Understanding Problem Structure as Heuristic Formation in Design

Brian Logan

Department of Artificial Intelligence
University of Edinburgh

Abstract At least two main strands can be distinguished in the design theory literature: that design is a knowledge-based process; and that design is a learning process. In this paper we attempt to relate these two views of design by showing that learning within and between design problems is a necessary consequence of a knowledge-based view of design. Drawing on work in artificial intelligence, we develop a model of design as a knowledge-based learning process based on the successive refinement of a body of core heuristics and embed this model within the abduction-deduction-induction framework proposed by March (1976).

1 Introduction

At least two main strands can be distinguished in the design theory literature: that design is a knowledge-based process; and that design is a learning process. For example, Schon (1988) has argued that the patterns of inference shared among designers are not significantly different from reasoning in everyday life. What distinguishes designers is not some mystical ability but the accumulated knowledge the designer brings to bear upon a problem. Others, such as Bazjanac (1974) have argued that design can be viewed as a learning process in which the objective is one of understanding the structure of the problem. In this view design proceeds through a process of analysis-through-synthesis in which the problem is explored through a series of attempts to create solutions and understand their implications in terms of design criteria. In this paper we attempt to relate these two views of design by showing that learning within and between design problems is a necessary consequence of a knowledge-based view of design. Drawing on work in artificial intelligence, we develop a model of design as a knowledge-based learning process based on the successive refinement of a body of core heuristics and embed this model within the abduction-deduction-induction framework proposed by March (1976).

In section 2 we discuss the nature of design problems. A design problem is characterised as as one in which both the objectives and the means available for achieving these objectives are (of necessity) initially only poorly defined. In section 3 some observations concerning the nature of the design process based on this characterisation are presented, and it is argued that the fundamental objective in design is one of understanding the structure of the problem. In section 4 the role of knowledge in this process is examined in more detail and we argue that design can only proceed through the development of new relationships and strategies within the context of the current design problem. In section 5 we investigate the nature of heuristic rules in some detail. The range of an heuristic is characterised as the task domain in which it is applicable and we show that for any given set of heuristics there exists (at least) one problem which cannot be solved using these heuristics. In section 6 we present an alternative view of the development and use of relationships in design based on these observations, which focuses on the critical role of learning in design. The idea of a set of 'core heuristics' is introduced and we argue that the elaboration and adaption of these heuristics can be seen as a simple model of this complex process. Section 7 develops a model of this learning process within the framework of the model of design proposed by March based on the three logical operations of abduction, deduction and induction. In particular, we attempt to clarify the role of induction in the formation of relationships between criteria and its relationship to the process of theory formation. In the final section we argue that despite its limitations, the simple model outlined in the previous sections is useful in gaining some insight into the role of learning in the design process.

2 The Nature of Design Problems

We begin by presenting our characterisation of design problems in more detail. At its most general level, a design problem is concerned with the production of a description of of an artefact or process which meets a given set of objectives or requirements. These requirements are often initially ill-defined and may be in conflict. In general they will not all be equally important; legislative controls are often value free, whereas user requirements may be modified during discussions with the client, and constraints generated by the designer may be extensively revised, or even abandoned altogether during the design process (Lawson, 1980). However, the real difficulty is that these objectives cannot easily be related to one another. Lawson (1980) states: "the relative importance of the various requirements change constantly during the design process as the designer's value system is itself affected by the exploration of objectives and what he finds to be possible". The value judgments regarding 'trade-offs' between criteria are therefore context dependent, and the balance of satisfaction for such requirements may not be clear until the designer explores the various possibilities in appropriate detail. Such value judgments apply not only to the 'qualitative' criteria such as aesthetics, but also to the relative importance of quantitative criteria which themselves may be susceptible to objective measurement. Questions about which are the most important problems and what kinds of solution most successfully solve these problems are also value laden, and the answers given by designers to these questions are therefore frequently subjective and highly context dependent.

The nature of the 'real' problem is thus often not apparent but must be discovered; problems may suggest certain features of solutions, but these solutions in turn create new and different problems. The initial expression of the problem is often misleading, and designers must typically expend considerable effort in identifying the actual nature of the problem which confronts them. Design problems have no obvious or natural boundaries, but rather seem to be organised roughly hierarchically. Many elements of the problem cannot be expected to emerge until some attempt has been made at generating solutions. Given the essentially subjective nature of design it is inevitable that some aspects of the problem will remain either unrecognised or undeveloped for much of the design process. As a result design problems are full of uncertainties both about objectives and their relative priorities, and both priorities and objectives are likely to change as solutions emerge. Simon (1973) calls such problems 'ill-structured' and argues that any problem with a large base of potentially relevant knowledge falls into this category. The design task is ill-structured in this sense in a number of respects. There is initially no definite criteria to test a solution, much less a formal process to apply the criteria. In addition the problem space cannot be completely defined due to a radical lack of knowledge. Also, while the set of alternative solutions may be given in a certain abstract sense, it is not given in the only sense that is practically relevant. As a result there can never be an exhaustive list of all the possible solutions to such problems.

Design problems are therefore often multidimensional and highly interdependent. It is rare for any part of a design to serve only one purpose, and it is frequently necessary to devise a solution which satisfies a whole range of requirements. Design decisions may have results other than those intended, which highlight previously unrecognised criteria and relationships. In many cases the stated objectives are in direct conflict with one another and the designer cannot simply optimise one requirement without suffering losses elsewhere. For example, though enlarging a window may well let in more light and give a better view, it will also result in greater heat loss and may create greater problems of privacy. Different trade-offs between the criteria result in a whole range of acceptable solutions, each likely to prove more or less satisfactory in different ways to different clients and users. It is the very inter-relatedness of these factors which is the essence of design problems rather than the isolated factors themselves, and it is the structuring of relationships between these criteria that forms the basis for the design process (Lawson, 1980). The fundamental objective thus becomes one of understanding the structure of the problem (rather than the solution), and analysing the inter-relationships between criteria to gain some insight into the relationships between each individual design decision and all of the other decisions which together define the solution.

3 The Nature of the Design Process

The designer's exploration of this structure begins with the initial formulation of the problem. To a large extent, a design problem has no inherent structure; it acquires structure as solutions are proposed and problems are reduced to subproblems. In a very real sense the relationships between criteria can be seen as a function of the approach to design embodied in the proposed solution rather than as inherent in the problem itself. This initial formulation forms the basis of subsequent exploration of the problem.

Consider, for example, the problem of providing a particular view from the living room in designing a house.[1] Such a requirement may have been specified in the original problem description or it may have been generated during the design process. In either case, an architect might choose to formulate the associated design problem in terms of some standard solution, as a problem of the arrangement of the living room and the placement of a window in a way which will provide the desired view from the relevant areas of the room. Through such a formulation of the design problem the designer has also formulated the general form of the solution; any particular design solution is determined by a specific placement of the window and a specific disposition of the living room. The designer then proceeds to explore the implications of this particular design decision. In doing so he or she may make further design decisions and consider a number of design alternatives. This process of exploration may lead to the discovery that in providing the desired view it becomes impossible to maintain the relationship between the living area and the entrance to the house, or that that the basic structural system of the house will not allow the positioning of a window of the required size in the desired location. As a result the designer comes to realise that the problem of providing a view from the living room has other aspects and its solution may involve finding a more appropriate layout for the house on the site, or the redesign of the structural system in a way which will accommodate the desired window. The formulation of both the problem and the solution may therefore change as a consequence of an attempt to solve a particular problem.

[1]This example is based on one given by Bazjanac (1974).

It might be argued that in this case the original formulation of the problem was unrealistic and that an experienced designer would approach the problem at a more appropriate level. However such an argument misses the point. In trying to develop a solution to a particular design problem even the most experienced designer will gain new insights which necessitate the redefinition of the problem and suggest alternative solutions. The need to understand (at whatever level) the details of a particular case and how they interact are in a sense what makes a design problem a design problem.

As a solution develops it provides an increasingly detailed context against which to test the designer's hypotheses, and the evaluation of a proposal can result in the discovery of previously unrecognised relationships and criteria. In a sense later decisions are constrained by earlier decisions in that they are taken within the context of an existing partial solution, and each decision further limits the range of possible alternatives. Solutions to particular subproblems are apt to be disturbed or undone at a later stage when new aspects are attended to and the considerations leading to the original solution are forgotten or not noticed. Such side effects accompany all complex design processes. As a result, while the final solution may satisfy all the requirements that are evoked when it is tested, it may violate some of the requirements that were imposed (and temporarily satisfied) at an earlier stage in the design. The designer may or may not be aware of these violations. Other appropriate design criteria may simply remain dormant, never having been evoked during the design process. The development of a design is thus constrained by what best fits the knowledge the designer has at that time.

The formulation of the problem at any stage is not final; rather it reflects the designer's current understanding of the problem. As the design progresses the designer learns more about possible problem and solution structures as new aspects of the situation become apparent and the inconsistencies inherent in the formulation of the problem are revealed. As a result, designers gain new insights into the problem (and the solution) which ultimately result in the formation of a new view; the problem and the solution are redefined. This process of exploration and redefinition continues until one or more of the following conditions is met (Bazjanac, 1974):

- the incremental gain in knowledge has become insignificant and the understanding of the problem (and the solution) cannot change enough to warrant further redefinition. (i.e. the designer has reached the limits of his or her understanding); or
- the available resources (primarily time) have become exhausted.

There is no meaningful distinction between *analysis* and *synthesis* in this process; problems and solutions are seen as emerging together rather than one logically following from the other. The problem is explored through a series of attempts to create solutions and understand their implications in terms of other criteria. The designer comes to understand the critical relationships and possible forms as a solution evolves. Between generic solutions planning is less a search for the best solution than an exploration of the compromises that give sufficient solutions. These explorations help the designer appreciate which requirements may be most readily achieved. As part of this process, the designer learns which criterion values will achieve the design requirements and how much variation of these values can be tolerated while still achieving acceptable performance, the implications of achieving the current goal, and any other decisions required to make the attainment of these goals consistent with the existing solution.

Learning more about the structure of the problem is the most important part of this process. The fundamental objective becomes one of understanding the structure of the problem, with a major part of the effort in design being directed towards structuring problems and only a fraction of it devoted to solving them once they have been structured (Simon, 1970). The design process can be viewed more generally as a process of discovering information about problem structures that will ultimately be valuable in developing possible solutions.

4 Knowledge in Design

The generation of solutions draws on an extensive knowledge of design methods, strategies and solutions to previous problems. Design proposals are not produced blindly but result from a general understanding of the kinds of solutions which may be appropriate in a given situation, and how these solutions may be pursued. An important component of this knowledge is what might be termed 'compiled experience'. The role of *a priori* knowledge derived either from a familiarity with related problems or, in the form of published guides and standards, has been widely recognised in studies of design. Foz (1972) has shown how exploration of the problem evokes previously known solutions from memory. These examples are used as 'guides' or 'templates' for analysing or developing possible solutions in terms of the problem requirements. Akin (1978) discusses the use of 'problem transformations', which "make the current solution more specific, such as a precompiled solution, an analogous solution, a generic solution etc. ... if explicit transformations are not possible at the time, use previous experience to assume that certain aspects of the current solution can be further specified". More recently, Gero et al (Gero, 1987; Gero, Maher & Zhang, 1988; Oxman & Gero, 1988) have proposed that design knowledge is stored and retrieved in a series of abstract schemata called 'prototypes' — "generalised groupings of elements in a design domain ... from which instances of elements can be derived".

These approaches to design may be broadly characterised as 'knowledge-based', in viewing the design process as a series of problem transformations governed by 'rules' or 'codes' linking design solutions and abstract requirements. There are clear parallels between, for example, Foz's 'templates', Akin's 'problem transformations' and Gero's concept of 'prototypes'. Common to all these approaches is the idea that design proceeds through the utilisation of an organised body of *a priori* knowledge, which is used both to structure and understand the design problem and which forms the basis of design hypotheses. In this view there exists at any given time a list of current goals together with a (potentially very large) set of heuristics, strategies, previous examples etc. embodying a set of perceived relationships between the solution and criteria spaces. These relationships function as 'production rules', mapping a problem expressed in terms of abstract requirements onto some solution or class of solutions which satisfies these requirements. The boundary between the criteria and solution spaces moves to include as criteria solutions to previous problems, as the problems represented by the criteria are reduced to sets of simpler subproblems whose solution is known or which are at least easier to solve. This process continues until no further reduction is possible or until the resulting problem is deemed to be no longer the concern of the designer; e.g. the problem of how to design a beam to support a given load or the problem of how to construct a wall of a given size in a given position.

Steadman (1979) has argued that in architectural design this body of general or collective knowledge is perpetuated through architectural education, architectural journals and publications and the study of existing buildings. However it is not, with certain exceptions, of an organised, explicit or scientific nature. Rather, empirical experience of a range of related designs provides a body of knowledge and understanding on the basis of which it is possible to build a generalised theory (or theories) of a class of artefacts, which is used to extrapolate, beyond the tried cases, to hypothetical but related designs yet to be constructed. It is this body of knowledge, concerning, for example, the relation of physical performance to shape, which informs the creation of solutions.

However when we turn our attention to a detailed consideration of these rules a major difficulty immediately becomes apparent. Design, by its nature, is largely concerned with the specification of objects that are unique. (If an object already exists which is recognised as satisfactorily achieving all the design goals there is by definition no design problem.) The question then arises: how can any necessarily finite collection of relationships cope with the infinite variety of possible design problems?. Or, more precisely, what happens when no rule can be found which is appropriate to the current problem.

A complete answer to this question is impossible given our current understanding of the design process, however we shall attempt to develop a small part of such an answer in the remainder of this paper. Briefly, we shall argue that, in general, design can only proceed through the development of new relationships and strategies within the context presented by the design problem. In the remainder of this paper we present a model of the development and application of design knowledge based on the successive refinement of a set of heuristic rules and embed this model within the model of the design process proposed by March (1976).

5 The Nature of Heuristics

An *heuristic* is a process or procedure that may solve a problem, but offers no guarantee of doing so (Newell, Shaw & Simon, 1963). The study of heuristics as a separate discipline effectively began with Polya (1945) who traced its origins back to Liebnitz, Descartes and even Pappus. More recently Pospelov and Pushkin (1972) have tried to define the field as "the science which studies the laws governing the design of new actions in new situations". In the main experience has tended to come from fields other than design, most notably mathematics (Davis & Hersh, 1980; Polya, 1945), and artificial intelligence (Lenat, 1982). Despite the major differences in subject matter and approach, we shall argue that the results of these studies are of relevance to design.[2] The material in this section draws on the work of Lenat (1982; Lenat, 1983a; Lenat, 1983b). Below we present a summary of Lenat's work and in subsequent sections we attempt to develop these ideas and relate them to the design activity.

Lenat (1982) argues that the power of heuristics results from a kind of two dimensional continuity over situations and actions: "If an heuristic H was (or would have been) useful in situation S then it is likely that heuristics similar to H will be useful in situations similar to S". In other words, if we could somehow graph the function *appropriateness(action, situation)*, that function would be continuous in both variables and vary very slowly. This is obviously an idealisation; there are many ways of characterising situations (the problem domain, the difficulty of the problem, the time available for its solution etc.) and many measures of appropriateness (the quality of the resulting solution, the computational cost of the using the heuristic etc.) with the result that we can do little more than estimate the utility of a particular action on a few criteria in a small number of situations. Nevertheless, Lenat argues that this assumption underlies much of the utility of heuristics in problem-solving and that it is often useful to behave as though the assumption were true, i.e. to behave as though it were true that the function *appropriateness(action, situation)* exists and is continuous and time-invariant).

[2]Note that in doing so we do not wish to adopt a 'realist' position with respect to design rules. We are not implying that the 'rules' used by designers can be made explicit or even that such rules exist, only that the behaviour of designers can be modelled using such rules.

Lenat develops several corollaries of this assumption:

Corollary 1 *If an action A is appropriate in a situation S, then A is also appropriate in most situations which are very similar to S.*

For a given action, its *appropriateness* is a continuous function of the situation. If the situation changes only slightly then the judgement as to which actions are appropriate changes only slightly. This is the basic justification for reasoning by analogy; if something worked in a similar situation in the past, it is likely that it will work in current problem.

Corollary 2 *If an action A is appropriate in a situation S, then so are most actions which are very similar to A.*

For a given situation, *appropriateness* is a continuous function of actions. If an action is particularly useful (or harmful) in some situation, it is likely that any very similar action will have similar consequences. This is the basic justification for 'satisficing', of accepting a solution which is 'good enough'; given that any similar action will, in general, have similar consequences, it is not worth searching for an optimal answer.

Corollary 3 *If an action A would have been appropriate in a situation S, then the rule "If the current situation is similar to S then try A" may be useful in the future.*

It is cost effective to form and use heuristics which would have helped in the past. This is basic justification for the utility of memory. If we conclude (via hindsight) that a rule would have been useful in the past, then it is likely that it will be of use in the future.

In the remainder of this paper we attempt to extend this framework to design. We begin by considering the nature of heuristics in more detail.

5.1 The Power of Individual Heuristics

A *situation* is a description of a problem or task: a list of goals together with the set of constraints, background assumptions and any existing partial solution which forms the context of the problem. For a body of heuristics to be effective in in guiding action, each heuristic must specify those situations in which its action(s) are especially appropriate or inappropriate. We can view an heuristic as a simple production rule of the form:

$$\text{if } \langle \textit{condition} \rangle \text{ then } \langle \textit{action} \rangle$$

If the *condition* is true (or approximately true) in the current situation then the *action* may be an appropriate one to try. Such rules may be either analytic or synthetic. For example, if the task is to design a beam capable of supporting a given load, the heuristic would be thought of as synthetic. Conversely, if the objective is to predict the heating energy requirements of a proposed design the heuristic would be thought of as analytic. In both cases there are a variety of approaches which could be used, ranging from rules of thumb to sophisticated finite element methods. More generally, there are also rules which are useful in less well defined situations; for example, what kind of structural system to use given the type of building and the cost constraints, or the evaluation of the layout of a building given the design requirements and the characteristics of the site.

Each heuristic will be more or less useful in a given situation. The utility of an heuristic is zero in some situations (where the heuristic is not considered relevant, i.e. the condition is false) and is more or less positive in others (where the condition is true).[3] A problem can be characterised in many different ways (e.g. the scope of the problem, the degree of difficulty, the resources available for a solution etc. — including the context of any partial solution which might exist). A given problem can be identified with a particular set of values on each of the situation dimensions. Naively we might classify problems into easy problems and hard problems or structures problems and environmental problems for example. Similarly, there are many different ways of characterising the utility of an heuristic (e.g. its reliability, the resources required to perform the action etc.). Each heuristic will have a particular level of utility on each utility dimension for a given problem. For example, in a given situation a particular heuristic might be quick to apply but unreliable. Note that while there may appear to be an overlap — the speed with which a solution is required *vs.* the speed with which a solution is produced for example — these dimensions are distinct. Situation dimensions characterise the problem whereas utility dimensions characterise the heuristic.[4] If there are n utility dimensions and m situation dimensions then the utility of an heuristic on a given utility dimension can be represented by a point in an $n+m$-dimensional space. For a given problem we will be at some point on each of the situation axes $\mathit{situation}_{1-m}$ and the utility of an heuristic on some utility measure $\mathit{utility}_i$ is given by

$$\mathit{utility}_i = f_u(\mathit{heuristic}, \mathit{situation}_{1-m})$$

For any pair of utility and situation dimensions $(\mathit{utility}_i, \mathit{situation}_j)$ and some assignment of values to the remaining situation dimensions we can imagine graphing the utility of an heuristic on the dimension $\mathit{utility}_i$ for varying values of $\mathit{situation}_j$. For n utility dimensions and m task dimensions there are $n \times m$ such power curves for each heuristic. Note that in many cases this will be a constant function. For example, while the reliability of an heuristic might be expected to depend on the problem domain or the degree of difficulty, it is unlikely to be affected by the time available for a solution.

The utility of an heuristic in any given situation is a function of both the utility of the condition and the utility of the action

$$\mathit{utility}_i = f'_u(\mathit{utility}_{ic}, \mathit{utility}_{ia})$$

where

$$\begin{aligned} \mathit{utility}_{ic} &= f_c(\mathit{condition}, \mathit{situation}_{1-m}) \\ \mathit{utility}_{ia} &= f_a(\mathit{action}, \mathit{situation}_{1-m}) \end{aligned}$$

If, for example, both the condition and the action of an heuristic have an associated cost in time or resources and an associated reliability then the total cost of using the heuristic will be the the cost of evaluating the condition (to find out if the situation is of the right type) plus the cost of performing the action, whereas

[3]Lenat argues that the utility of an heuristic can be negative in some situations, i.e. the heuristic appears to be relevant (the condition matches the situation) but is in fact counterproductive — it gives the 'wrong' or a poor solution, or takes longer than alternative approaches etc. depending on the utility dimension being considered. However this implies a normative theory of utility, and it is not clear from Lenat's argument where the zero point should be located. Assuming that we have utility axes at all, it is simpler for all utilities to be positive with axes ordered as required (e.g. lower cost = greater utility).

[4]We could chose to view the range of situations in which an heuristic is applicable as a kind of utility, with those heuristics applicable in a wider range of situations having greater utility. However this would obscure certain aspects of the argument.

the overall reliability of the heuristic may be the product of the reliability of the condition and the reliability of the action. An heuristic such as "if the problem is like one we have solved previously, then copy or adapt the previous solution" may have high utility (e.g. low cost or high reliability) for 'typical' problems. The overall cost of the heuristic will depend on the cost of the search for the previous solution (the condition) and the cost of copying or adapting that solution (the action). The reliability of the heuristic will be some function of the reliability of the search (e.g. the percentage of previous solutions examined, or the failure to notice significant differences between the previous situation and the current problem) and the reliability of adapting the previous solution (e.g. were any irrelevant details carried over from the previous solution).

Estimating the utility of a condition is often difficult. While it may be possible to determine the cost of evaluating a condition, it is often much harder to estimate its reliability in selecting the intended set of situations. This is essentially the utility of planning, of determining how much time time or resources should be allocating to deciding what to do next — how long to spend classifying the situation, or searching for ways in which the current situation is similar to or different from previously encountered situations. The utility of planning varies both with the difficulty of the problem and cost of the proposed action. For trivial problems, planning may be largely a waste of time (almost any heuristic will solve the problem), however for complex problems or in situations where there are many interacting problems planning may be essential to avoid wasted effort. Estimating the anticipated utility of an action in a given situation is also often difficult. The estimate of utility may be based simply on the subjective experience — how many times has the action been successful in similar situations in the past — or it may be some measure of the cost of performing the action in time or resources. If an action has frequently been successful in similar situations in the past, it may be a good heuristic to try in the current situation (depending on other factors such as the time required to perform the action against the time available to solve the problem). On the other hand if it is only occasionally successful, but on those occasions where it did succeed, it produced a 'good' solution and is easy to apply, it may still be worth trying.[5]

We can trade-off utility between the condition and action to achieve a given level of utility for the heuristic as a whole. For example, in solving structures problems the heuristics "always use finite element methods" and "identify those situations in which simpler analytic methods may be used" may produce an equally 'good' solution in the same amount of time. However, in general, a good heuristic is one in which there is a high degree of overlap between those situations where the condition has high utility and those situations where the action has high utility. If the two sets only partially overlap, the resulting heuristic will be unreliable — in some situations it will be successful while in others it will fail (e.g. take too long or produce the 'wrong' answer). If the two sets of situations are disjoint the resulting heuristic will have low or zero utility even if the condition and action are themselves successful. As the cost of the proposed action in time or resources increases, so it becomes worthwhile spending more time deciding which approach to adopt.[6] Finite element methods are applicable to most structural design problems, but they are cost effective only in those situations where simpler methods are unavailable and careful consideration is often given as to whether the problem is one which requires such methods.

[5]If on the other hand it is expensive to apply, it also suggests that it may be worth trying to isolate the characteristics of those situations in which it was successful. Such heuristics about generating heuristics are discussed in more detail below.

[6]Unless, of course, there are no alternative approaches available. In such a situation, the risk of failure increases dramatically.

5.2 The Space of Heuristics

In reality, of course, a designer will draw on a wide range of heuristics in solving any given problem. In general a set of heuristics will have higher utility than any of its members considered in isolation. Different heuristics are applicable in different situations and using several heuristics in combination may be more effective than any single heuristic, for example if the action of one heuristic matches the condition of another. However the overall utility of a set of heuristics is not simply the sum of the utilities of the individual heuristics. The interactions between heuristics are often quite strong and independence is the exception rather than the rule. Often two heuristics will be different methods of achieving the same result and the overall utility of the set is not greatly increased by having both of them present.[7] While heuristics sometimes interact synergystically, it is often the case that there are several internally consistent but mutually inconsistent sets of heuristics within the total set. Using only heuristics from one of these sets will result in a consistent solution to the problem, but unstructured use of heuristics from different sets results in inconsistencies within the solution. Such interactions are common in domains such as design, where the attempt to achieve a particular requirement often generates problems elsewhere. Simon (1973) argues that good heuristics are those which minimise such interactions and that part of learning to be a designer is learning which approaches to solving the problem are mutually consistent.

Using a particular heuristic in a given situation constrains which other heuristics can be employed in the new situation which results. As in a game of chess, any particular move in a given situation constrains future moves. In theory, there is, in any given situation, a best possible move, which can be found by exhaustive enumeration of all possible moves and countermoves from the current position. Similarly, in any given design situation, we can imagine computing the 'global utility' of an heuristic such that the heuristic with the highest global utility ultimately leads to the best overall solution achievable in the current situation. Note that 'best' here means best on all utility dimensions (weighted for the relative importance of each utility dimension) as these appear in the context of the completed design, i.e. modified by the designer's increased understanding of what is possible and the implications of the various alternatives. Computing such a global utility is obviously impossible — it would involve the exhaustive enumeration of all designs possible in the current situation (i.e. all possible designs, including backtracking and the abandonment of the current partial solution) — and would require a well defined space of possible designs. Each individual design decision must be made in the context of the problem as a whole. This context includes both the problem specification and the rest of the design solution — both those decisions made to date and those that will be made in the future, as the decision not only has to satisfy the current goals it must continue to satisfy them in the context of the final solution when all of the remaining decisions have been made. Not only is the set of future decisions undefined but the part which is defined, the current problem definition, will change during the course of the design as it is modified by the designer's increased understanding of the implications of the current solution. Like chess programs, designers must work with essentially local information and any estimate of the utility of an action is typically limited to the immediate consequences of that action.

More generally the underlying structure of the space of heuristics — the number and type of heuristics and the concepts they relate — depends on how the space of concepts is conceptualised into problems. We can associate with each heuristic one or more domains, or 'sets of problems', for which the heuristic has non-zero utility, where 'domain' is defined as a range of values for the problem dimensions of

[7] However the redundancy provided by multiple heuristics can be important in the (consistent) attainment of multiple goals.

subject matter, complexity, time available for solution etc., which together bound a region of the problem space. The set of situations (defined as a range of values for each situation dimension) for which a heuristic achieves some given level of utility on each utility dimension is given by

$$\mathit{situation}_{1-m} = f_d(\mathit{heuristic}, \mathit{utility}_{1-n})$$

Conversely, the set of heuristics appropriate to a given problem or set of problems (i.e. to a domain) is given by

$$\mathit{heuristic} = f_h(\mathit{utility}_{1-n}, \mathit{situation}_{1-m})$$

To solve any given problem, it is necessary to find an heuristic which has satisfactory performance on all of the relevant utility axes. In other words, given a minimum performance level for each utility dimension and the position of the problem on each of the situation dimensions, the set of heuristics relevant to this problem lie within the hypervolume defined by the situation/utility parameters.

The set of heuristics associated with each domain can be thought of as being in some sense relevant to the set of problems which constitute the domain (and all its subsets) in having non-zero utility for that set of problems. Domains do not partition the space of concepts — in a sense everything is linked to everything else. Rather, they are largely conceptual devices for structuring our consideration of the problem. For example, it is unlikely that in general a detailed consideration of window type will be required in general layout planning, although it may be. Lenat (1982) argues that one of the major tasks in mastering any domain is learning the proper level(s) at which to state and use heuristic knowledge. If the heuristics are too small their domain of application becomes too narrow to make them worth remembering relative to the range of problems associated with the domain. They stop being meaningful pieces of knowledge and risk having many stray interactions.

A problem, such as 'scheme design' or the 'design of a structural system', can be seen as a meta-level name of a set of object-level concepts. The attributes of the problem, such as level of difficulty, etc., are in fact attributes of this set of concepts.[8] The problem of, for example, 'designing a structural system' or, more generally, of 'structural design', can be viewed as the problem of finding some relationship between a structure and a set of elements comprising that structure; where 'structure' and 'elements' are variables ranging over possible structures and elements respectively, that is, over subsets of the set of structural concepts. The task of 'solving a problem' becomes one of relating a set of goal concepts to some, as yet unspecified, set of hypotheses. While the set of possible hypotheses does not form part of the current problem description, the set of possible or candidate concepts from which the hypotheses are constructed is known in some sense through experience of similar problems. It is this set of relevant concepts which forms the problem 'field' or domain, and which an heuristic attempts to structure by partitioning it into the (sub)set of concepts which will form part of the current solution, and those that will not. Heuristics can be seen as structuring relationships between sets of concepts (between sets of 'wholes' and 'parts' or 'causes' and 'effects'), and hence the relationships between heuristics — viewed as meta-level relationships such as more-general-than, more-powerful-than — are largely determined by the resulting partitioning of the space of concepts into problems. A corollary of this is that the way problems are seen in large part determines the way in which they will be solved.

[8]The term 'problem' as used here refers to the class of problems associated with a given domain independent of any particular goal or set of goals.

5.3 The Space of Situations

Different heuristics will have different utilities in a given situation. If an heuristic has lower utility than another on all utility measures in all situations then we say the first heuristic is *dominated* by the second. Clearly such a set is context dependent. New approaches are always being discovered which are (or are perceived to be) better in some respect than those they replace. However for any given set of heuristics (representing some understanding of the world or level of problem solving ability) it is possible to extract the non-dominated subset. Such heuristics can be thought of as the 'best available'. This is clearly an idealisation, however nothing hinges on it and it simplifies the presentation.

Within such a set of non-dominated heuristics there is a trade-off between generality and power. An increase in utility along any given utility dimension (e.g. one with a better chance of success) can only be achieved by using an heuristic which has lower utility on some other dimension (e.g. it is more expensive), or is applicable to a narrower range of problems, or both. What seems powerful or efficient in a general context becomes less remarkable in comparison with heuristics of similar range. A more general heuristic applicable over a wider range of problems generally has a lower utility for any given problem while more specific heuristics with higher utility are limited to a narrower range of problems.[9] This much seems borne out by everyday experience — there are no all-powerful problem solving strategies which are applicable in all situations or to all problems. If we take the range of an heuristic to be its domain, then its total utility — defined as the multiple integral of all the power curves of the heuristic — is limited and depends on the relative importance of (or more accurately the trade-off between) each of the utility axes in the context of the current problem/solution. The total utility of an heuristic, u_t is given by

$$u_t = \sum_{i=1}^{n} u_i w_i$$

where u_i is utility on the i-th utility measure and w_i is the weight or relative importance of that utility measure in the current context. Whether a gain or loss on any particular utility dimension represents an increase in total utility depends on its implications for other dimensions. For example, for a very good solution, cost may be less important.[10]

Lenat (1982) argues that in any given situation we should apply the most powerful heuristic available first and only resort to those with lower utility if the initial attempts to solve the problem fail. Of course it is unlikely we would know the power of an heuristic precisely in each possible situation. It is more likely that we would have some knowledge of the average power of each heuristic, and would use that as a guess of how useful each one would be in the current situation. If we assume that all heuristics are non-dominated, this corresponds to trying the most specific heuristic first, followed by the next most specific and so on.

More importantly, the trade-off between generality and power means that for any given set of heuristics, there exist problems which cannot be solved using those heuristics. That this is so can easily be demonstrated simply by selecting a problem which is sufficiently difficult relative to the domain of the heuristics. The existence

[9]In some circumstances the utility axis may have some absolute desirable point along it, e.g. some guarantee of correctness or efficiency. If an heuristic exceeds this value (even if only over a relatively narrow range of tasks) the way we view the heuristic may change; for example, we may term it 'algorithmic' or 'real time'. From this viewpoint algorithms are merely heuristics which have sufficiently high utility for guarantees to be made concerning their use, albeit in a restricted set of situations. conversely, one can try to apply an algorithm outside its domain of applicability, in which case the result may be useful and the algorithm is then used as an heuristic (Lenat, 1982).

[10]A belief held by many architectural students.

of such problems does not necessarily present any difficulties for an heuristic-based model of design, so long as most design problems can be solved using the available heuristics. This is after all what we would expect; heuristics are derived from experience with previously encountered problems. However, we shall argue that many if not most design problems cannot be solved using the available heuristics. This apparently paradoxical conclusion — that most design problems cannot be solved using existing design knowledge — is a consequence of the generality *vs.* power argument.

There are an infinite number of design problems associated with any given domain, differing on one or more of the situation dimensions, or the minimum utility required on any given utility dimension, or the relative importance the designer assigns to these utilities. We can imagine that a typical heuristic will be capable of solving some non-trivial percentage of the problems associated with its domain if it is to be worth remembering in the first place. However, while the set of heuristics associated with a domain may typically be successful in solving many of the problems occurring within that domain, each design decision must be made in the context of the design problem as a whole. The interaction between criteria which we have argued is a defining characteristic of design problems means that, in general, the effective domain of an heuristic is the design problem as a whole, and this can vary in many more ways than the putative domain of the heuristic. For example, the range of situation dimensions may be extended to include "problems in which *heuristic 101* has already been employed" or utility measures normally falling outwith the domain of the sub-problem such as "adequate utility on utility dimension u_x (in a problem in which *heuristic 101* has already been employed)". This does not imply that the total utility of each heuristic must somehow be increased. In many, *but not all* problems, the attainment of adequate utility on utility dimension u_x will be irrelevant to solving problem in the given domain. Rather there are many more ways the total utility of an heuristic might be distributed within the situation/utility space. Each unique distribution requires a unique heuristic.[11] An heuristic capable of producing the 'right' solution in all these situations would have to have high utility on all the relevant situation dimensions, but this is impossible because the total utility of an heuristic is limited. It has significant utility only because it is specialised to a small range of problems. If this is so, the question then arises: given that the total utility of an heuristic is limited, how many heuristics would be required for us to be sure of being able to assemble a consistent set which solves the problem?

The total utility associated with an heuristic may be distributed in a number of different ways. For example, some heuristics may be very good for a small range of problems while others are of lower utility but are appropriate in a wider range of situations. If we assume that an heuristic must have some non-trivial level of utility on some utility dimension to count as any sort of solution to a problem, then the number of heuristics required to solve the range of problems notionally occurring 'in' a given domain will typically be quite large. For example, if the total utility of non-dominated heuristics tends towards a constant, then for any given level of utility relative to some domain the number of heuristics required to completely cover the problem space will be a function of the power of the level of utility required: doubling the utility required along some axis increases the number of heuristics by a factor of two; doubling the utility required on all utility dimensions increases the number of heuristics required by a factor of 2^n etc.[12] Since there are many ways

[11]Note that this rules out the possibility of generating an infinite number of solutions by simply combining 'standard' heuristics associated with each domain.

[12]This assumes that all problem dimensions are equally important and that there is a uniform distribution of problems within the domain. However in reality it is likely that some will be more important than others and in some cases (e.g. problems of extreme difficulty) there may be no

in which the situation can vary (i.e. the set of typical problems is large) and the volume of an heuristic is limited, it seems unlikely that an appropriate heuristic will be available at all stages of the design process. Note that this argument applies at all levels of abstraction from layout to detail design if we consider the power of an heuristic relative to its domain.

The more inter-dependent the criteria are, the greater the number of heuristics required to solve any given problem. Even if we assume, as seems likely, that problems are not uniformly distributed along the utility and situation axes, the number of possible problems is still very large. We would therefore require a corresponding number of heuristics to be sure of solving a typical design problem selected at random from a domain. Leaving aside the problem of simply remembering this set of heuristics (for each domain) we are left with the difficulty of explaining how such a set could ever arise. If heuristics are derived from experience with a range of problems, knowledge of such a set would seem to imply experience of all possible problems. If, as seems likely, this is impossible, we are forced to conclude that we can never have enough rules to be sure of solving any given problem.[13]

If a suitable heuristic cannot be found, the criteria are deemed to be inconsistent, i.e. no solution exists within the solution space defined by the heuristics. This happens when no way can be found of achieving a particular requirement or set of requirements within the constraints imposed by the rest of the solution context. This may either be because while the available heuristics are capable of achieving their immediate goals, the resulting 'local' solutions have unacceptable consequences elsewhere (solutions can be found for each of the sub-problems considered in isolation, but these partial solutions are mutually inconsistent), or because no way can be found of achieving a particular goal even in the absence of other constraints (the problem lies outwith the scope of any known heuristic). Such conflicts are common in design, indeed we have identified the existence of conflicts between criteria as one of the characteristics of design problems (Logan & Smithers, 1989). In such a situation the designer must either modify the problem, i.e., relax one or more constraints until the current solution meets the revised design requirements, or modify the means available to solve the problem by modifying an existing heuristic or creating a new heuristic. (In many situations, it will be necessary to modify both the requirements description and the available problem solving strategies to achieve a consistent solution.) Often when we start solving a problem, the criteria are inconsistent. When we are finished, the solution and criteria are consistent either because we relaxed the criteria or because we found a new way to solve the problem which satisfies the goals. Problems which were perceived as inconsistent or contradictory cease to seem so.

6 Heuristic Formation and Learning in Design

The results outlined in the previous section, i.e. the need for domain specific heuristics and the existence of problems which cannot be solved using a given set of heuristics, are both consequences of the generality *vs.* power argument.[14] Yet problems are solved and there is considerable evidence from cognitive psychology and design research that designers do use some form of prestructures or relation-

heuristics available.

[13]Even if such a set of heuristics existed, the problem of finding the 'right' heuristic to solve the current sub-problem remains. As we saw in section 5.2, while heuristics are evaluated on their global utility, they are of necessity selected on their local utility as part of the design context is unavailable when the decision to use the heuristic is taken.

[14]Note that this conclusion is not dependent on the details of the appropriateness vs. situation curves, but rests on the much more general notion that any approach to problem solving is restricted to some (limited) range of problems.

ships to structure and solve their problems. In an attempt to explain this paradox we present an alternative view of the development and use of prestructures in design which accommodates these observations and theoretical results, and illuminates the critical role of the learning process in design.

Design problems by definition are unique. We have argued that it is unlikely, given the large number of criteria which form the definition of a design problem, that any given problem will be identical to a previously solved problem. However in general the designer will have solved similar problems in the past and will use this experience as a guide in solving the current problem. We hypothesise that there exists a set of 'core' heuristics associated with each domain which have shown themselves to be useful in the past and worth remembering, and that these core heuristics are adapted (to a greater or lesser extent) to the problem at hand. For each problem domain, such as layout design, structural design, materials selection, detailing etc. whatever its level of abstraction, there is a collection of rules of thumb, typical examples and relationships which provide a more or less powerful heuristic core for that domain, and which are adapted to the details of any given problem.

The set of heuristics is constantly being adapted to new problems and situations, and the core heuristics are continuously refined through experience gained in new situations. For example, the strategies and relationships commonly used in the design of a structural system will be different from those used in determining the arrangement of elements in a particular system, but in each case the rules must be modified to accommodate the details of the particular problems to be solved.[15] Something of this sort seems to be required to explain the use of previous experience in solving entirely new problems (as opposed to simply copying old solutions), and indeed how design or problem solving is possible at all (Petrie, 1979). Whether it is more useful to view the generation of new heuristics as the definition of a new (sub)domain or the adaptation of an existing heuristic is an interesting question. The latter course has been adopted here to highlight the distinction between operations over domains (reasoning by analogy) and operations within domains (adaption of existing heuristics). We have argued elsewhere (Logan, 1987) both of these operations can be considered forms of reasoning by analogy carried out at different levels.

At its simplest the core can be seen as simply representing all that a designer knows about a particular class of problems. More generally the core represents some compromise between the number of heuristics required to adequately cover a domain (and the associated overhead in remembering and searching for any particular heuristic), and the effort involved in adapting a small number of general rules to a wide variety of situations (or reinventing the wheel). We further hypothesise that such specialisations, or more generally 'adaptions' of existing heuristics, are necessary in all goal directed heuristic systems, and that the development of these problem specific heuristics is the motivation for analysis through synthesis, and underlies the process of understanding the structure of a design problem identified above as central to the design process.[16]

The definition of a design problem can be viewed as a point in $n \times m$ dimensional space — that is, as a specification for an heuristic which will achieve a given level of 'performance' in a given situation, (typically an heuristic which has a reasonable chance of providing the 'right' answer to the current problem). The process of adapting an existing heuristic can be viewed graphically as moving its characteristic

[15]Note that these 'core heuristics' do not necessarily have to have shown themselves useful in a wide range of situations; there exist some specialised but very powerful heuristics which are useful in recurring problems.

[16]The formation of new heuristics is also closely related to the formation and use of analogies in the process of 'reasoning by analogy'. This is discussed in more detail in the next section.

function left or right to peak in a different situation.[17] This process of adaption can be seen as a natural extension of the 'central assumption of the theory of heuristics' presented in the previous section: i.e., if an heuristic H was (or would have been) useful in situation S then it is likely that heuristics similar to H will be useful in situations similar to S.

Several authors (Hillier & Leaman, 1974; Steadman, 1979) have argued that there exist cultural stereotypes or 'templates' which form the basis of design solutions in a way which could be considered analogous to the core heuristics discussed above. Hillier and Leaman (1974; Hillier & Leaman, 1976) have characterised this process as one of the elaboration and modification of cultural stereotypes or 'templates'. They argue that that the designer is situated in "a richly connected universe whose connections are those dissimilar domains that must be related in design; activity and space, psychology and climate and so on". These structures are embedded in the language the designer uses and in the instrumental set—the technologies or kits of parts and typical design solutions to which his systems of representation refer (Hillier & Leaman, 1974). Even to name an architectural problem—say, 'design a school'—implies a whole range of solutions which will be more or less immediately activated by the designer's prestructures. These structures form an evolving typology of standard solutions to recurring problems in design, modified by the designer's experience, ideology and the physical, social, and cultural environments which form the context of design. In this view design is seen as the process of discovering "the appropriate transformation or 'unfolding' of prestructures in relation to the constraints imposed by the environment of the problem" (Hillier & Leaman, 1974). Both the transmission and transformation of prestructures form a process of elaboration and discovery which underlies the active formation of relationships and within which every solution may be unique.

Similarly Schon (1988) argues that designers make use of 'design rules' to "reason their way to moves, draw out consequences of possible moves [and] make and evaluate design decisions". Rules are derived from types. Types function as leading ideas to generate sequences of design experiments including "chains of reasoning, consideration of possible moves, detection of consequences and implications and choices". They guide the selection of rules, provide the information necessary for their application and provide the basis for challenging and correcting them. Rules are seen as contingent and contextual; they are held tentatively and are subject to exceptions and critical modification. However Schon argues that while designers do share rules, different designers also use different rules. While some rules may be common to many designers (and may determine the form individual development can take), designers develop many individual strategies for solving problems as a result of their education, professional experience etc.

In all these views, design is seen as the modification or refinement of the designers' general codes and relationships within the context of the current problem. More generally we can see the problem of design as one of learning how to develop these basic prestructures to solve a particular problem. Indeed, as has already been noted, the development of such an understanding of the structure of the problem is the objective of the process we have termed 'analysis-through-synthesis'

7 Induction and Rule Formation

In this section we attempt to develop a model of the learning process described above within the framework of the model of design outlined in previous sections. In

[17]This can be reinterpreted as finding a relationship which will link some solution space to a given criterion space; or in object level terms, finding a relationship which structures the concepts comprising the 'problem domain'.

particular we attempt to clarify the role of induction in the formation of relationships between criteria and its relationship to the process of theory formation. In doing so we shall try to show that in closing the cycle of abduction, deduction and induction, the model proposed by March (1976) can be viewed as a simple model of the design process (or more precisely the process of analysis-through-synthesis), and hence of the development of the designer's prestructures or solution strategies.

March's model is based on the work of the American philosopher C. S. Peirce. Peirce takes the Aristotelian syllogism:

$$x \text{ is } y;\ y \text{ is } z; \text{ hence } x \text{ is } z$$

as typifying deductive or analytic reasoning; the application of a general rule (y is z) to a particular case (x is y) to give a logically determined result (x is z). The deductive syllogism is based on the concept of necessary truth; that is that the facts presented in the premises could not under any imaginable circumstances be true without involving the truth of the conclusions. More simply, deduction can be considered the inference of a result from a case and a rule. However Peirce argues that inductive or synthetic reasoning, being something more than the application of a general rule to a particular case, can never be reduced to this form and he goes on to develop two further modes of reasoning which he terms *abduction* and *induction.* Neither of these forms of inference are logically determinate. Abduction reflects the reasoner's presumption that a certain phenomenon exists to account for his observations, given that a particular theory holds. Abduction is the inference of a case from a rule and a result. Induction mirrors the reasoner's search for a law to account for the regularities among phenomena, and is responsible for engendering new habits of thought. Induction is the inference of a rule from a case and a result.

March relates the three forms of reasoning proposed by Peirce to the context of design in terms of their results:

1. The creation of a case or 'composition' which is accomplished by abductive reasoning;

2. The prediction of results or a 'decomposition' comprising the characteristics of the design which emerge from an analysis of the whole composition, accomplished by deduction; and

3. The derivation of rules or 'suppositions', an idea, a theory, or in the modern usage a model, a type, accomplished by induction.

March argues that the designer uses his previous experience and knowledge of solution types in an attempt to produce a solution which satisfies the problem criteria. Such a speculative design cannot be determined logically because the mode of reasoning involved is essentially abductive. It can only be inferred conditionally from our state of knowledge and the available evidence. Deductive inference is then be used to predict measures of expected performance by the application of further models and theories to the particular design proposal. As the design proceeds new relationships and criteria are added which may critically augment the original set in the previous abductive phase. In the inductive stage the design and its predicted characteristics are used to infer new generalisations and suppositions. General rules are refined in the context of the current design solution as induction criticises the original hypothesis from the abductive phase and provides more discriminating tools for the next round of the cycle. In doing so it evaluates. "In itself a design, or rather the set of pertinent characteristics by which it is perceived, has no value. It assumes relative value through comparison with other designs both existing and entertained, as well as with the environment as a whole. Indeed evaluation assumes that suppositions about worth, preference and desirability can be inferred. It is these suppositions that form the basis of the abductive phase of designing. That is to say the models required to produce design alternatives are value laden" (March, 1976). March characterises this as an iterative process in which there are constant refinements and redefinitions of characteristics, design, and suppositions as the solution evolves. The model is envisaged as "representing a critical learning process, in that statements inferred at later stages may be used to modify those used in earlier stages, and thus to stimulate other paths of exploration" (March, 1976).

However given the large number of criteria which form the problem definition it is unlikely that the current problem will be exactly similar in all respects to any of the previously solved problems which form the basis of the designer's prestructures. Indeed it can be argued that rules are inherently fuzzy in defining a relation between two sets of concepts at a higher level of abstraction than that of either the individual cases or results subsumed by the rule. It therefore seems unlikely that at the level of its application, (as opposed to its level of definition), any rule will be a perfect 'fit' for a given set of criteria, as the concepts involved in rule definition can be seen as labels of fuzzy sets defined by a membership function.[18] The means to solve the problem will not exist and must be created. We will now consider this process in more detail.

7.1 Learning and Analogy

In a sense all reasoning can be seen as what might be loosely termed reasoning by analogy or reasoning from similar cases.[19] In reality both situations and the relationships between them are inherently fuzzy. Neither can be completely characterised and our everyday inferences must rely on the various relations of similarity between the current situation and the previous experience we are using as a justification for our conclusions. We continually adapt existing rules from the domain of interest (or indeed any other domains which are in some way considered relevant) to the current situation. It is this adaption which we shall argue can be viewed as a form of learning.

The designer begins by assuming that a rule which worked in a similar situation, or a rule similar to it, will work in the current situation. (This is in effect a revised version of the 'central assumption' underlying Lenat's 'second order theory of heuristics': that "in a complex, knowledge-rich, incompletely-understood world, it is frequently useful to behave as if *appropriateness(action, situation)* is continuous and time-invariant"). Of course this approach is unlikely to give a completely satisfactory answer, but it provides a starting point for the exploration of the implications of a particular solution in terms of the relationships between criteria.[20] Subsequent modifications bring the relationship closer to the current problem. No relationship can be characterised in absolute terms as 'core' or 'problem specific'. Both generic solutions and solutions to previously solved problems will, in general, have to be adapted to the current problem context. Rather the process is recursive in involving a series of successive approximations to the required relationship.[21] From this viewpoint learning by discovery can be seen as the refinement of such an

[18]It also implies that the alternative solutions resulting from the application of a rule will satisfy the wider context of the rule criterion to differing degrees, and evaluation of a case within this context can be interpreted as the redefinition of the membership function of a fuzzy set of solutions in the context of a particular set of problem requirements.

[19]This of course includes the three forms of inference identified by March. The perfect match required between a rule and a case or a case and a result can be seen as special cases of a more general fuzzy matching procedure.

[20]This is the motivation underlying much of the process of analysis-through-synthesis identified in section 3. The successive refinement of rules can be seen as a further corollary of the designer's inability to consider all the aspects of a design problem simultaneously.

[21]Petrie (1979) has argued that the use of analogy is basic to all learning, maintaining that it is epistemologically necessary in relating the unknown to the known and familiar.

analogy; the discovery of the errors and omissions in the initial model and to what extent the structure of the new domain does in fact resemble the structure of the domain which provided the initial analogy.

To simplify the exposition we shall assume that some appropriate rule has been found or generated, i.e., we shall start with the assumption that a rule has been found which matches the current result (abduction) or case (deduction), and we shall restrict our attention to the adaption or refinement of a rule to a given situation. More general cases await a more detailed treatment of analogy. Learning within a domain can be viewed as a particular case of adaption or refinement of analogies or guesses where there already exists a rich collection of similar relationships, which differ only slightly from the current situation (e.g. which are valid over slightly differing ranges of problem criteria) to serve as a basis for our conjectures.[22]

7.2 Maintaining Consistency in the Problem Model

Implicit in this view of learning as successive refinement is the idea of some kind of independent check on the success or otherwise of a proposed rule. We require some non-tautological way of assessing how well a case, (and hence a rule) achieves a goal, or conversely how well a rule predicts a (known) result from a given case. If there is only a single relationship linking two sets of concepts we have no independent means of assessing the result of applying the rule. In any particular situation cases and results derived using the rule are, in a sense correct by definition. More generally such a relationship is difficult to modify or adapt to a particular situation as no means exists of determining whether a series of modifications is converging on the desired relationship. In practice rules usually form part of an interdependent system of relationships which together comprise the body of knowledge associated with a domain. Different rules express different conceptualisations of the relationships within a domain. Some relationships will be specialised for generating hypotheses, while others will be intended to be used analytically. Each individual rule will have a particular set of attributes which determine its range, reliability, accuracy, speed etc. and will be linked to other rules with different attributes (more accurate but slower etc.).

Thus the performance of any one relationship can be determined using some other relationship from the domain which is in some sense considered more reliable or appropriate in the current situation. For example, existing analytical (deductive) models can be used to check the performance of an heuristic by inferring the consequences of a proposed solution hypothesis. Similarly initial assessments based on simple rules of thumb can be checked using more detailed (and more accurate) relationships as more data becomes available. Conversely existing abductive models (in the form of examples or case-result pairs) can be used to assess the performance of a proposed (deductive) simulation model, by comparing the predicted result with the previously recorded values. In general the utility or performance of a rule, and our confidence in it (or more precisely in the results of using the rule in a particular situation), will be a complex function of the purpose of the rule, previous experience in the domain, the data available and the time available for its use etc.

In what follows, we will model heuristics as conditionals of the form

$$A_1, \ldots, A_n \leftarrow B_1, \ldots, B_m$$

where $A_1, \ldots, A_n$ are the condition(s) and $B_1, \ldots, B_m$ are the action(s). This approach has the advantage of preserving the monotonicity of logical inference and

[22]Although we shall only consider adaption within a given domain here we have argued elsewhere (Logan, 1987) that the techniques presented below are equally relevant to the production of analogies, and that the refinement of techniques or solution strategies (what might be termed 'between designs' learning), also proceeds through trial and error guided by heuristics.

allows us to work within a standard logical framework (see (Logan, 1987) for details). We will distinguish between deductive rules of the form

$$A(f(x_1,\ldots,x_n)) \leftarrow B_1(x_1),\ldots,B_n(x_n)$$

and abductive rules[23]

$$A(x) \leftarrow B_1(f_1(x)),\ldots,B_n(f_n(x))$$

according to the relations of functional dependency between the criteria (Logan, 1989).[24] Rules with no function terms, e.g. $A(x) \leftarrow B(x)$, are both abductive and deductive in the sense defined above. Note that both sets of rules can be used abductively and deductively. Given a rule $A(x) \leftarrow B(f(x))$ and an hypothesis $B(f(a))$ we can derive $A(a)$ as required by the definition of abduction. Rather this distinction is more a question of how the relationships are conceptualised and how they are typically used, rather than constituting a basic addition to the framework outlined by March.

Typically each set of rules will only be consistent for a limited range of values, and in general there is no guarantee that any given subset will be consistent for any range of values. For example, the rules

$$A(g(x)) \leftarrow B(x)$$

and

$$A(x) \leftarrow B(f_1(x))$$

may be consistent for certain sets of values $m < x < n$ but not for others $x < m, x > n$ where a different rule such as $A(x) \leftarrow B(f_2(x))$ may be required to maintain consistency. This context dependency can be expressed explicitly in the form of 'context switches'; literals which have to be true for the rule to be used. For example, the rule

$$A(x) \leftarrow B(f_1(x)) \wedge x < n \wedge x > m$$

will only be used if x is in the range $m - n$. (If there are no assertions regarding the value of x within the the current problem model, a constraint limiting it to this range will form part of the hypothesis).

7.3 An Example: Generating A Valid Hypothesis

The problem of, for example, generating a valid hypothesis can be reformulated within this framework as that of finding an abductive rule which is consistent over the range of the goal criterion relative to the subset of relevant (deductive or abductive) rules which together define the problem dependent subtheory for the domain. For example, in attempting to achieve some criterion $A(a)$ a designer may use a relationship such as

$$A(x) \leftarrow B(f(x)) \tag{1}$$

[23]We have not adopted the conventional practice of denoting an 'heuristic' (in the sense of an abductive relationship) as an inverse conditional: i.e. a relationship of the form $A(x) \rightarrow B_1(f_1(x)),\ldots,B_n(f_n(x))$. Within a conventional first-order framework the meaning of a statement such as, for example, $Area(x,u) \rightarrow Length(x,f(u)), Width(x,f(u))$ is that if the area of x is u then it is *logically necessary* that it's length and width be $f(u)$. This means we can't represent alternative ways of deriving the length and width of a room unless we are willing to specify mutually exclusive sets of situations in which they are appropriate. To do otherwise entails either reinterpreting the semantics of the conditional or the introduction of modal operators.

[24]More generally rules of both types can be either heuristic (in that the results of their use cannot be guaranteed correct in all situations) or algorithmic.

representing a rule of thumb, previous example etc. to generate an hypothesis, or case

$$B(f(a)) \tag{2}$$

This is essentially the process of abduction; the reduction of a set of abstract constraints (expressed as a series of atomic propositions) to some set of simpler constraints by using the relationships embodied in the composite propositions as production rules. The problem state containing the criterion $A(a)$ is transformed into a more detailed one containing the additional assertion $B(f(a))$. The resulting expanded set of criteria, if achieved, would result in the attainment of the constraint denoted by the higher level concept. In conceptualising the problem in terms of a relationship between these two criteria the designer has already determined the overall form of the solution.[25] Any specific design solution is defined by the assignment of a particular value to the concept $B(x)$.

Once specified, each new partial solution is checked against the criteria or constraints used in generating all previous partial solutions. In particular it is checked against the relationships forming the subtheory for the current domain. For example, a more accurate rule with higher reliability but which is difficult to invert can be used to determine the success of the proposed hypothesis in achieving the desired criterion. For example, from the rule

$$A(g(x)) \leftarrow B(x) \tag{3}$$

and the hypothesis

$$B(f(a)) \tag{2}$$

we can derive the result

$$A(g(f(a))) \tag{4}$$

This is essentially the process of deduction. In the context of the model developed above deductive inference can be viewed as the determination of the implications or emergent properties of the current problem description based on the functional dependencies between concepts. As the solution develops new higher level propositions become derivable from the statements defining the current problem. Typically this entails the prediction of one or more of the performances or characteristics of the design from some subset of its attributes.

In many cases the hypothesis may not achieve the required criterion value i.e. $A(a) \neq A(g(f(a)))$ or $g \neq f^{-1}$ at a, or it may fail to achieve some one of the other criteria such as $C(b)$, i.e.

$$B(f(a)),\ C(h(x)) \leftarrow B(x)\ \vdash\ C(h(f(a)))$$

and

$$C(h(f(a))) \neq C(b)$$

or both. More generally an hypothesis fails to attain the desired criterion value if the result predicted by the deductive rule $g(f(a))$ is outwith the range required for attainment of the goal i.e. $g(f(a)) < a_1$ or $g(f(a)) > a_2$, where $a_1 - a_2$ denotes the range of criterion values which would result in the attainment of the goal. That is, the values of the goal and objective together with some integrity constraint imply an inconsistency. In formal terms we can represent this as

$$A(a),\ A(g(f(a)))$$

together with

$$\neg(A(x) \wedge A(y) \wedge \mathit{Diff}(x,y))$$

[25]We shall ignore the problem of finding or generating this relationship and limit our consideration to the variation in the value of the criterion $f(a)$.

implies an inconsistency. In a sense we can regard the appropriateness of the abductive rule as inversely proportional to the 'distance' between the goal and the deductively derived result.[26] This hypothetico-deductive cycle underlies the analysis-through-synthesis design methodology outlined in section 3, in which solutions are proposed to discover their implications in terms of other criteria. In a sense abduction could be said to subsume deduction in that the role of deductive inference can be seen as simply determining the implications of proposals with respect to constraints, guiding the transition between states and ensuring their consistency.

In reality, of course, the process is rather more complicated. Whether an hypothesis attains a given criterion value is intrinsically context dependent in being at least potentially dependent on the values of all the other criteria forming the problem model. The attainment of a goal by any given criterion value is dependent on (amongst other things) what is possible in terms of the criterion in the current context, the sensitivity of the criterion value to minor design changes and the implications of such changes for other criteria. The designer dynamically redefines the satisficing level for each criterion, and hence the performance level which must be attained by any hypothesis, in response to the current problem context. Lower criterion values may be accepted if the original goals are found to be unattainable in the context of the current solution, or if their attainment would have unacceptable consequences for other criteria.[27] While the simple view of deduction and goal attainment presented above is incapable of modelling this process of dynamic goal redefinition (and many other things), we believe it is an adequate model of a failure to achieve a goal, and hence can give some insight into the process of rule formation.

As the design develops the designer learns more about the problem and the solution as new aspects of the problem become apparent, and the conflicts inherent in his view of the problem are revealed. The designer uses this increased understanding to generate new structures and relationships. The rules or problem transformations which form the basis of abduction and deduction are continually refined and modified as the design progresses. The design and its derived criteria are used to infer new generalisations and relationships better adapted to the current problem context (for example, by modifying the criterion values or introducing constraints requiring that if a particular solution is to be used in the current context other design criteria must take certain values). This integration is the step Peirce termed 'induction'.

Thus from a case

$$B(f(a)) \tag{2}$$

and a result

$$A(g(f(a))) \tag{4}$$

derived within the context of the current problem we can infer a new rule, say

$$A(x) \leftarrow B(f'(x)) \tag{5}$$

which subsumes both Eqn.(2) and Eqn.(4) and hopefully is better adapted to producing hypotheses within the range of the criterion value $A(a)$. In general there will be many such rules and it will be necessary to select between them on the basis of some criterion (simplicity, plausibility etc.). However we will not consider these problems further here. Like abduction, induction is not logically determinate. A newly inferred relationship cannot be guaranteed to be consistent with either the derived criteria or the set of hypotheses forming the current problem state. A derived relation is said to be valid if it is consistent with the set of constraints which

[26]Indeed we can view the utility of a rule in attaining some goal criterion in these terms. The function *appropriateness*(*action*, *situation*) becomes the inverse of the objective — the result of taking some action in some situation expressed in terms of problem criteria rather than meta-level properties. However we shall not pursue these ideas further here.

[27]These problems are discussed in more detail in (Logan, 1987).

constitute the current problem model, and irredundant if it allows the derivation of new consequences and/or hypotheses. As such it subsumes both deduction and abduction as necessary to the determination of its consistency.

The cycle then begins again. This new rule can be used to generate a new hypothesis which forms the basis of a more detailed exploration of the problem, leading to the derivation of new criterion values and the generation of further hypotheses. Hopefully the deductive and abductive models will tend to 'converge' as the structure of the problem is understood. In the simple example above this happens when some $f^* = g^{-1}$ over the range of the goal criterion, i.e. $g(f^*(a)) = a$. If no modification of Eqn.(1) leads to success then either the problem is overconstrained, (i.e. a basic conflict exists within the criteria forming the current problem context), and the solution must be modified or the constraints relaxed, or the original (meta)-hypothesis of using Eqn.(1) was flawed (e.g. an inappropriate analogy), and the (meta)-rule that lead to the generation of Eqn.(1) must be modified accordingly. March (1976) has argued that the phases follow one another in the iterative sequence abduction- deduction-induction with constant refinements and redefinitions of characteristics, design, and relations as the solution evolves. However while this is the general direction of the argument there is no logical necessity for any particular operation to follow any other; it is common, for example, to defer evaluation until several decisions have been made or to consider several hypotheses simultaneously, and in general a simple iterative process is inadequate to represent the complexity of design.

This cycle of abduction-deduction-induction is a learning process. In this view 'learning' is seen as the formalisation (through induction) of the understanding of the relationship between an action (or more generally a situation) and its consequences. In design this takes the form of learning about critical relationships and possible forms as the solution evolves. Between generic solutions planning is less a search for the best solution than an exploration of the compromises that give sufficient solutions. These explorations help the designer appreciate which requirements may be most readily achieved. Learning more is the most important part of this process, and redefinition of the problem and solution can only result if more knowledge about them is acquired. The 'right' abductive relationship is one which not only achieves the original (local) goal, but which achieves this goal in the context of the current problem. The resulting abductive function f^* represents the designer's understanding of the structure of the problem; it embodies the knowledge of how to achieve the goal in the context of the problem, i.e. the modifications which must be made to the standard solution for it to work in the context of the current problem.

8 Conclusion

In this paper we have tried to relate two of the main approaches to design found in the design theory literature — that design is a knowledge-based process and that design is a learning process — in an attempt to explain why design happens the way it does. Starting with a small number of (hopefully) plausible assumptions about the nature of design problems and the limitations of design knowledge, we have developed a theoretical model which tries to account for some of the results from empirical studies of the design process. In particular, we have tried to explain why even in the case of reasonably simple problems designers find it necessary to refine and extend their knowledge in the context of the current problem and how in doing so they come to understand the pattern of relationships between criteria which together define the structure of the problem. By embedding our model of design knowledge within the framework of the three logical operators proposed by March we have tried to show how the development of design knowledge might proceed.

The refinement of an heuristic through a process of successive approximation can be viewed as a very crude model of, (or perhaps a metaphor for), what happens when a designer begins to 'understand' the structure of a problem, in terms of the relationships between the problem criteria. The resulting hypothetico-deductive process can be seen as a natural consequence of the need to refine and adapt a set of basic or core relationships to a given problem, and forms the context of a learning process through which designers refine their knowledge both within and between design problems.

The use of heuristics to model design knowledge is not meant to imply any ontological commitment, e.g. that the rules used by designers can be made explicit or even that such rules exist. Rather heuristics provide a useful framework for exploring certain assumptions about design knowledge; any equivalent representation such as 'prestructures' or 'prototypes' would do equally well. Indeed, we would argue that the analysis presented above applies to any knowledge-based view of design, irrespective of whether design knowledge is represented as heuristics, problem transformations or prototypes. Nor do we claim that March's abduction-deduction-induction model of the design process has any psychological validity. Rather the intention is to place some broad constraints on how such a process of rule formation might work. However we would argue that even such a limited model is useful in gaining some insight into the role of learning in design and that it can provide a useful framework for future work.

Acknowledgements

This paper is largely based on work carried out while the author was a student at the Department of Architecture, University of Strathclyde, and reported in (Logan, 1987), and is currently supported under SERC grant No. GR/F/6200.1. I am grateful to the staff and students of the ABACUS CAAD group within the Department of Architecture for numerous discussions during the development of these ideas. I am also indebted to the staff and students at the AI in Design research programme in the Department of Artificial Intelligence at the University of Edinburgh for many helpful discussions of these ideas. Dave Corne, Tim Smithers and Nils Tomes read an earlier version of the paper and provided many helpful comments.

References

Akin, O. (1978). "How do Architects Design", in *Artificial Intelligence and Pattern Recognition in Computer Aided Design*, J. C. Latombe, ed., North Holland, 65–98.

Bazjanac, V. (1974). "Architectural Design Theory: Models of the Design Process" , in *Basic Questions of Design Theory*, W. R. Spillers, ed., North Holland, Amsterdam, 2–19.

Davis, P. J. & Hersh, R. (1980). *The Mathematical Experience*, Birkhauser.

Foz, A. T. K. (1972). "Some Observations on Designer Behaviour in the Parti", MIT Press, MA Thesis, Cambridge, Mass.

Gero, J. S. (1987). "Prototypes: a new schema for knowledge-based design", Architectural Computing Unit, Department of Architectural Science, University of Sydney, Working Paper.

Gero, J. S., Maher, M. L. & Zhang, W. (1988). "Chunking structural design knowledge as prototypes", in *Artificial Intelligence in Engineering: Design*, J. S. Gero, ed., Elsevier, Amsterdam, 3–21.

Hillier, W. & Leaman, A. (1974). "How is design possible", *Journal of Architectural Research* 3, 4–11.

Hillier, W. & Leaman, A. (1976). "Architecture as a discipline", *Journal of Architectural Research* 5, 28–32.

Jones, J. C. (1970). *Design Methods: Seeds of human futures*, Wiley.

Krauss, R. I. & Meyer, J. R. (1970). "Design: A Case History", in *Emerging Methods in Environmental Design and Planning*, G. T. Moore, ed., The MIT Press, 11–20.

Lawson, B. (1980). *How Designers Think*, Architectural Press, London.

Lenat, D. B. (1979). "On Automated Scientific Theory Formation: A Case Study Using the AM Program", in *Machine Intelligence 9*, J. Hayes, D. Michie & L. I. Mikulich, eds., Halstead, New York, 251–283.

Lenat, D. B. (1982). "The Nature of Heuristics", *Artificial Intelligence* 19, 189–249.

Lenat, D. B. (1983aa). "Theory Formation by Heuristic Search", *Artificial Intelligence* 21, 31–59.

Lenat, D. B. (1983bb). "EURISKO: A program that learns new heuristics and domain concepts", *Artificial Intelligence* 21, 61–98.

Logan, B. S. (1987). "The Structure of Design Problems", Department of Architecture, University of Strathclyde, PhD Thesis, (unpublished).

Logan, B. S. (1989). "Conceptualizing Design Knowledge", *Design Studies* 10, 188–195.

Logan, B. S. & Smithers, T. (1989). "The Role of Prototypes in Creative Design", in *Preprints of the International Round-Table Conference: Modelling Creativity and Knowledge Based Creative Design.*, Department of Architectural and Design Science, University of Sydney, Sydney, 233–248.

March, L. (1976). "The Logic of Design and the Question of Value", in *The Architecture of Form*, L. March, ed., Cambridge University Press.

Newell, A., Shaw, J. C. & Simon, H. A. (1963). "GPS, A Program that Simulates Human Thought", in *Computers and Thought*, E. A. Feigenbaum & J. Feldman, eds., McGraw Hill, New York, 279–296.

Oxman, R. & Gero, J. S. (1988). "Designing by prototype refinement in architecture", in *Artificial Intelligence in Engineering: Design*, J. S. Gero, ed., Elsevier, Amsterdam, 395–412.

Petrie, H. G. (1979). "Metaphor and Learning", in *Metaphor and Thought*, A. Ortony, ed., Cambridge University Press, Cambridge, 438–461.

Polya, G. (1945). *How to Solve It*, Princeton University Press.

Pushkin, V. N., ed. (1972). *Problems of Heuristics*, Keter, Jerusalem.

Schon, D. A. (1988). "Designing: Rules,types and worlds", *Design Studies* 9, 181–190.

Simon, H. A. (1970). *The Sciences of the Artificial*, MIT Press.

Simon, H. A. (1973). "The Structure of Ill Structured Problems", *Artificial Intelligence* 4, 181–201.

Steadman, P. (1979). "The History and Science of the Artificial", *Design Studies* 1, 49–58.