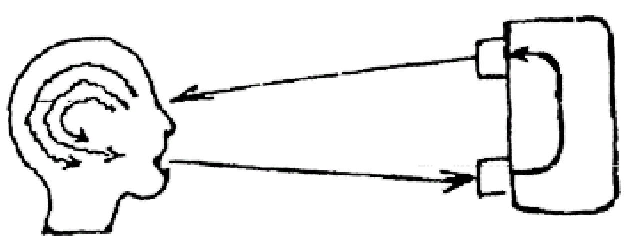
Chapter 2: The Process

The Creative Process

In the previous chapter, the role of computers in design was examined from a historical and technological perspective. In the foregoing sections, the role of the computer in the creative process is discussed. The main direction here is to investigate whether computers can be regarded as candidates for sharing or participating in a task that has been attached almost exclusively to humans: that of creativity.

The Machine as a Reflection of the Human Mind

According to Negroponte, "computers are intellectual machines that allow us to simulate human behavior." [Negroponte, 1970: 1] In developing computer programs one is forced to question how people think and how designs evolve. In other words, computers must be acknowledged not only as machines for imitating what is understood, but also as vehicles for exploring what is not understood. The entire sequence of specifying computer operations is similar to that of human thinking. When designing software for natural language understanding, knowledge representation, inference, or learning, one is actually transferring to a machine, processes of human thinking. The computer becomes a mirror of the human mind, and as such, it reflects its thinking. In that sense, if computers are encoded with the basic principles of human logic, taught ways to acquire their own knowledge, and be given ways to import and export information from the real world, then they can probably behave in ways similar, if not more advanced, to that of humans.



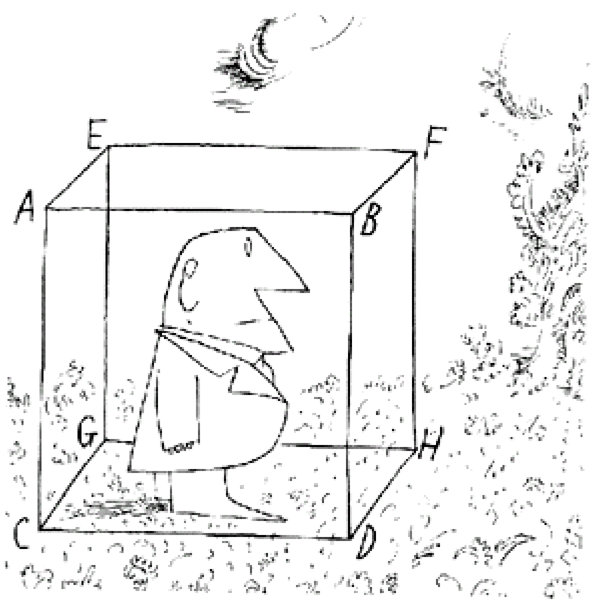
A computer can become a reflection of human thinking.

In addition, design can be explored as mental process not only by observing human behavior, but also by observing the machine's behavior. To do this, it is necessary to perform individual operations with substantial independence. The entire sequence of operations must be such that there is no human intervention from the time data is entered until the results are obtained and that design decision-making mechanisms be built into the machine itself. This does not mean that a "computer-designer" is to be created even though that may be desirable eventually. Rather, it suggests the attainment of independence in solving particular design problems. Thus, the designer can observe via the computer his decision-making process and compare it with that of others.

Limitations of the Machine

According to Sapir and Whorf, "a world view of a culture is limited by the structure of the language which that culture uses." Maybe this applies also for the world of computers. To understand the way computers work one must know and understand their technical structure and the logic of their language. In the same way that a poet or an author is bounded by language, context, or culture, a "computer-designer" may also be bounded by issues of technology, language, or algorithmic complexity.

At present machines have denatured language-codes. To employ computers more effectively in the creative process, databases must extend beyond mere geometric and non-geometric codicil information. They must also include the meaning of that information. For example, in graphics design, software should allow recognition of a square regardless of its size and orientation. Or an object, such as a statue, should be understood both as a form and as for what it represents. Today, one can begin to see systems that allow one to perform a large number of calculations and which are moving towards larger databases that permit human interpretation. In the future, databases may include information on the meaning of objects, events, and relations thereof. If and when that happen, computer can move closer to the creative process.



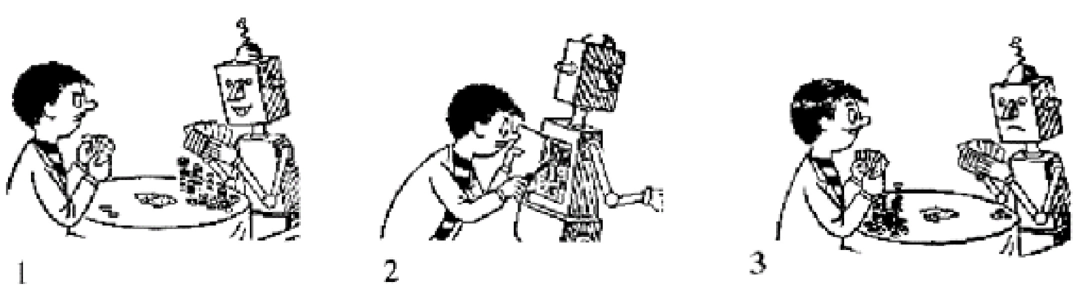
Machines, like humans, are bounded by their own limitations

Games and Design

According to Norbert Wienner, computers are able to play games and, for that matter, are able to follow rules, tactics, and strategies. "It is not very difficult to make machines that will play chess of a sort. The mere obedience to the laws of the game, so that only legal moves are made, is easy within the power of quite simple computing machines." [Wiener, 1947: 171]

Games employ rules, strategies, tactics, and goal seeking that may be useful beyond the game's boundaries, e.g., in design. However, it is unclear whether it is possible to construct a game-playing system that can use design rules and whether this capability would represent an essential difference between the potentialities of the machine and those of the mind.

There is a theory of games (Von Neumann, 1945) which establishes a way to describe and analyze them and their strategies by working from the end of a game rather from its beginning. In the last move of the game, a player strives to make a winning move if possible, or, if not, at least a drawing move. When the entire strategy is known, this is manifestly the best strategy for playing the game. However, in design, available knowledge is not sufficient to permit the formulation of a complete strategy of this sort. In design, strategies can only be approximated. Such a strategy would rely on the relation of local actions to global intents, on attitudes not always justified. For example, the local moves embodied in construction procedures are characterized by specificity. In contrast, global goals can be quite ambiguous.



Games employ rules, strategies, tactics, and goal seeking.

Given that the global intentions of design are, at best, ambiguous it is not necessary to question whether it is possible to construct a game-playing machine which will generate an optimum solution by following rules in Von Neumann's sense. On the contrary, it is unquestionably possible to construct systems that undertake local actions irrespective of the final goal. The real problem is to construct a computer system, which will offer opportunities for interesting and challenging dialogue with a designer. Such a system would be capable of learning by experience, which will enable it to improve its knowledge of the strategy and rules of design. The point of departure for this learning would be a set of rules derived statistically from design precedent.

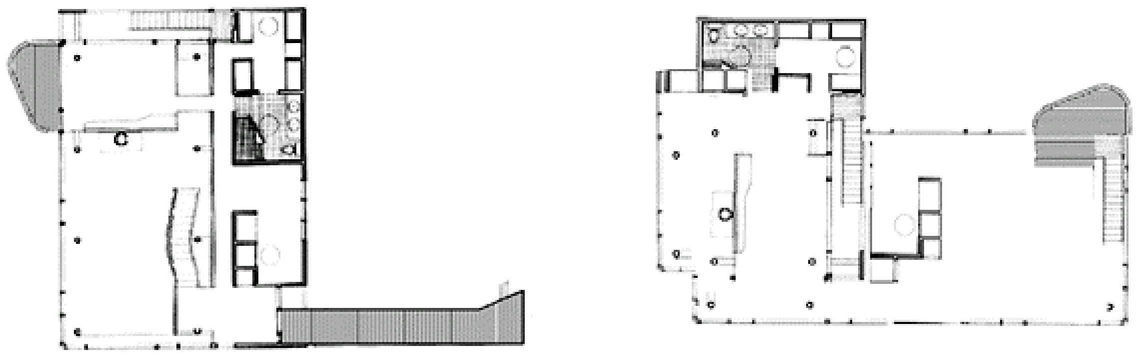
The Machine as an Assistant Designer

"The aim has been to leave the value judgement to the user and have the system do all the procedural and tedious work which will check the feasibility of the user's propositions " [Yessios, 1975: 3]

In the 1960s the role of computers in architecture was to replicate of human endeavors and to take the place of humans in the design process. In the 1970s the role was to create systems that would be intelligent assistants to designers, relieving them from the need to perform the more trivial tasks and augmenting their decision-making capabilities. Today, computers are increasingly involved in the design process. Their roles vary from drafting and modeling to intelligent knowledge-based processing of architectural information. While the future of computers appears to include a variety of possible roles, it is worth exploring these roles in the context provided by the question: “Who designs?” If one takes the position that designing is not exclusively a human activity and that ideas exist independently of human beings, then it would be possible to design a computer mechanism which relates ideas.

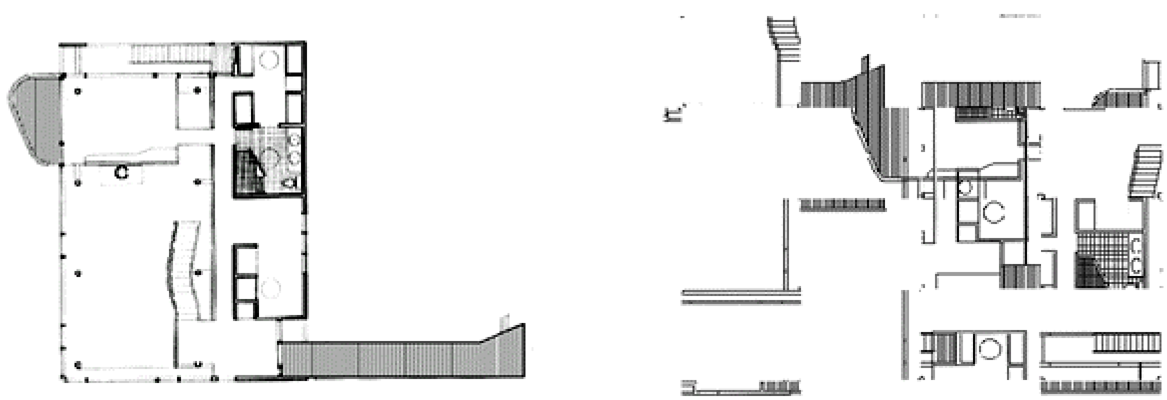
Visual Thinking

It can be argued that designers construct their drawings in a very precise way, and that we, as viewers, can recognize that the compositions are not random, perceiving the presence of a relational structure even if the construction itself evades us. In most paintings, drawings, or sketches we distinguish certain visual relationships that makes us believe that this must be a “meaningful” composition. Let’s take an example: In the architectural plans of Richard Meier one may notice that the vocabulary of building elements -- such as walls, columns, curvy walls etc.-- are repeated over and over in his plans. The plan in the figure below (left) corresponds to an actual house. The plan on the left is a re-arrangement of the same elements found in the first plan but in a different order.



A plan (left) and a re-arrangement of the plan (right).

The difference between the two buildings - or rather between the two compositions -- is not one of the number of elements -- since no parts are added or subtracted. The difference lays in the relationships of the parts within the plan. After all, of all the infinite ways that these elements can be arranged on a white piece of paper, only a few would be seen as architectural plans, and even fewer still as those of Meier. The composition on the right below is a complete random generated composition of pieces taken from the plan (left).



A plan (left) and a random re-arrangement of the plan (right).

But then, what is that which makes a plan to look like a plan, a map to look like a map, or a poem to look like a poem? Two answers seem possible: first, since constructed objects show a high degree of interrelationship, we can conclude that “coherence” is something that we have the intrinsic capacity to discern and which we value, since it helps us distinguish between randomness and order. We can speculate that we respond to forms as groups of relations, or rules. Second, it appears that sometimes the relationships, which hold together the elements of a composition, are stronger than the vocabulary of elements. This can be expressed by representing the structure of a composition as opposed to its content. In other words, what makes a Meyer plan look like a Meyer plan relies strongly on the relationships, which all Meyer plans have in common?

Along with the increased sophistication of Computer-Aided Design (CAD), it has been becoming increasingly necessary and desirable for a computer to recognize visual compositions. The recognition process can be used not only as means to input information into computer but also as a vehicle to investigate how we as human see, recognize, and appreciate visual information. Methods exist for inputting and recognizing engineering drawings and diagrams. This is primarily because they are drawn to conform to specific standards. In contrast, graphic or architectural designs are not always prepared in accordance to existing standards.

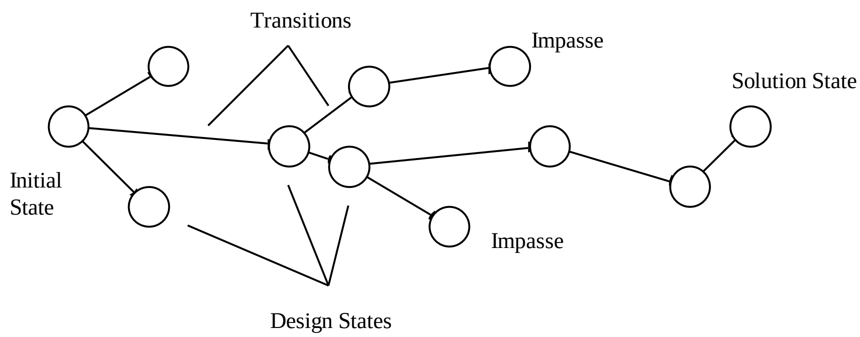
Computer vision is an area of artificial intelligence that investigates into how computers can perform the process of vision. Computer vision systems are designed to capture, analyze, and recognize visual information, create high level descriptions from sensed data, and, then, feed this information to CAD systems for further processing. In general, computer vision systems attempt to recover useful information about the three-dimensional world from image arrays of sensed values. Vision algorithms try to detect relations among image data. These data are numbers representing light intensity or depth range. A typical vision system operates in the following manner:

- Clears the raw image data from noise and “outliers”, that is, from extreme or missing information.
- Detects features, such as, edges and lines. This is done by observing local changes of the image's intensity. The detection of the geometrical information is similar to that of solving a statistical regression problem.
- Models the data in parametric form. This is usually referred to as segmentation and reconstruction. Here, the main task is to fit models to the image data.
- "Recognizes" objects. This task is usually accomplished by comparing the sensed data to a given model. If a match occurs, the object is assumed to be "recognized," otherwise it is labeled as "unknown."

It may be possible to assume that vision systems can enhance the conduct of visual interpretation and analysis by allowing the researcher to access information more systematically and efficiently. For example, the automatic extraction of compositional rules from drawings may provide one with insight concerning the designer's form-making approach and intentions. Computer systems are good at dealing with vast amounts of information, sorting, classifying, and analyzing it. In contrast, human experts are good at detecting higher level abstractions. Hence, it is more likely than not that visual analysis can be enhanced substantially by the interaction of humans and machines. Thus, knowledge concerning the visual structure of notable compositions and classes of compositions can be acquired.

Intelligent Process

The process of confronting design as a structured problem has been discussed by many theorists in the area of artificial intelligence and many models have been developed and implemented. The main concern of those theorists is to what degree can design be rationalized. Simon's thesis [Simon, 1981] is that design is an ill-structured problem, but it can be solved by considering, not one, but a spectrum of alternative solutions and choosing the most satisfying one. In order to produce those alternative solutions, design has to be first viewed as a problem-solving process.



Design as a goal-directed search process

If the design process is viewed as *a problem-solving process*, design may be conceived as a far more systematic and perhaps rigorous activity. As defined under a theory formalized by Simon in collaboration with other researchers like Newell [Newel, 1990] over the past twenty years, for every problem a *solution space* exists, that is, a domain that includes all the possible solutions to the problem. Problem-solving can then be characterized as the process of identifying and evaluating alternative solutions in this space in order to discover one or several which will meet certain goals and may therefore be considered to be appropriate and desirable.

Areas of AI

Artificial intelligence (AI) is a branch of computer science concerned with the problem of how to simulate human intelligence. AI is as old as the invention of the first computer, or, to be more precise, of the first counting machine. The main areas, which AI is concerned with, are:

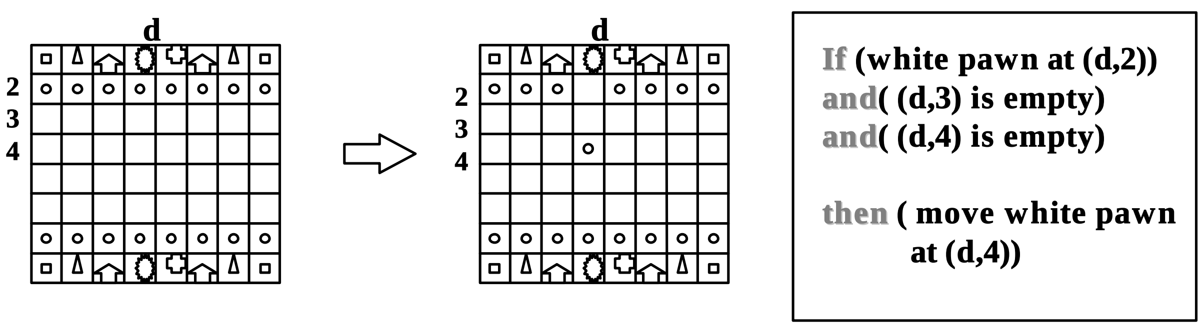
1. *Natural Language Understanding*, which is concerned with problems of understanding the syntax and semantics of natural language.
2. *Knowledge representation*, which is concerned with how do humans represent and process information in the form of symbolic structures.

3. *Inference or Learning*, which is concerned with the process of learning and the development of strategies that will eliminate repeated mistakes by learning (also referred by Simon as *chunking* [Simon, 1981]).
4. *Pattern Recognition*, which is concern with problems of human vision and ways to develop artificial senses.

Among these areas the tasks of game playing, problem solving, perception, and natural language understanding can be more directly associated with the design process. Each of these tasks seeks to abstract the processes of human thinking and intelligence. By doing so they hope to extract rules, methods, or patterns, which they can use to “teach” a computer how to “think” in the same way. Design is a mental process that among other things requires thinking and intelligence. In the following paragraphs we will attempt to inquire into the connections between design and intelligence from an artificial intelligence point of view.

Game Playing

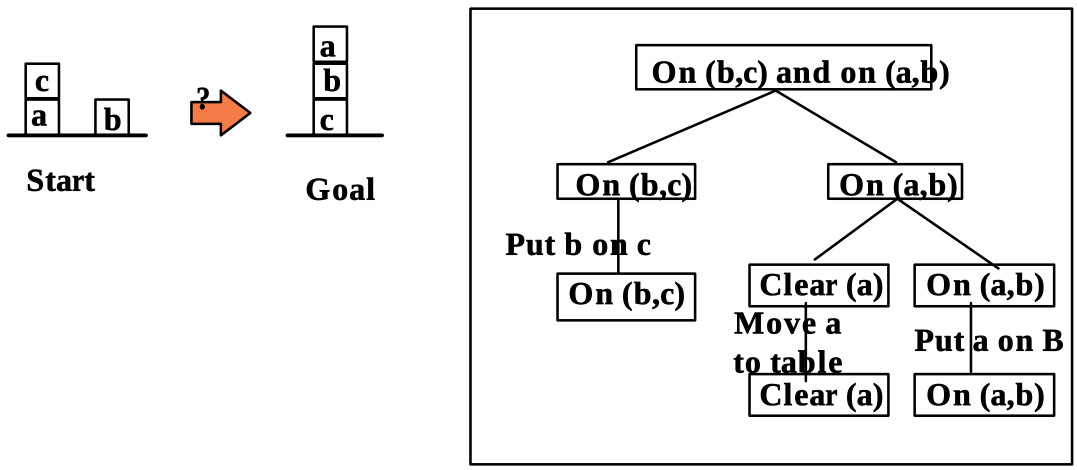
In game playing one of the objectives is to make intelligent moves. For example, in chess, there are rules, strategies, and tactics. Every move has to fulfill local and global goals. In design we can also acknowledge the involvement of rules, strategies, and tactics during the design process. The question however is what are the goals in design. What is the local and what is the global goal? One of game playing properties is that although people who do them well are considered to be displaying intelligence, it appears that computers can perform well by simply exploring a large number of solution paths in a short time and then selecting the best. It seems that this process required little knowledge and could be therefore easily programmed. In other words, computer’s involvement in the design process does not have to be that of imitation rather than that of extension.



A chess move is being codified as a set of logical predictions.

Problem solving

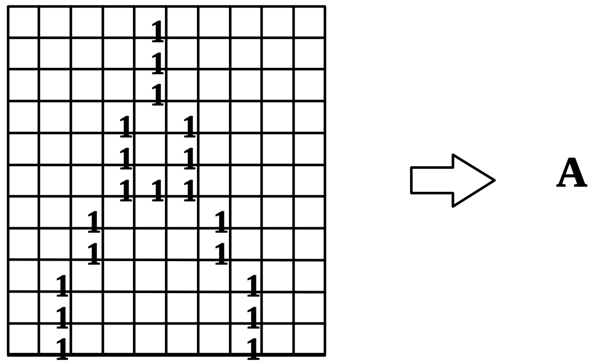
Problem solving is another area of AI that may be useful in design. It involves a start, a goal, and a strategy. Every move has to be analyzed extensively and by considering a set of logical possibilities a solution strategy may eventually be found. To investigate this kind of reasoning methods must be developed that incorporate matching, indexing, heuristics, generate-and-test, hill-climbing, and breadth-first searches. All of these methods are general and aim at solving a problem: any problem. The question is what are the problems of design. Can they be defined? What are the goals in design?



Sorting a set of block involves a series of logical actions.

Perception

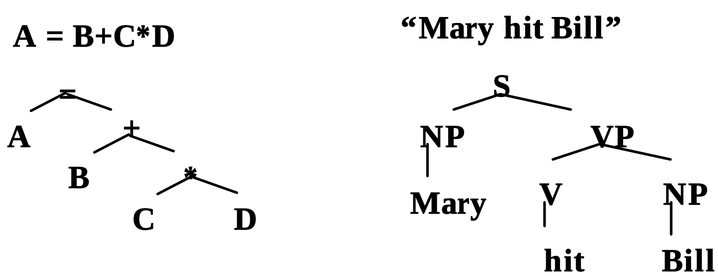
Perception is an area of AI that includes vision and speech. Among others perception tries to find how do we recognize patterns; how do we see. Perceptual tasks are difficult because they involve analog (rather than digital) signals, the signals are typically noisy, and usually a large number of things (some of which are obstructing others) must be perceived at once. Nonetheless their contribution to intelligence are extremely significant since perception is the main link with the external world. For design tasks, perception can contribute not only in the design process but also as a vehicle for understanding, recognizing, and criticizing design solutions. But what are the criteria for evaluating design? Can they be specified?



A letter recognition

Natural Language Understanding

The ability to use language to communicate is an important medium that separates people for animals. The problem of understanding spoken language includes the problems of perception and is extremely difficult. Nonetheless, even if we restrict the problem to written language it is still very difficult. This problem is also referred to as natural language understanding. In order to understand a sentence it is not only necessary to incorporate knowledge about the language's structure (vocabulary, grammar, spelling, and syntax) but also to know about the topic discussed so that implied statement are understood. One of the problems of design is how do designers communicate ideas. This involves not only the actual exchange of straightforward information but the expressions, metaphors, gestures, sketches, etc. involved in conversations between designers. How AI can help designers communicate locally or remotely may be an interesting undertaking especially through the use of computer networks. Another area of design where natural language understanding may be useful is in the notion that design is a language of shapes. By analyzing natural languages is it possible to help designers compare and understand their formal languages.



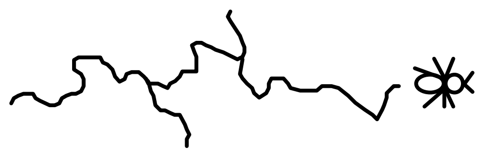
Analysis of a sentence

Techniques of AI

Typically, problem-solving tasks utilize various techniques for extracting, locating, and evaluating information. Three major steps are required to build a system that will solve a particular problem [Rich, 1983]:

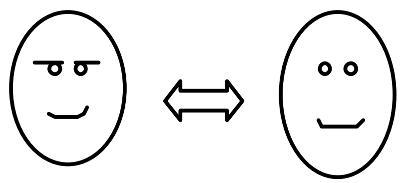
1. Precise *definition* of the problem. This includes precise specifications of the initial situation(s) and what final situations constitute acceptable solutions to the problem.
2. *Analysis* of the problem. Precise analysis of sub-problem, alternative paths, appropriateness of each path.
3. Selection of the *best* techniques and implementation.

Some of the techniques most commonly used in AI that can be also found in the design process are search, matching, and planning. In the case of search, a series of guesses are being investigated and each one is evaluated on how close it brings the strategy to the final solution. Those guesses that are not promising are abandoned and those that are promising are implemented.



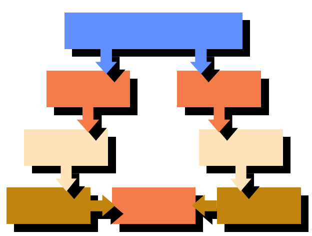
Search

In the case of matching, a series of samples are compared to a prototype that functions as a model solution. Each comparison is evaluated on the basis of percentage of match.



Match

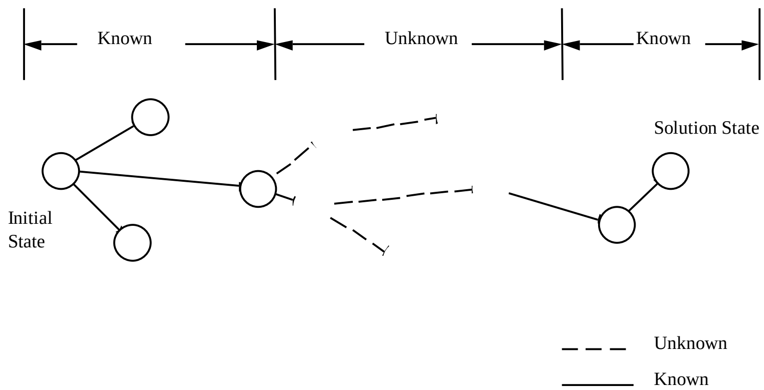
In the case of planning, a problem strategy is broken into sub problems and by solving each one separately one gets closer and closer to the final solution.



Plan

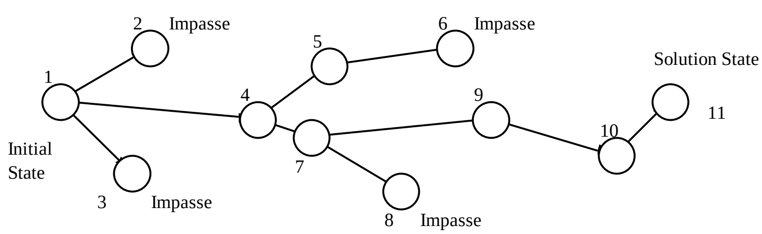
From the content of each step or technique of AI, it becomes apparent that problems have to first be defined in order be solved. However, there are problems where either there is not enough information about them or they are ill defined. Design problems, especially in architectural design, are often ill defined or misinterpreted. In those cases, a different approach is being applied that of on-line algorithms. In such techniques, like stochastic reasoning, fuzzy logic, or Markov chains, partial information is codified and percentages of possibility are attached to each direction. Two methodologies tend to be most representatives in dealing with the unknown. These two methodologies develop "strategic knowledge to enable one to decide when to do what" [Simon, 1981]. The two methodologies are means-ends analysis and generate-and-test cycle.

Means-ends analysis works in a backward mode, employing the simple strategy of assessing the difference between the current and the final state and then taking steps to reduce this difference.



Means-ends analysis

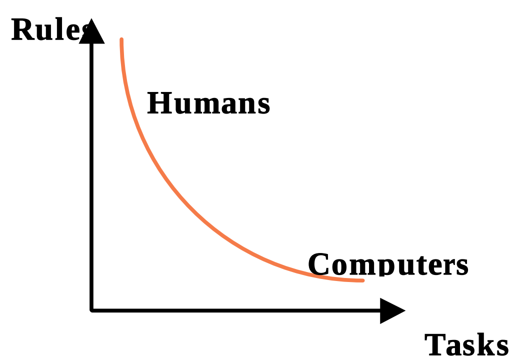
Generate-and-Test Cycle begins with the generation of alternative paths and then proceeds to test those alternatives against a whole array of requirements and constraints. These need not be merely single generate-test cycles, but there can be whole nested series of such cycles.



Generate-and-Test Cycle

Human-machine Interaction

Computers are powerful processing devices with enormous memory capacities. Human beings are very good in solving complex problems or in artistic creation. What is the power of the machine? What is the power of the humans? It seems that humans are good in processing high-level abstract information. To mention a few, judgement, interpretation, and creativity are unique characteristics of human thought. On the other hand humans are very slow in complex number calculations, or in memorizing large amounts of information. Computers and humans seem to be good each one in different areas of thought. The following diagram [Newel, 1990] shows that humans are good in using many rules and a few tasks, whereas computers are good in using less rules and performing many tasks. The question that may be posed, is whether computers and humans can co-exist and whether they can complement one another. This and other possibilities will be discussed in the next section in the context of design.



Human-machine relationship

AI in Design: Scenarios

As computers become more and more capable of performing many tasks in speeds that surpass by far human performance it is important to ask what kind of relationship will humans and computers have in the future. In other words, as computer performance increases and their capabilities become more and more close to those of humans how can AI research relate to humans? What operations make a computer useful to a human or what computer operations are impossible for humans to perform? Three scenarios seem possible. The computer may:

1. complement the human thinker
2. extend the process of thinking
3. replace the human thinker

Each of these scenarios is possible to happen either in the near or the far future. Each of them is based on extrapolation of current trends and research projections in AI.

According to the three scenarios shown above, the following speculations may be possible to happen in intellectual processes related to design:

1. scenario A: The computer can assist the designer
2. scenario B: The computer can extend the designer's mind
3. scenario C: The computer can replace the designer

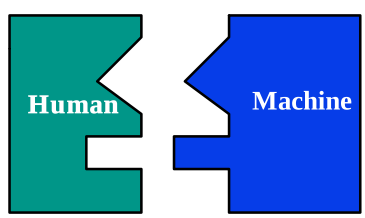
In the following sections we will briefly describe, investigate, and evaluate the impact of these scenarios for design and the designer. It important to repeat here, that these scenarios come as a logical consequence of extrapolating today's machines' capabilities, their trends, and the promises of AI research. There is no intention to intimidate or give false hopes. These are merely speculations that may or may not turn out to be true.

Scenario A: The computer can assist the designer

According to this scenario, computers provide useful information, advise, appraise, and assist the designer. The goal of Computer-aided Design (CAD) and computer vision systems is included in this direction. They function mainly as tools for the designer and can help either during the design process or in the early stages of design. Expert systems are also an area of AI whose goal is to assist the designer. Their target at providing the designer with useful information and functions for consultation during the design process.

Because of the large number of constraints to be simultaneously considered in any design problem, computer can be very useful in helping the designer meet them all. Moreover, the complexity of the design problem is so great that a designer would be unable to arrive at an appropriate solution unless a computer is used to break down the problem into sub-problems and use a computational approach to solve them [Bernholtz, 1969].

A

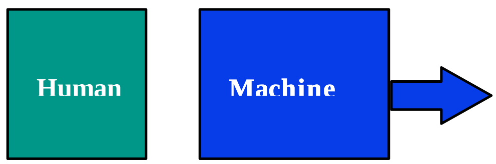


Scenario B: The computer can extend the designer's mind

According to this scenario, the computer functions as a tool that allows the designer to explore alternative possibilities and extrapolate into unknown intellectual territories. Abstract entities such as events, experiences and ideas, can become symbolically represented and transmitted through electronic devices. Through the use of mathematical models, it is possible to visualize those abstract entities, verify their existence and project their behavior into a once unimaginable world.

The introduction of new electronic media in the last fifty years gave a different twist to the exploration of these mathematical notions. The ideas of mathematical models and simulations were realized through fast computations and large memory capacities. A world was *discovered*, the world of *virtual reality*, which is a "make-believe" representation of mathematical models. This world can be projected to the computer screen or animated through real-time computations. Objects, represented through instructions in the computer's memory, were projected to a screen by simple algorithms, then transformed as if they were physically there, occasionally dressed in fancy textures and, in special cases, animated and transformed infinitely.

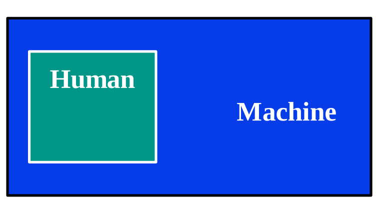
B.



Scenario C: The computer can replace the designer

According to this scenario, the intellectual capabilities of a computer may become far superior to those of the human designer. In other words, a computer program may replace the designer. Since the invention of the digital computer, theorists strive to find ways to relate computers to human thinking. Computers are arithmetic devices, which can perform all basic arithmetic operations, such as addition, subtraction and multiplication. By combining basic operations computers are also able to perform complex algebraic operations and derive accurate results in minimum time. Furthermore, computers have the ability to operate as logical devices, in the sense that they can perform logical operations, such as AND, OR and XOR. Given a number of truth tables, computers are able to verify the truth or falsity of a logical sentence and therefore to determine the validity of an argument [Von Neumann, 1945]. This latter capability led computer theorists to inquire whether those arguments could be compatible to problems taken from the real world. In other words, whether it is possible to develop cognitive mechanisms, which would process information from the real world and derive answers or propose solutions, as if they were carried out by humans. Some theorists expect to see even more than that. They expect to see computers which would be able to simulate human thinking to a degree such that they would perform tasks which are considered by humans to be "highly intellectual", such as design.

C.



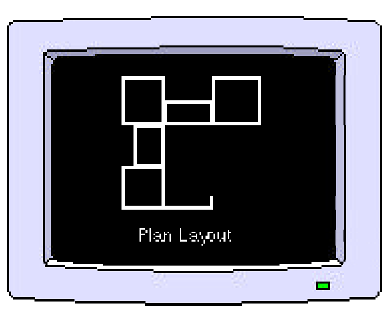
Design as a Language

An approach to solving design problems is that of *linguistics*. Here, the designer attempts to structure the problem by grouping the constraints into thematic areas (e.g. zoning, circulation), and proceeding to design by considering each group of constraints more or less independently. This information is converted into linguistic structures through the use of transformational rules. Then, the designer represents these linguistic structures in the form of sentences [Chomsky, 1957], that is, specific sets of design elements, which include not only the elements but also the rules which allow a designer to combine them into feasible and meaningful compositions. The aim of this approach became one of writing algorithms for the generation of feasible and meaningful design "sentences."

Although not directly related to linguistics, the theory of algorithms was worked out in considerable detail by Markov. He suggests that any algorithm should be definitive, universally comprehensive, general, and conclusive. Algorithms have been written for the design of buildings, or rather, for designing parts of buildings. Over half of these algorithms have been concerned with *space allocation* problems, some of which resulted in formal definitions or languages which the computer can be programmed to resolve, such as SIPLAN [Yessios, 1975].

In addition to synthesizing form, computers must also be able to accept and process non-geometric information about form. Therefore, it is necessary for design languages to be invented to describe operations on geometric databases. One pioneering effort in this area is GLIDE [Eastman and Henrion, 1976], a language which allowed the user to assemble buildings.

Another approach in the direction of computer-augmented design, was the manipulation of forms according to rules [Mitchell, 1974]. Basic structural and functional elements were assembled to make volumes (elements of composition) which, in turn, were assembled to make buildings. All elements were stored in the computer's memory in symbolic form, and the user operated on them by manipulating symbols in accordance with rules derived through the classic academic tradition.



A layout diagram on a raster screen

Formalistic Process

To address issues of architectural or graphics design process and products it is necessary for the researchers to consider the idiosyncrasies of those types of design design. In architecture, for example, design quality is reflected in forms and their relationships. Many designers and theorists have argued that what distinguishes a well designed form from one that is poorly designed can only be found in the morphological relations that the former embodies. "One can have a beautiful idea of winning a chess game. One can brutally win a chess game in a very inelegant way. But there can be elegance in the process of winning itself that is poetic. We are looking for the poetic in the process, regardless of the result. We are looking for a beauty internal to the idea of the play, that is, when one suddenly gets the shiver." [Ford, 1986: 34]

Formalistic design is viewed as an activity, which entails invention and exploration of new forms and their relations. Various methods of analysis have been employed in the search of new forms: formal analysis involves the investigation of the properties of a formal subject. Composition, geometrical attributes, and morphological properties obeying Galilean and Newtonian principles are extracted from figural appearances of an object. In contrast, structural analysis deals with the derivation of the motivations and propensities which are implicit within form and which may be used to define the limit between what it is and all other possibilities. [Brown, 1986]

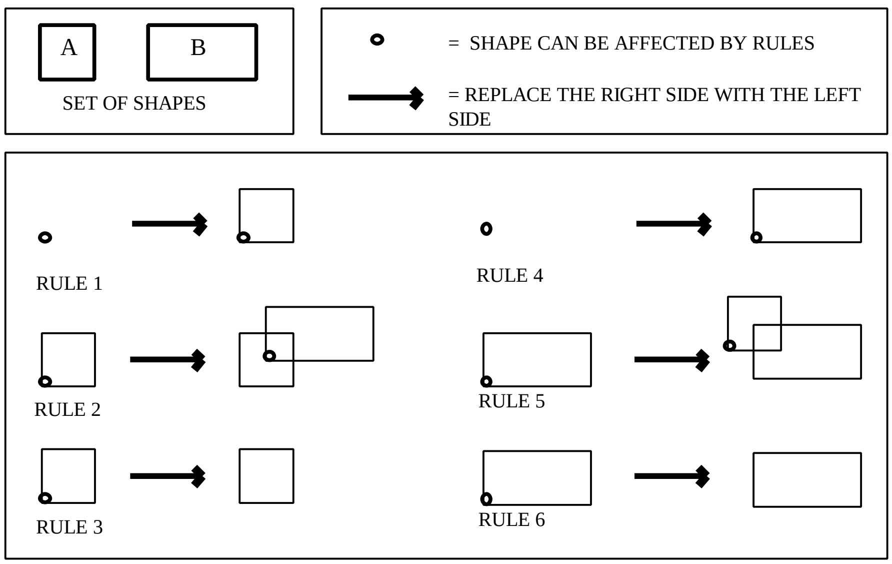
Shape Grammars

The Shape Grammars approach [Stiny, 1985] was developed as an architectural theory to analyze and synthesize architectural schemes. Although initially developed to carry out spatial computations visually, it was later on extended to explain design phenomena, such as stylistic changes [Knight, 1986], and to simulate behavioral patterns of design, such as languages of design [Flemming, 1987].

A Shape Grammar consists, in general terms, of an *initial* shape, and a set of *production rules*. The rules apply to the initial shape and to shapes produced by previous rules applications, to generate designs. All designs generated by the rules comply with the language generated by a grammar. A shape grammar has four components:

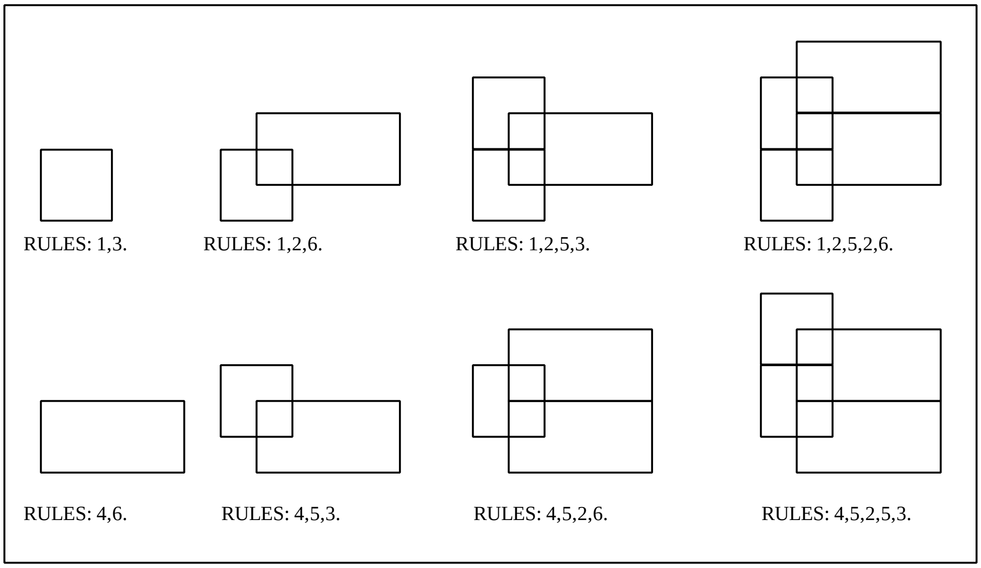
1. S is a finite set of shapes;
2. L is a finite set of symbols;
3. R is a finite set of rules of the form a " b, where a is a labeled shape in (S, L)+", and "b is a labeled shape in (S, L)*"; and,
4. I is a labeled shape in (S, L)+ called the initial shape.

Thus, after establishing a vocabulary in the form of a finite set of shapes and symbols, the shape grammar formalism makes possible to define a set of production rules with them (see figure below). A rule in this sense is not a restriction but rather a representation of actions to be performed when specific conditions are met. The right side of the rule represents the conditions to be met for the rule to be applied, and the left side represents the action to be taken if the conditions are present. The grammar thus defined allows transforming a shape or configuration into another shape or configuration by applying one rule at a time. This is done by replacing "marked" instances of the shape on the left-hand side of the rule, with the shape or shapes in the right hand side. A special symbol such as a dot (.) is used to identify and distinguish "marked" instances of a shape, and an implementation arrow to separate the right from the left-hand sides of the rule. All shapes on the left-hand side of the rule definition must be marked while the shapes in the right hand side may or may not be marked. If the shape or shapes on the right side are not marked, the instance of the application of that particular rule cannot be further transformed, since only marked shapes can be affected by rules. Figure 1 shows a simple six-rule grammar derived from a vocabulary of two primitive shapes, and two basic symbols.



Sample Shape Grammar

The dot on the left side of rules 1 and 4 indicates an empty shape. Rules like these are used to start the production process by adding the initial shape to a layout (shape being created by the grammar). Rules 2 and 5 are typical production rules that add a new shape to the layout. In this grammar, both rules "unmark" the shape to which they were applied (base shape of the rule), and add a "marked" new shape so that further rules can be applied to it. Rules 3 and 6 "unmark" shapes A and B respectively. In this particular grammar, any application of rule 3 to shape A or rule 6 to shape B, will terminate the production process since in both cases the only marked shape left will be unmarked, eliminating the possibility of further rule applications (see figure below , where the number of rule applications is limited to 5 rules only).



Sample Layout Using the Shape Grammar

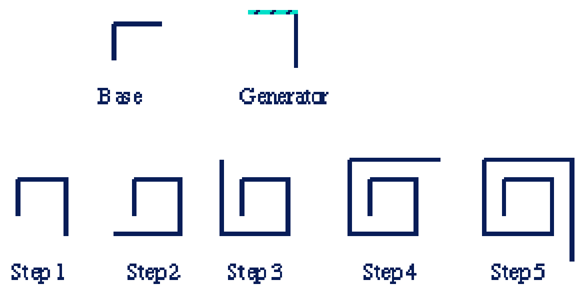
The Shape Grammars Theory was originally developed as an alternative approach to *algebraic structures*, a branch of modern algebra, which argues that within an algebraic system there are operations, which can describe and demarcate the system. By defining a number of operations it possible to describe any pattern within the system. Those operations were extended and interpreted by other theorists as *production rules*, and used to explain cognitive phenomena.

Shape Grammars can be used to define new languages of design. They can also be used to define languages of design in known styles. Work concentrating on styles has been reported by Knight [Knight, 1981b; 1983a; 1983b; 1983c], Flemming [1981], Mitchell [Stiny and Mitchell, 1980] and others. Shape Grammars are characterized by (1) clarifying the underlying structure and appearance of known designs, (2) supplying the conventions and criteria necessary to recognize whether any other design is an instance of a style and (3) providing the compositional machinery needed to generate new designs.

In summary, Shape Grammars provide a theoretical framework for understanding designs, for constructing languages of designs and for explaining phenomena of design such as stylistic changes. Furthermore, as a theory of architecture, Shape Grammars provide mechanisms for understanding shapes as designs, by first appealing to compositional styles characterized by languages defined by grammars, and then to languages of descriptions to provide accounts to designs in terms of their function, meaning, etc.

Generative Systems

An interesting variation of shape grammars is that of *fractal generative systems*. Based on a scheme, formulated by the German mathematician Von Koch, a fractal process consists of an initial shape (the base) and one or more generators. From a practical point of view, the generator is a production rule: each and every line segment of the base is replaced by the shape of the generator. The implementation of an interactive computer program has been reported by Yessios [1987] that allows the fractal to be generated one at a time or at multiple increments, backwards or forwards. As described by Yessios, "a building typically has to respond to a multiplicity of processes, superimposed or interwoven. Therefore, the fractal process has to be guided, to be constrained and to be filtered. The fractal process has to be 'mutated' by the utilitarian requirements of the functionality of a building."" [Yessios, 1987: 7]



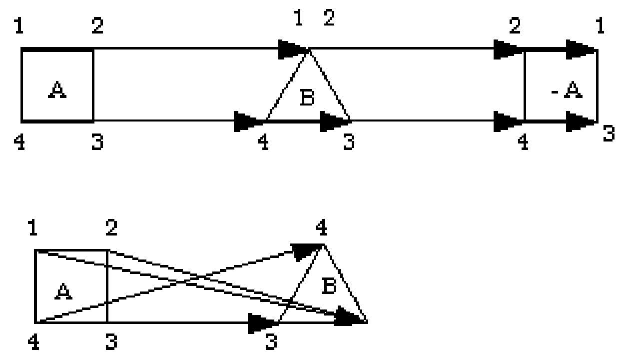
A generative scheme

Transformation (Morphing)

Transformation or *morphing* is a process in which an object changes its form gradually in order to obtain another form. The operation of transformation consists basically of the selection of two objects and the assignment of *n*, the number of in-between steps. The first object then transforms into the second in *n* steps. This process is illustrated in figure below. The transformation preserves the structural integrity of the objects involved, that is, an object changes into another object as a single entity. There are many possible ways an object can be transformed. By matching pairs of points, edges, or faces, one of each object, the transformation process can be altered.

Orchestration is a term used to describe the actions of selecting, assigning, directing and evaluating the performance of objects, which participate in a transformation. Transformations can happen concurrently and/or in different speeds. The result is a *moving image* the behavior of which becomes the responsibility of the user. As in an orchestra performance the designer/composer selects a number of objects s/he wants to include, assigns the proper transformation paths and speeds, and then directs the performance through time, form and color.

The essence of such transformational design is not that much in the final form but rather in the intermediate phases these transformations pass through, as well as, in the extrapolations which go beyond the final form. The user has the capability, through the system, to modify and control the flow of the compositional evolution and replay it many times by varying some or all of the transformational parameters.



The transformation process

One interesting exploration of *shape transitions* has been reported by Yessios [1987]. According to him an initial shape A can be transformed to a target shape B by applying any number of in-between steps. All the points of shape A are mapped onto shape B and vice-versa. Furthermore, once the rules of transition have been established, the transition can be allowed to continue beyond its target, to infinity.

Transformations involves two important principles of form: stability and change [Eisenman, 1986]. A transformation is not exactly a form-making procedure because the subject of transformation must already be complete. In a transformation, only relations change. No new elements can be introduced or removed; bits cannot be added or taken away. However, the illusion of movement, often described as "frozen movement", has been argued to have a high formal value. [Evans, 1986] It illustrates the forces designers have referred to, as “punctured volumes,” “compressed planes,” “interpenetrating spaces,” or “agitated surfaces.”

KEYWORDS

AI, 7
algebraic structures, 19
Artificial intelligence, 7
breadth-first searches, 9
coherence, 5
Computer vision, 6
Computer-aided Design, 14
computer-designer, 2
Expert systems, 14
Formalistic Design, 17
fractal, 20
fuzzy logic, 11
Games, 3
Generate-and-Test Cycle, 12
generative systems, 20
goal seeking, 3
heuristics, 9
hill-climbing, 9
Human-machine, 12
Inference or Learning, 8
Knowledge representation, 7
Limitations, 2
linguistics, 16
Markov chains, 11
matching, 11
Means-ends analysis, 12
morphing, 20
moving image, 20
Natural Language Understanding, 7
on-line, 11
Orchestration, 20
Pattern Recognition, 8
Perception, 9
planning, 11
problem-solving, 7
production rules, 17
rules, 3
search, 11
Shape Grammars, 17
shape transitions, 21
solution, 7
stochastic reasoning, 11
strategies, 3
styles, 19
tactics, 3
Transformation, 20
virtual reality, 14
Vision algorithms, 6

SUMMARY

- Computers must be acknowledged not only as machines for imitating what is understood, but also as vehicles for exploring what is not understood.
- To employ computers more effectively in the creative process, they must also include the meaning of information.
- Games employ rules, strategies, tactics, and goal seeking that may be useful beyond the game's boundaries, e.g., in design.
- While the future of computers appears to include a variety of possible roles, it is worth exploring these roles in the context provided by the question: “Who designs?”
- In most paintings, drawings, or sketches we distinguish certain visual relationships that makes us believe that this must be a “meaningful” composition.

- Along with the increased sophistication of Computer-Aided Design (CAD), it has been becoming increasingly necessary and desirable for a computer to recognize visual compositions.
- Computer vision is an area of artificial intelligence that investigates into how computers can perform the process of vision.
- Design is an ill-structured problem, but it can be solved by considering, a spectrum of alternative solutions and choosing the most satisfying one.
- If the design process is viewed as *a problem-solving process*, design may be conceived as a far more systematic and perhaps rigorous activity.
- *Artificial intelligence (AI)* is a branch of computer science concerned with the problem of how to simulate human intelligence.
- Among these areas the tasks of game playing, problem solving, perception, and natural language understanding can be more directly associated with the design process.
- In design we can also acknowledge the involvement of rules, strategies, and tactics during the design process. The question however is what are the goals in design. What is the local and what is the global goal?
- Problem solving involves a start, a goal, and a strategy. What are the problems of design. Can they be defined? What are the goals in design?
- Perception is an area of AI that tries to find how do we recognize patterns; how do we see.
- In order to understand a sentence it is not only necessary to incorporate knowledge about the language's structure (vocabulary, grammar, spelling, and syntax) but also to know about the topic discussed so that implied statement are understood.
- Some of the techniques most commonly used in AI that can be also found in the design process are search, matching, and planning.
- There are problems where either there is not enough information about them or they are ill defined. Two methodologies tend to be most representatives in dealing with the unknown. Means-ends analysis, and Generate-and-Test Cycle.
- Computers and humans seem to be good each one in different areas of thought.
- The computer can possibly assist, extend, or replace the designer
- An approach to solving design problems is that of *linguistics*.
- Formalistic design is viewed as an activity, which entails invention and exploration of new forms and their relations.
- A Shape Grammar consists, in general terms, of an *initial* shape, and a set of *production rules*. The rules apply to the initial shape and to shapes produced by previous rules applications, to generate designs.
- *Transformation* or *morphing* is a process in which an object changes its form gradually in order to obtain another form.

REFERENCES

- Alexander, C. Notes on the Synthesis of Form, Cambridge: Harvard University Press, 1967.
- Berkeley, E. "Computers for Design and Design for the Computers," Architectural Forum, 128 (2), (March 1968).
- Bernholtz, A. and Bierstone, E., "Computer Augmented Design," Design Quarterly, (1966/67).
- Boullee, E.-L., "Architecture, Essay on Art," in *Treatise on Architecture*, Rosenau, H. (ed.), Paris: MS Francai's 9153 Bibliotheque National, 1953, p.83.
- Brown, A., "In Caesura", in Eisenman Studios at GSD: 1983-85, Cambribge: Harvard University Graduate School of Design, 1986, p. 35.
- Calinescu, M., The Five Faces of Modernity, Duke University Press, 1987.
- Chomsky , N., Syntactic Structures, The Hague: Mouton & Company, 1957.
- Dietz, A. , Dwelling House Construction Cambridge: MIT Press, 1974.
- Eastman C. M. and Henrion M., M. GLIDE: Language for a Design Information System. Pittsburg: Carnegie-Mellon University, Institute of Physical Planning, 1967.
- Eisenman P., "The Futility of Objects", Harvard Architecture Review 3, (1984), p. 66.
- Evans, R. ,"Not to be Used for Wrapping Purposes", AAFiles 10, (1987), p. 70.
- Flemming, U., "The Role of Shape Grammars in the Analysis and Creation of Design", Proceedings of Symposium on Computability of Design at SUNY Buffalo, (December 1986).
- Ford, K., "In Caesura", in Eisenman Studios at GSD: 1983-85, Cambridge: Harvard University Graduate School of Design, (1986), p. 35.
- Gill, A., System Modeling and Control, New York: John Wiley & Sons, 1978.
- Kalay, Y., "Redefining the Role of Computers in Architecture: from Drafting/Modeling Tools to Knowledge-Based Design Assistants", Computer Aided Design, 17 (7), (1985).
- Knight, T. W., *Transformations of Languages of Design*, Ph.D. Dissertation. Los Angeles: University of California, 1986.
- Lafue, G. M. E., "Integrating Language and Database for CAD Applications", Computer Aided Design 11 (3), (1979).
- Leighton, N., Computers in the Architectural Office, New York: Van Nostrand Reinhold, 1984.
- Mitchell, W., "Vitruvius Computatus", in W.F.E. Preiser (ed.), Environmental Design Research, Proceedings of EDRA 4 Conference, Stroudsbourg: Dowden, Hutchninson and Ross, 1973.
- Negroponte, N., The Architecture Machine. Cambridge: MIT Press, 1970.
- Negroponte, N., Soft Architecture Machines. Cambridge: MIT Press, 1974.
- Newell, A. Unified Theories of Cognition, Cambridge: Harvard University Press, 1990.
- Newell, A. and Simon, H., Human Problem Solving, Engelwood Cliffs: Prentice-Hall, 1972.

- Novac, M., "Computational Compositions in Architecture," *ACADIA Proceedings*, Ann Arbor, 1988, p.7-25.
- Rasdorf, W. J. and Kutay, A. R., "Maintenance of Integrity During Concurrent Access in a Building Design Database." Computer Aided Design 16(4), (1982).
- Rich E., Artificial Intelligence, New York: McGraw-Hill, 1983
- Simon, H., The Sciences of the Artificial, Cambridge: MIT Press, 1981
- Stiny, G., "Computing with Form and Meaning in Architecture", Journal of Architectural Education 39, 1985, pp. 7-19.
- Stiny, G. and W. Mitchell, "Counting Palladian Villas", *Environment and Planning B* , 5, 1978, pp.189-198.
- Yessios, C., "A Fractal Studio", ACADIA 87 Proceedings, North Carolina State University, (1987).
- Yessios,C., "Syntactic Structures for Site Planning", in W.F.E.Preiser (ed.), Environmental Design Research, Proceedings of the EDRA 4 Conference, Stroudsbourg: Dowden, Hutchninson and Ross, 1973.
- Rich, E., Artificial Intelligence, New York: Mcgraw-Hill, 1983.
- Terzidis, K. And Vakalo, E.-G., "The Role Of Computers In Architectural Design," IAPS Proceedings, 1992.
- Von Neumann, On Computers and Human Thinking, Cambridge-Massachusetts: MIT Press, 1945.
- Yessios C. "A Fractal Studio", *ACADIA 87 Proceedings*, North Carolina State University, 1987.

FURTHER READINGS

- Simon, H. A. & A. Newell 1958. Elements of a Theory of Human Problem Solving, *Physiological Review*, 65: 151-66.
- Simon, H. A. 1972a. Human Problem Solving. Prentice-Hall., Englewood Cliffs, N.J..
- Simon, H. A. 1974a. How big is a chunk? *Science*, 183.
- Simon, H. A. & G. Lea. 1974b. Problem Solving and Rule Induction: A Unified View. In Lee W. Gregg (ed.), *Knowledge and Cognition*, pp. 105-28.
- Simon H. A. 1989. Models of Thought, vol. 2. New Haven: Yale University Press.

EXERCISES

1. Write short papers on the following subjects:

a) Can computers improve design? Why or why not? How?

b) Is it possible to create something with a computer that you cannot imagine in advance? (If yes, give an example. If no, explain why not)

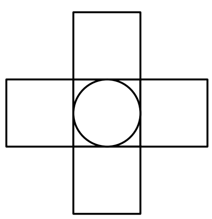
c) Is geometry a human invention or a human discovery? Why? (Hint: there is no right answer. You will be graded on the degree of argumentation)

d) Find out what does Godel’s theorem claim? How does it relate to computers and human thought?

e) Can the shape grammar theory explain how designers think?

2. How would you transform (morph) a square into an octagon? Show the connections.

3. Make a shape grammar that will produces a simple Palladian plan:



QUIZZES

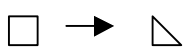
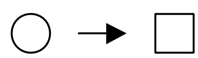
Quiz 1

1. Which branch of computer science concerned with the problem of how to simulate human intelligence?

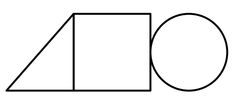
a) Computer vision
b) Problem solving
c) Perception
d) Artificial Intelligence

Circle all that apply: a b c d

2. How would you transform (morph) a triangle into a square? Show the connections.

3. If this is rule 1:  and this is rule 2: 

Apply the rules in the following sequence: rule 1, then rule 2, then rule 1 to the shapes below:



Show the result:

4. What is common between game playing and design?

Answer: ______________________________________

5. Which area of AI is concerned with how a computer recognizes visual compositions?

a) Pattern Recognition

b) Computer Vision

c) Perception

d) Shape grammars

Circle all that apply: a b c d

Quiz 2

1. What did automated design aim at:

a) Assisting the designer

b) Extending the designer’s mind

c) Replacing the designer

d) None of the above

Circle all that apply: a b c d

2. What does computer-aided design (CAD) aim at:

a) Assisting the designer

b) Extending the designer’s mind

c) Replacing the designer

d) None of the above

Circle all that apply: a b c d

3. What is the assumption for dealing with design as a problem solving activity?

a) be able to recognize patterns in a drawing

b) use on-line algorithms to solve the problem

c) reduce design to a rational activity

d) break the design process into linguistic sentences

Circle all that apply: a b c d

4. According to Sapir and Whorf, "a world view of a culture is limited by the structure of the language which that culture uses."

a) False

b) True

Circle one: a b

5. In the context of this chapter, what is “the external world”?

a) the world of imagination

b) the impression taken from projections on the surface of the eye

c) the world of philosophy

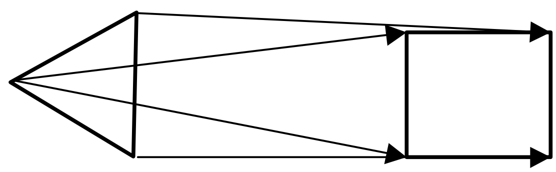
d) anything not related to computer software

Circle all that apply a b c d

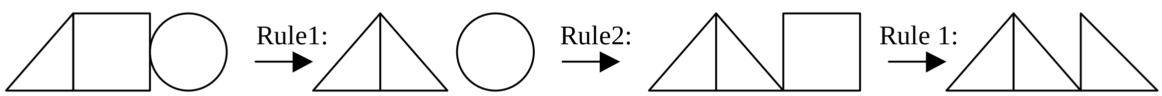
ANSWERS

Quiz 1

1. d) Artificial Intelligence
2. One solution is this:



3. The result is:



4. Rules, strategies, and tactics are common between game playing and design
5. Pattern Recognition, Computer Vision and Perception are all concerned with how a computer recognizes visual compositions

Quiz 2

1. A) Replacing the designer
2. B) Assisting the designer
3. The assumption for dealing with design as a problem solving activity is

c) reduce design to a rational activity

4. b) True
5. The “external world” is anything not related to computer software