manchester Institute of Science and Technology, Manchester, UK
Present address: The Center for Advanced Computer Studies, University of South Western Louisiana, Lafayette, LA 70504–4330, USA

E-mail: dasgupta@cacs.usl.edu

Abstract. A substantial part of the intellectual content of what H A Simon called the 'sciences of the artificial' is contained in the activity we call *design*. A central aim of *design theory* is to construct testable, explanatory models of the design process that will serve to enhance our understanding of how artifacts are, or can be, designed. In this paper, we discuss how some of the basic concepts underlying the discipline of *artificial intelligence* (AI) can serve to provide an *explanatory paradigm* for understanding design. We present an AI-based model of the design process and describe some of the implications of this model for our understanding of design – including that aspect of it we call 'invention'.

Keywords. Design; artificial intelligence; explanatory model of design; creativity; invention.

1. Design theory

Anyone who devises a course of action to change an existing state of affairs to a preferred one is involved in the act of design. As such, design is of central concern not only in traditional engineering – dealing with *material* artifacts such as structures, machines, circuits and production plants – but also in the generation of *symbolic* devices such as plans, organisations and computer programs. Indeed, it is this larger sense of the word 'engineering' that Herbert Simon had in mind when, in 1969, he coined the term 'sciences of the artificial' to designate all such disciplines that are concerned with the conception and production of useful artifacts (Simon 1981).

There has been a long-held notion that the sciences of the artificial (or, more conveniently, the *artificial sciences*) were simply *applications of the natural sciences*: that civil engineering, for example, is the application of mechanics, and mechanical engineering of mechanics and thermodynamics; or that electrical engineering is the application of electro-physics, and metallurgy of chemistry and solid state physics. The fact that the engineer and the researcher in the artificial sciences are concerned with the effecting of artifacts intended to serve some *purpose* and that purposiveness is totally at odds with the natural sciences hardly seemed relevant (according to

conventional wisdom) as far as the intellectual foundations of the artificial sciences were concerned.

Since the 1960s, several works have appeared which, in one way or another, have all been dedicated to the proposition that the world of the artificial contains its own logic which is related to but is quite distinct from the logic of the natural world (Jones & Thornley 1963; Pye 1964; Jones 1980; Cross 1984; Aguero & Dasgupta 1987; Brown & Chandrasekaran 1989; Coyne *et al* 1990; Dasgupta 1991). It has also come to be explicitly recognised that while there are many distinct artificial sciences – civil, mechanical, chemical and electrical engineering, metallurgy, aerospace technology, agriculture, computer science, organisation theory, economic and social planning, architecture etc. – there is one kind of intellectually nontrivial activity that is shared by all, viz., *design*. Furthermore, if we examine what the various kinds of designers have to say about their respective domains (be they bridges, machines, cities, software, administrative organisations or integrated circuit chips) we discover that the same *kinds* of things are being described regardless of the domain. The vocabulary may differ but the concepts are the same.

From such observations it has come to be realised that there is a significant component to all these domain-specific design processes that is essentially independent of what is being designed. That is, irrespective of whether we are designing chips, programming languages, computers, robots, airline reservation systems, bridges, cities or chemical plants, the processes of design have a strong *domain-independent* component.

The implication of this is considerable. For, it means that we can conceive of a discipline *the subject matter of which is the design process itself*. In recent years, this discipline has come to be known as *design theory* (Coyne *et al* 1990; Dasgupta 1991), and its scope or aim is essentially twofold:

(a) to construct *explanatory models* of the design process – models that will serve to clarify, explain and enhance our understanding of the acts or processes whereby artifacts are, or can be, designed; and, consequently,
(b) to establish foundations for the implementation of rational methods, tools and systems that may aid the activity of practical design.

These two objectives are mutually reinforcing in that each furthers the cause of the other: a better understanding of design as a cognitive process is likely to provide a sounder basis for inventing design methods and tools; conversely, the development and implementation of such methods and tools provide important data for constructing better explanatory models as well as for testing or evaluating such models. The two objectives also complement each other in that the first is concerned with the *description* of design viewed as an empiricial cognitive process whereas the second relates to *prescribing* ways of doing design.

2. Artificial intelligence as an explanatory paradigm

In its ordinary sense, a *paradigm* is an example, a pattern or a model, as when we refer to the stored program computer as conceived in the 1940s as a paradigm for computer architecture, or as when Petroski recently referred to certain kinds of errors leading to engineering design failures, as 'paradigms for human error in design' (Petroski 1991).

This dictionary notion of paradigm was greatly enlarged in the 1960s by Kuhn

who in his (now classic) studies on the nature of scientific revolutions used this word in a very special way to advance an account of the origin and development of scientific disciplines (Kuhn 1962, 1970, 1977).

In essence, a Kuhnian paradigm is a network of generalised theories, metaphysical assumptions, metaphorical and heuristic models, methodological commitments, values and exemplars that are shared by, or are common to, a given scientific community. A paradigm provides the framework within which members of that community recognise and solve problems.

Kuhn's theory of paradigms and the role he ascribed to them in the development of scientific thought has been the subject of considerable discussion and criticism (Shapere 1964; Lakatos & Musgrave 1970; Laudan 1977, 1984; Suppe 1977; Lakatos 1978; Cohen 1985; Bohm & Peat 1987; Thagard 1990). Our concern here, however, is not with the nature of these arguments. In fact, we accept the essential substance of Kuhn's general thesis and wish to put it to a particular use. Our aim in this paper is to examine how the concepts underlying the discipline of *artificial intelligence* (AI) can serve as a Kuhnian paradigm for understanding the nature of the design process.

It is important to note that practically all research in the application of AI to the topic of design has been concerned with the prescriptive aspect of design theory – that is for automatising the design process (Mostow 1985; Brown & Chandrasekaran 1989; Chandrasekaran 1990; Coyne *et al* 1990; Gero 1991). This paper focuses on the role that AI plays or may play in the descriptive arena. More specifically, we shall be concerned here with the issue of how the concepts of AI can assist in the construction of *explanatory models* of the design act – including in the realm of the most creative level of design which we call invention. Thus, viewing AI as a Kuhnian paradigm for the exploration and understanding of design as a cognitive process makes it, in the context of this particular paper, not so much a technology as a theoretical handmaiden for cognitive science, much as mathematics serves as a servant for physics or for some aspects of engineering.

3. The metaphorical role of the artificial intelligence paradigm

The question that may obviously be raised at this stage is: how may AI be appropriate for this purpose? And why should AI be preferred as a basis for explanation (of design processes or any other phenomena demanding explanation) to some other paradigm? To answer these, we begin with the fact that explanations in many arenas of science – including cognitive science – frequently draw upon the use of *metaphors*; and that metaphors of a particular kind serve as *models*. Computational schemes of the type provided by AI are especially useful for the purpose of constructing such models. Let us elaborate on these points.

That metaphors are used as a means of understanding even in every day discourse is a commonplace idea. Indeed, Jaynes has made the point that understanding is *primarily* a matter of constructing metaphors whereby what we wish to understand (the *metaphrand*) is related to (or mapped onto) what we do understand or are familiar with (the *metaphier*) (Jaynes 1976).

What is less understood is that metaphors may play significant roles in scientific explanations. To take two celebrated examples, both Darwin and Lavoisier drew upon the use of metaphors to arrive at their respective conclusions about evolution and the chemistry of respiration (Gruber 1981; Holmes 1985).

As we have discussed elsewhere (Dasgupta 1993, 1994), the kind of metaphors

evoked by Lavoisier and Darwin are especially useful in that they serve to establish *analogical relationships* between metaphrand and metaphier. As a result, one can draw inferences or extract facts from the metaphier's domain and transfer them to that of the metaphrand where they can serve as sources of explanation. For example, in Lavoisier's case, the known chemistry of the burning of a candle (the metaphier) was exploited to suggest the unknown chemistry of respiration (the metaphrand) (Holmes 1985). In the case of Darwin, one of the metaphiers was artificial selection. This allowed him to draw upon facts pertaining to the hybridisation of plants and animals through breeding as a suggestive mechanism for how variations in species occur in nature (Gruber 1981).

Metaphors of these types, then, have instrumental or *heuristic* value because they can be used to *explain* as well as to evoke images. For this reason, it is more appropriate to call them *metaphorical models* (Dasgupta 1993, 1994). It is in this context that the language and concepts of AI are useful. For, if we are willing to accept that the act of design involves the *processing of symbolic structures* (see §4 and 5 below) then computation (in its most general sense) seems to provide the most appropriate tool for explaining the nature of such processes – since computation is, fundamentally, *the* discipline concerned with symbolic transformations and the processing of symbolic structures. Computation – and the particular form of computation that is the hallmark of AI – thus becomes a metaphorical model for explaining design.

It is important to emphasize, once more, the *heuristic* nature of such models. To take another instance from the history of science, the development of the kinetic theory of gases relied on 'seeing' gas molecules as hard, elastic and spherical – i.e., as microscopic billiard balls (Holton 1952). It is not *really* thought that gas molecules are billiard ball-like. Models are constructed and (tentatively) accepted *as if* they are true because it is useful or fruitful to do so. Viewing gas molecules *as if* they are hard, elastic spherical entities paved the way for classical mechanics to be applied in order to explain the known behaviour of gases.

Correspondingly, it does not have to be that computational models must capture the reality of the cognitive act of design in a 'truthful' way. Rather, we desire that such models should be able to *represent* design processes in the sense that:

(a) the known or documented phenomena surrounding design acts can be explained by the model in a consistent way;
(b) using the operational power of the model one can provide plausible explanations of cognitive acts of design for which there are no documented accounts;
(c) the model provides a better explanatory framework than any other known paradigm.

In other words, if a computational account of design 'works', then, in the absence of a rival paradigm that 'works better', we should be willing to adopt, at least *tentatively*, the computation-based paradigm as an instrumental theory of the design process.

4. A knowledge level model of the AI paradigm

At this time of writing there is, of course, something of a struggle between two schools of thought concerning the 'true' nature of the AI paradigm (Papert 1988). One is the *symbol processing model* which has its origins in the work begun in the 1960s by Simon, Newell and their collaborators (Newell *et al* 1960; Newell & Simon 1972, 1976;

Newell 1982) and the other is the *connectionist* model which, though having roots in the work of Pitts and McCulloch in the 1940s, assumed its modern form relatively recently (Papert 1988). Fortunately, this debate need not detain us here for *as far as design is concerned*, the dominant model is the symbolic version. Thus, in this paper at least, the AI paradigm is based on the symbol processing model.

To be more precise, we shall present a characterisation of the AI paradigm at what has come to be called the *knowledge level* of cognition. This term was actually coined, and a systematic treatment of its features first presented, by Newell (1982) although the knowledge level as an appropriate level at which cognitive processes could be described has long been tacitly recognised in the AI literature.

A system at the knowledge level will be referred to as an *agent*. The main entities with which an agent is concerned are *goals*, *actions* and *knowledge*. As Newell (1982) put it:

> To treat a system at the knowledge level is to treat it as having some knowledge and some goals and believing it will do whatever is within its power to attain its goals insofar as its knowledge indicates.

In Newell's formulation, the connection between knowledge, goals and the choice of which action to take (in order to achieve the goals) is established by a behavioural principle which he termed

> *The principle of rationality* (PR): If an agent has knowledge that one of its actions will lead to one of its goals then the agent will select that action.

A problem with PR is that it tells us nothing about what the agent might do if it does not possess the requisite knowledge. Nor is it helpful in the situation where we observe an agent making a choice in response to a goal. Are we, for instance, to infer *abductively* that the agent possesses the requisite knowledge that that particular action will lead to the desired goal?[1]

Such a conclusion may be wholly unwarranted. An agent may possess *incomplete* or *partial* knowledge concerning the appropriate action to take in response to a goal. Alternatively, the *computational cost* of determining which action to select from a set of alternatives may be so *high* as to render such determination impractical. In other words, in addition to the rationality principle PR, an agent is governed by Simon's (1976, 1982)

> *Principle of bounded rationality* (*PBR*): Given a goal, an agent may not possess perfect or complete knowledge of, or be able to economically compute or access, the correct action (or sequence of actions) that will lead to the attainment of the goal.

The consequence of PBR for the theory of the knowledge level agent is that, given a goal, there is no guarantee that in selecting an action (or a sequence of actions) the goal will, in fact, be attained.

Ideally then, an agent's behaviour at the knowledge level is governed by PR. In reality, it is constrained by PBR. This means that any action(s) the agent chooses in

---

[1] Abduction is the rule of inference,

(IF A THEN B, B/A).

For a comprehensive discussion of abduction, see Thagard (1988)

order to attain a goal represents, in general *a hypothesis* (on the part of the agent) that the action(s) will lead to the goal.

An individual action *does* something. It has an *input* to it and it produces an *output*. In general, both input and the resulting output may be in the form of matter, energy or symbols. However, in the specific context of design, our concern is only with *symbol processing actions* in which the input and output are both symbol structures.

Symbols or structures composed out of symbols may, in general, be either *formal* (in that they stand for or represent mathematical sentences) or *physical* (in that they stand for or represent entities in some external universe – and so their 'meaning' are interpreted with reference to that universe). We shall use the term *general symbol structure* to refer to either formal or physical symbol structures.

We have noted above that the actions of interest here are symbol processing actions. In fact, actions may themselves be represented by symbol structures. More generally, *all goals, knowledge and actions pertaining to an agent are representable at the knowledge level by general symbol structures.*

Every action consumes some amount of time. While the actual duration of an action is unimportant here, it is to be recognised that an action has a *beginning* point and an *end* point in time; this means that an action may begin or end earlier or later than some other actions.

Actions may take place in sequence or in parallel. A *sequence* of actions $a_1, a_2, \ldots, a_n$, where a_i ends before a_{i+1} begins $(1 \leqslant i \leqslant n-1)$ will, as a whole, have an input I which is the input to a_1 and an output O which is the output of a_n such that the output of a_i is the input to a_{i+1}. Actions may also be conducted in *parallel* by an individual agent or a team of agents. It is assumed that parallel actions satisfy

> *The principle of determinacy* (PD): If a set of actions $a_1, \ldots, a_n$ are conducted in parallel and if I is the input to this set then the output O will be identical to the output O' which would be produced if the same actions $a_1, \ldots, a_n$ were to be conducted in some *arbitrary* sequential order with the same input I.

In other words, according to PD, the input/output behaviour of a set of parallel actions is indistinguishable from the input/output behaviour of the same set of actions performed in *any* sequential order.

We shall refer to any sequential or parallel set of actions as a *structured set* of actions. Such a set will have one or more actions that are its *earliest* if no action outside this subset begins earlier than those within the subset.

Upto this point, actions have been linked with goals – that is, actions are assumed to be invoked in response to goals subject to the behavioural principles PR and PBR. However, it may also be possible for an action to be initiated *without* the stimulus of a goal. It may be initiated by virtue of an element or *token* in the agent's knowledge body – in which case, such an action is not governed by PR or PBR. We shall, therefore, distinguish between *rational* actions (actions that are invoked in response to goals) and *nonrational actions* (those that are invoked in response to tokens in the agent's knowledge body).

In summary, actions and the conditions of their invocation can be characterised as follows.

[1] The input to an action is one or more symbol structures representing goals or knowledge tokens. If at least one of the inputs is a goal, the action is termed rational. Otherwise, it is nonrational.

[2] The output of an action is one or more symbol structures representing either a knowledge token or a goal.
[3] Every action entails the retrieval and application of tokens contained in the agent's knowledge body.
[4] The choice of an action in response to a goal is governed by the principle of rationality (PR). That is, if an agent has knowledge (where such knowledge may be as weak as a belief) that one of its actions will lead to the goal being achieved, it will select that action.
[5] Because of the bounded rationality principle (PBR), however, an action so chosen may not be the correct action or may not be economically computable by the agent.
[6] Every action consumes time.
[7] Actions may be performed sequentially or in parallel. A set of actions, some of which are sequential, others parallel, is said to be structured.
[8] In a structured set of actions, its parallel subsets obey the principle of determinacy (PD).

Finally, the AI paradigm as a whole can be concisely described in the following terms:

DEFINITION 1

A *knowledge level process* $P(KL)$ is a structured set of actions conducted by an agent (or, cooperatively, by a collection of agents) in response to a goal (or a conjunction of goals) G such that:

(a) The input to $P(KL)$ is a set of symbol structures at least one of which represents G.
(b) The output of $P(KL)$ is a set of symbol structures that represent goals or knowledge tokens where the latter includes, possibly, a *solution* to G – that is, tokens that represent a solution to, or achievement of, G.
(c) $P(KL)$ terminates when either (i) its output contains a solution to G or (ii) its output is such that no further action is (or can be) selected. *End Def*

Thus, the AI paradigm is defined here in the form of a symbol-transforming process. Such a process begins with a goal. The latter, subject to the principle of rationality, prompts an action (or a structured set of actions) involving the selection of tokens from the agent's knowledge body. The output produced by the action(s) may be a symbol structure which the agent believes is a (possibly partial) solution to the original goal.

However, because of bounds on the agent's rationality, the output may be a new (and more tractable) goal. The latter prompts one or more new actions to be performed and so the process continues. The process terminates when the original goal is achieved or when no further action can be performed by the agent.

5. Design as a knowledge-level process

One of the very real problems encountered by design theorists is the difficulty of defining the act of design in a form which, on the one hand, satisfies our intuitive idea of design and, on the other, permits useful and interesting inferences to be extracted from the definition. As we have discussed elsewhere (Dasgupta 1991), the many defnitions advanced by theorists in the past have proved rather unsatisfactory

in these two collective aspects. Thus, rather than beginning with a definition, we may be forced to rely on our intuitive notion of design and examine its many characteristics in an empirical fashion. This was the approach we took, for example, in a previous work (Dasgupta 1991).

However, we believe that the knowledge level model of the cognitive agent as just described does provide the basis for the definition we seek – and herein lies the first benefit of the knowledge level AI model as a Kuhnian paradigm for design theory. Thus, we have:

DEFINITION 2

A *design process* is a knowledge level process that satisfies the following properties:

(a) The input to the process designates (or specifies or represents) a set of properties to be met by some artifact in some given universe. These properties are referred to as the *set of requirements, R.*
(b) The output of the process designates (or represents) the artifact. This representation is referred to as *the design, D.*
(c) The goal of the agent in conducting the process is to produce a representation or design (*D*) such that if an artifact is implemented according to *D* then it will satisfy the properties constituting *R*. This goal is referred to as the *design goal* and may be stated tersely as *D satisfies R.*
(d) The agent has no knowledge of any design that satisfies *R. End Def.*

Let us consider, first, how this definition coheres with what we know empirically about the design process.

(i) According to the above, design, being a knowledge-level process, is a structured set of actions that can be conducted by an individual agent or a team of agents. Thus, the definition recognises that design may be performed by a single designer or by a design team.[2]
(ii) The actions performed do not lead to matter or energy to be transformed. They are *symbol processing* actions.[3]
(iii) Furthermore, both the input and the output symbol structures designate entities in some given universe: the (input) requirements designate properties demanded of some artifact; the (output) design represents the artifact itself. The symbol structures are, then, *physical* symbol structures. The definition, thus, excludes purely formal symbol processing activities such as mathematics from its scope. This is intuitively satisfactory: we do not normally think of constructing theorems or proofs of theorems as designing.
(iv) Because the output of a design process, as defined above, is a (physical) symbol structure, what it produced is a representation of the artifact, never the artifact itself. It is the representation that constitutes 'the design'. Thus, the definition allows us to distinguish between 'designing' and 'making' (Alexander 1964; Jones 1980; Dasgupta 1991). Obviously, the traditional craftsman of old also conceptua-

[2] For convenience, we shall talk simply in terms of *an agent* with the understanding that whatever is said applies equally to a team of agents.
[3] Of course, at some lower levels of abstraction (e.g. at the neuronal level) symbol processing actions will entail the transformation of matter and energy.

lised the form of the artifact he was creating. However, the essence of design is that it results in a *symbol structure* – that is the result is externalised and, consequently, communicable.

(v) According to the definition, a design process is initiated only when the agent is posed with a set of requirements such that the agent is unaware of any other design or artifact satisfying the requirements. For a given set of requirements, if there already exists an artifact that satisfies it then there would be no need to design. Thus, 'newness' or change (in even the most modest of terms) is a condition for a design process to be initiated, according to the definition. This conforms to the observation that one designs in order to initiate change (Simon 1981; Dasgupta 1991).

(vi) However, note that according to definition 1, 'newness' is always in the context of, or relative to, the agent's knowledge body. Empirically, this is entirely reasonable. For example, given a specification of requirements R, for an integrated circuit chip, the experienced engineer (i.e., one whose knowledge body contains many 'cases' – exemplars, in Kuhn's terms (Kuhn 1962, 1977) – of prior designs) may know of a chip or a chip design that satisfies *exactly* the requirements. In that case, there is no need to design the chip. The same problem given to a student or a neophyte engineer may lead to a design process being initiated simply because the latter has no knowledge of appropriate exemplars.

Most real design situations fall within these two extremes. For instance, the civil engineer is posed with requirements for a new bridge that include details of the required span, the local soil conditions, the topography and the expected loads. These specifications may be found to be similar but not identical to the characteristics of a particular design known to that engineer. In that case, a design process will be initiated which takes the known bridge design as the starting point. Thus, definition 2 excludes neither 'design from scratch' nor 'redesign'.

(vii) It is well known that many design problems belong to the class of what Simon (1973) termed *ill-structured* problems. That is, the requirements are incomplete or imprecise or ambiguous or the spaee of potential solutions is unbounded.[4] In the rarer situations, design problems may be *well-structured* – that is, the requirements are stated in such a manner that one can immediately devise tests to determine whether or not a given design satisfies those requirements (Dasgupta 1991).

Along a different axis, the requirements may be such that the collective (or 'public') knowledge body of the relevant design community has no tokens that may provide the starting basis for a solution. In that case, the agent has to literally *invent a new artifactual form*. This situation corresponds to the most creative form of design, viz., invention. At the other extreme, the requirements may be such that the agent's knowledge body (or that of the relevant design community) has a very precise *archetypal form* or *schema* for the artifact. In that case, the design act may entail *instantiation* of the schema by fixing or setting some parameters to specific values. Brown & Chandrasekaran (1989) refer to this as *routine* design.

It will be noted that definition 2 allows for a range of design problems that fall within a space determined by both these axes (figure 1).

[4] Dasgupta (1991) gives many examples of ill-structured design problems

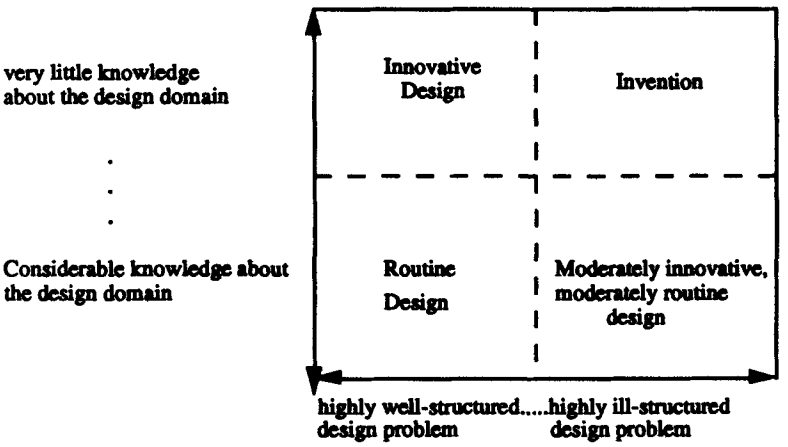

Figure 1. Space of design problems.

6. Implications of bounded rationality. I. Designs as satisficing solutions

Since the act of design is an instance of a knowledge level process, it is subject to the constraints of bounded rationality. This leads to several insights into the processes and nature of design. We consider some of these in this section and in the sections to follow.

One such insight is the distinction Simon made between 'optimal' and 'satisficing' designs. We have just seen that according to definition 2, design is a process entailing the construction of a representation of some artifact that meets the given requirements. It is understood that the solution sought is one that is the 'best' in some sense. In designing a computer system, for example, we seek to arrive at the best possible instruction set or memory management scheme or data path, as the case may be.

According to the principle of bounded rationality (PBR), however, even when all possible alternatives *are* known in advance, the cost of deriving an optimum or 'best possible' design may be prohibitively high. In spite of knowing that there exists an optimal solution to a design problem or even the actual procedure (that is, the set of knowledge level actions) that would yield the optimal, PBR tells us that the designer may not possess sufficient cognitive or computational resources to *actually determine* the optimum. Many of the optimisation problems encountered in design are what computer scientists call *intractable* in the sense that their solutions require processes of exponential time or (memory) space complexity (Dasgupta 1991).

So what does the designer *actually* do in the case of intractable problems? Or in the case of ill-structured design problems that are not amenable for formulation as optimisation problems?

One of Simon's major insights was that for most nontrivial design problems, levels of aspiration or *satisfactoriness* are established rather than criteria of optimality (Simon 1976, 1981). For instance, a bridge design is accepted if its estimated cost is 'below a certain amount'; a computer design project begins with the requirement that its peak performance must be 'twice that of its predecessor system' or have 'a better cost/performance ratio than that of its competitor'. If the design meets such criteria the problem is considered to have been solved. Simon named such solutions *satisficing*

solutions. Thus, in general, the design process attempts to satisfice rather than optimise.

7. Implications of bounded rationality: II. Two laws of design

There is yet another important consequence of PBR as far as design is concerned. It is that there is no guarantee that the design process conducted by an agent will, *in fact*, meet the design goal. A design D produced in response to the given requirements R embodies a *hypothesis* that D satisfies R. More formally, Dasgupta (1992) has recently proposed, and provided arguments in support of

> *the hypothesis law*: A design process that reaches termination does so through one or more cycles of hypothesis creation, testing and modification.

While this law was derived directly from the knowledge level definition of design (Dasgupta 1992) and, in particular, from PBR, some form or another of this law has been widely recognised in the literature of design theory though couched mostly in terms of the concept of evolution. For example, Chandrasekaran's (1990) concept of a class of design methods which he called *propose-critique-modify* is, clearly, along the lines of the hypothesis law although he does not quite claim that his model constitutes a universal characteristic of the design process. Dasgupta has previously described in detail, with many examples from the domain of computer systems, the general concept of *design as an evolutionary process* and the idea of a design as constituting a theory or a hypothesis (Dasgupta 1989b, 1991). Finally, an earlier more informally stated suggestion, that designs signify hypotheses, is due to Petroski (1985).

Another ramification of the knowledge level model of the design process (and of bounded rationality) is captured by a law presented by Dasgupta (1992) called

> *the impermanence law*: A design in any given state is never guaranteed to remain in that state.

Here, 'state' refers to the *state of belief* that may be held about the hypothesis that the design satisfies the requirements. Possible states are defined according to the following.

DEFINITION 3

A design D produced in order to achieve a goal G: 'D satisfies R' for a given set of requirements R is said to be

(i) VALIDATED when an agent produces a structured set of actions T (called a *test*) drawn from some knowledge body K that demonstrates that G has been achieved.
(ii) REFUTED when an agent produces a test T that demonstrates that G has *not* been achieved.
(iii) TENTATIVE when an agent can produce neither a test T_1 that demonstrates that G has been achieved nor a test T_2 that demonstrates that G has not been achieved.
End Def.

The structured set of actions – the tests T, T_1, T_2 in the above definition – may take many forms. It may involve invoking some items from the agent's knowledge body, e.g. some previously published analysis or data; the construction of a mathematical

proof; a simulation experiment; or experiments constructed on a prototype. The outcome of the tests performed constitutes the *evidence.*

8. The non-monotonicity of design

As in the case of the hypothesis law, one can provide arguments in support of the impermanence law (Dasgupta 1992). The significance of the latter is considerable for it asserts that any evidence we summon in support of a claim about a design (that is, that it does or does not satisfy the requirements) *is itself conjectural.* That is, the reasoning underlying any claims we make about a design, like all empirical reasoning, is *non-monotonic* in nature (Reiter 1987). No matter how sure we may be at time t_1 that the design is in the validated state (because, say, the evidence at time t_1 happens to be compelling), there is no guarantee that this state of affairs will remain so at some (possibly much) later time t_2 – when new contrary evidence may have come to light. For example, a new set of tests may falsify the earlier claim about the design being in the VALIDATED state; or we may realise that our earlier reasoning was faulty; or the assumptions upon which we had staked our claim may be discovered to be wrong. Anyone of these will result in the design being shifted to the REFUTED state.

The *practical* implications of the impermanence law – that is, of the non-monotonicity of designs – is also considerable when we consider the prescriptive side of design theory (refer § 1) in which the concern is to propose effective design methods and tools. For, if the impermanence law is indeed universal then any design method we

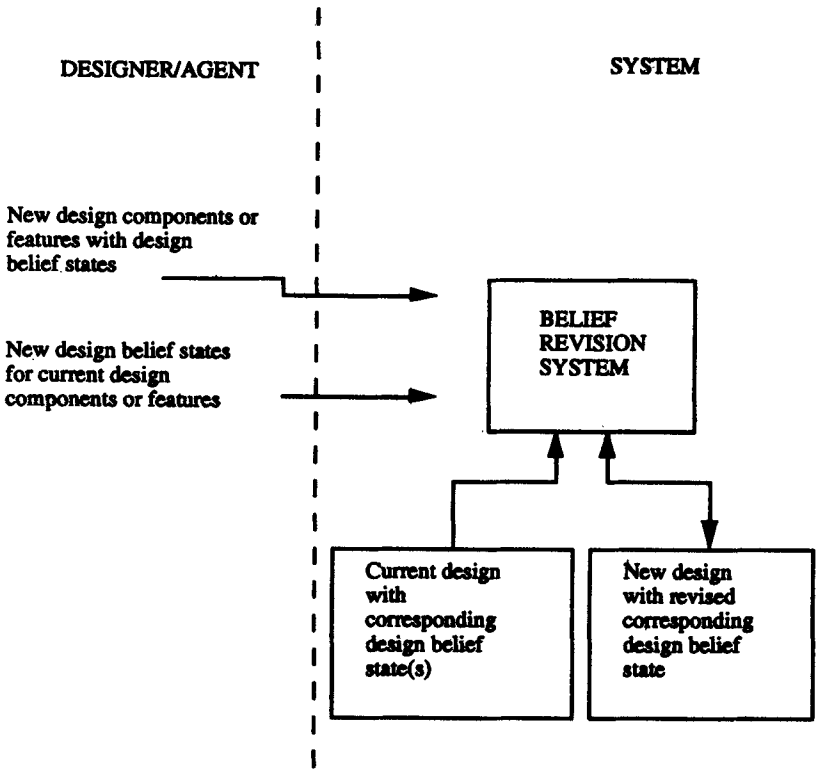

Figure 2. Architecture of a computer-aided belief revision system.

may propose (whether to be performed by 'cognitive' or 'computational' agents) must take into account the fact that the state of belief about the overall design *must be constantly revised* as and when new evidence is invoked or the design itself evolves or changes over time.

This fact – that the designer needs to constantly revise his or her claim about the design and maintain consistency amongst the belief states pertaining to different components of the design – was recognised explicitly and formed a central element of one practical design approach called the *theory of plausible designs* (TPD) developed by Dasgupta and his collaborators (Aguero & Dasgupta 1987; Hooton *et al* 1988; Dasgupta 1989b, 1991). In fact, this work demonstrated quite clearly that the need for belief state revision during design is so immediate that automating this aspect of the design process is virtually imperative.

In this regard, AI in its more computational *persona* provides additional benefits since the technique of what in AI is called *truth maintenance* (Doyle 1979; de Kleer 1986) can be applied. Patel & Dasgupta (1991) describe one such computer-aided system for belief revision, the overall architecture of which is outlined in figure 2.

9. Understanding invention

We noted in § 6 that design problems may be mapped into some region of a space whose axes relate respectively to the 'structuredness' of the design problem and the amount of available knowledge about the relevant design domain (figure 1). The most intriguing region of this space concerns the *invention of original artifactual forms* which, as figure 1 suggests, is determined by situations where the design problem is highly ill-structured and virtually nothing is known about the nature or form of the artifact. Clearly, invention (or inventive design) is an aspect of the general problem of *creativity* in the artificial sciences and, thus, has many features in common with other kinds of creative acts, in particular, scientific discovery.

Consider, as a specific example, the invention by Wilkes (1951) of *microprogramming*. If creativity in the artificial sciences is strongly associated with the invention of new form then perhaps no better example can be found. For, the development of microprogramming led to an entirely new architecture for the control units of computers.

It is not our intention in this paper to present the technical details of microprogramming. For this, the reader may refer to any text on computer architecture (see, e.g., Dasgupta 1989a). However, the general history of the origins of microprogramming is well documented and is recounted here very briefly in order to illustrate the highly ill-structured nature of inventive design problems.

In the middle of 1949, the EDSAC computer, designed and built by Wilkes and his colleagues at the University Mathematical (later, Computer) Laboratory in Cambridge became the world's first fully operational 'stored program' computer. Soon after, Wilkes became preoccupied with the issue of *regularity* of computer designs. In particular, he was concerned with the fact that the organisation of EDSAC's control unit was irregular and *ad-hoc* (and, consequently, complex) in contrast to the highly regular organisation of EDSAC's memory unit.

What is interesting to note is that Wilkes *invented a problem*; moreover, it was a problem of a rather abstract kind, for it pertained to such qualities as 'regularity' and 'complex'. Wilkes problem was a *conceptual* problem (Laudan 1977; Dasgupta 1991) and such problems are inherently ill-structured. They are also of particular

interest in the context of creativity since the recognition of a conceptual problem by an individual is often motivated by philosophical or aesthetic viewpoints rather than strictly 'scientific' or 'technical' considerations.

In response to this particular problem, the principles of microprogramming were invented by Wilkes and first presented in a short paper (Wilkes 1951). Over the next two or three years, Wilkes and his colleagues developed the idea further and the first practical microprogrammed control unit was implemented in the EDSAC-2 which became operational in 1958 (Wilkes *et al* 1958).

Suppose now, we wish to investigate this process of invention; that is, we want to construct an explanation of how this cognitive act, performed by Wilkes, might have come about. How can we proceed?

Clearly, we cannot address the issue directly. What we might hope to do is to use the available historical evidence as recorded in the original papers, in subsequent retrospective accounts, in Wilkes's autobiographical memoirs and other sources (including personal communications and diaries) in order to construct a coherent structure of cognitive events which could serve as a plausible account of how Wilkes *might* have been led to his invention. The general idea, then, is to construct a *plausible model of creativity* which can explain this particular act of creativity in the realm of inventive design in a manner that is consistent with the historical evidence on hand.

We have recently completed a study of the invention of microprogramming using

---

A. The problem and observations

1. *The metaphrand*: The cognitive structure of creativity in the (natural and artificial) sciences.
2. *Relevant observations concerning the metaphrand*:

(a) A creative process involves changes in knowledge structures.
(b) Creativity involves the combination of known ideas or concepts with the resultant generation of novel ideas.
(c) The creative agent is purposeful and goal seeking.
(d) The creative process is protracted and evolving – and involves small changes of earlier ideas from moment to moment.
(e) Creative thinking entails searching for the 'right' ideas or concepts.

3. *Relevant observations concerning computation*:

(a) Computation entails the continuous modification of symbol structures.
(b) Computation begins with a goal and is directed, at all times, towards the attainment of the goal.
(c) Computations of a certain kind – 'knowledge level computations' – entail searching a space of possible and partial solutions with the aid of rules or heuristics to reduce the extent of search.

B. Formation of the metaphor

4. *The metaphor*: Scientific creativity as a cognitive process is like a knowledge-level computational process.
5. *The metaphier*: Knowledge-level computation.

C. Relevant knowledge about knowledge-level computation

6. The body of knowledge called (broadly) the 'Artificial Intelligence paradigm.'

D. Solution to the problem

7. A computation-based theory of scientific creativity.

Figure 3. The structure of a computational metaphorical model of scientific creativity.

the knowledge level form of the AI paradigm as the basis for a metaphorical model of creativity. As discussed in § 3, a metaphor entails the mapping of the unknown entity, the metaphrand, onto the known entity, the metaphier.

Figure 3 depicts the structure of our particular metaphorical model. As can be seen, relevant observations concerning the metaphrand and the metaphier (in this case, knowledge level computation) are listed in part A and are used to form the metaphor (part B). Once the metaphor is in place, the other relevant tokens of knowledge pertaining to the metaphier can be drawn upon to construct a 'computational theory of (scientific) creativity'.

It is not our intention, in this paper, to describe the details of this theory. We present it in great detail elsewhere (Dasgupta 1994). However, the general outline can certainly be given here. Basically, the 'computational theory of scientific creativity' is such that the process conducted by an agent leading to an original output can be described solely in terms of[5]

(i) Symbol structures that represent goals, solutions and knowledge.
(ii) Actions that operate upon symbol structures generating other symbol structures such that:
(iii) Each symbol processing transformation is only a function of the agent's knowledge and the goal(s) to be achieved at that moment of time.

In other words our metaphorical model of creativity is that of the knowledge level agent described in § 4. It is such that a creative process such as the one conducted by Wilkes can be described in the form of a knowledge-level process. The details of a plausible knowledge level process – plausible in that it is consistent with the historical and documented record – whereby Wilkes might have been led to the invention of microprogramming is described in Dasgupta (1994). Note that since the design process as previously described in § 5 is itself a knowledge-level process, we arrive at the tentative conclusion that invention involves essentially the same kind of cognitive process as incurred in less creative acts of design. This is consistent with the conclusions reached by some others – both psychologists and computer scientists – who have investigated creativity (Newell *et al* 1962; Perkins 1981; Weisberg 1986; Langley *et al* 1987).

10. Conclusions

A substantial part of the intellectual content of the artificial sciences is contained in the activity we call design. A central aim of design theory is to construct testable, explanatory models of the design process that will serve to enhance our understanding of the processes whereby artifacts are or can be designed. The range of design problems include, at one extreme, routine design where the problem is very well-structured and there exists a large body of knowledge concerning the class of artifacts in question and, on the other, invention where the problem is highly conceptual, abstract and ill-structured and very little is known about the nature and form of the artifact.

[5] The criteria whereby an agent's output is deemed original must, of course, be quite independent of the theory of creativity. The latter attempts to explain how a creative process, i.e. a process the output of which is known to be original, may work. Elsewhere (Dasgupta 1993, 1994) we discuss in some detail the independent criteria whereby some cognitive act of discovery or invention may be judged to be original.

In this paper, we have discussed how some of the basic concepts underlying the discipline of artificial intelligence can serve to construct an explanatory Kuhnian paradigm within which the design process can be examined. The concept of a knowledge-level process provides such a paradigm. We have described here some of the implications of the knowledge-level model of design for our understanding of design and how the same model can serve to enhance our understanding of the act of invention.

References

Aguero U, Dasgupta S 1987 A plausibility driven approach to computer architecture design. *Commun. ACM* 30: 922–932

Alexander C 1964 *Notes on the synthesis of form* (Cambridge, MA: Harvard University Press)

Bohm D, Peat F D 1987 *Science, order and creativity* (New York: Bantam)

Brown D C, Chandrasekaran B 1989 *Design problem solving* (London: Pitman)

Chandrasekaran B 1990 Design problem solving: A task analysis. *AI Mag.* Winter: 59–71

Cohen I B 1985 *Revolution in science* (Cambridge, MA: Harvard University Press)

Coyne R D, Rosenman M A, Redford A D, Balachandran M, Gero J S 1990 *Knowledge based design systems* (Reading, MA: Addison-Wesley)

Cross N (ed.) 1984 *Developments in design methodology* (New York: John Wiley & Sons)

Dasgupta S 1989a *Computer architecture: A modern synthesis. Vol. 1. Foundations* (New York: John Wiley & Sons)

Dasgupta S 1989b The structure of design processes. In *Advances in computers* (ed.) M C Yovits (New York: Academic Press) vol. 28, pp. 1–67

Dasgupta S 1991 *Design theory and computer science* (Cambridge: University Press)

Dasgupta S 1992 Two laws of design. *Intell. Syst. Eng.* 1 (Winter), 2: 146–156

Dasgupta S 1993 Creativity, invention and the computational metaphor: Prologemenon to a *case study. In Creativity and artificial intelligence* (ed.) T Dartnall (Boston: Kluwer) (forthcoming)

Dasgupta S 1994 *Creativity in invention and design* (Cambridge: University Press)

de Kleer J 1986 An assumption based TMS. *Artif. Intell.* 20: 127–162

Doyle J 1979 A truth maintenance system. *Artif. Intell.* 12: 231–272

Gero J (ed.) 1991 *Artificial intelligence in design'91* (Oxford: Butterworth–Heinemann)

Gruber H 1981 *Darwin on man: A psychological study of scientific creativity* 2nd edn (Chicago, IL: Univ. of Chicago Press)

Holmes F L 1985 *Lavoisier and the chemistry of life* (Madison, WI: Univ. of Wisconsin Press)

Holton G 1952 *Introduction to concepts and theories in physical science* (Reading, MA: Addison-Wesley)

Hooton A, Aguero U, Dasgupta S 1988 An exercise in plausibility driven design. *Computer* 21: 7

Jaynes J 1976 *The origin of consciousness in the breakdown of the bicameral mind* (Toronto: Univ. of Toronto Press)

Jones C 1980 *Design methods: Seeds of human future* 2nd edn (New York: John Wiley and Sons)

Jones C, Thornley D G (eds) 1963 *Conference on design methods* (Oxford, New York: Pergamon/ Macmillan)

Kuhn T S 1962 *The structure of scientific revolutions* (Chicago, IL: Univ. of Chicago Press)

Kuhn T S 1970 Reflections on my critics. In *Criticism and the growth of knowledge* (eds) I Lakatos, A Musgrave (Cambridge: University Press) pp. 231–278

Kuhn T S (ed.) 1977 Second thoughts on paradigms. In *The essential tension.* (Chicago, IL: Univ. of Chicago Press)

Lakatos I 1978 *The methodology of scientific research programmes* (Cambridge: University Press)

Lakatos L, Musgrave A (eds) 1970 *Criticism and the growth of knowledge* (Cambridge: University Press)

Langley P *et al* 1987 *Scientific discovery* (Cambridge, MA: MIT Press)

Laudan L 1977 *Progress and its problems* (Los Angeles: Univ. of California Press)

Laudan L 1984 *Science and values* (Berkeley, CA: Univ. of California Press)
Mostow J 1985 Towards better models of design processes. *AI Mag.* Spring: 44–57
Newell A 1982 The knowledge level. *Artif. Intell.* 18: 87–127
Newell A, Shaw J C, Simon H A 1960 Report on a general problem-solving program for a computer. *Information processing* (Paris: UNESCO) pp. 256–264
Newell A, Shaw J C, Simon H A 1962 The processes of creative thinking. In *Contemporary approaches to creative thinking* (eds) H E Gruber, G Terrell, M Wertheimer (New York: Atherton) pp. 63–119, 933–951
Newell A, Simon H A 1972 *Human problem solving* (Englewood-Cliffs, NJ: Prentice-Hall)
Newell A, Simon H A 1976 Computer science as empirical inquiry: Symbols and search. *Commun. ACM* 19: 113–126
Papert S 1988 One AI or many. *Daedalus* Winter; also *Proc. Am. Acad. Arts Sci.* 117: 11–14
Patel S, Dasgupta S 1991 Automatic belief revision in a plausibility-driven design environment. *IEEE Trans. Syst. Man Cybern.* 21: 933–951
Perkins D N 1981 *The mind's best work* (Cambridge, MA: Harvard Univ. Press)
Petroski H 1985 *To engineer is human* (New York: St. Martin's Press)
Petroski H 1991 Paradigms for human error in design. *Proc. 1991 NSF Design and Manufacturing Systems Conf.* Austin TX., Jan, pp. 1132–1146
Pye D 1964 *The nature of design* (London/New York: Rheinhold/Studio Vista)
Reiter R 1987 Nonmonotonic reasoning. *Annu. Rev. Comput. Sci.* 2: 147–186
Shapere D 1964 The structure of scientific revolutions. *Philos. Rev.* 73: 383–394
Simon H A 1973 The structure of ill structured problems. *Artif. Intell.* 4: 181–200
Simon H A 1976 *Administrative behavior* 3rd edn (New York: The Free Press)
Simon H A 1981 *The sciences of the artificial* 2nd edn (Cambridge, MA: MIT Press)
Simon H A 1982 *Models of bounded rationality* (Cambridge, MA: MIT Press)
Suppe F (ed.) 1977 The search for philosophic understanding of scientific theories. In *The structure of scientific theories* (Urbana, IL: Univ. of Illinois Press)
Thagard P 1988 *Computational philosophy of science* (Cambridge, MA: MIT Press)
Thagard P 1990 The conceptual structure of the chemical revolution. *Philos. Sci.* 57: 183–209
Weisberg R W 1986 *Creativity: Genius and other myths* (New York: W H Freeman)
Wilkes M V 1951 The best way to design an automatic calculating machine. *Rept. Manchester Univ. Comput. Inaugural Conf.*, Manchester, UK
Wilkes M V, Renwick W, Wheeler D J 1958 The design of a control unit of an electronic digital computer. *Proc. Inst. Electr. Eng.* 105: B121