

MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1309

June 1991

**Principles, Opportunism and Seeing in Design:  
A Computational Approach**

**Pegor Papazian**

**Abstract**

This thesis introduces elements of a theory of design activity and a computational framework for developing design systems. The theory stresses the opportunistic nature of designing and the complementary roles of focus and distraction, the interdependence of evaluation and generation, the multiplicity of ways of seeing over the history of a design session versus the exclusivity of a given way of seeing over an arbitrarily short period, and the incommensurability of criteria used to evaluate a design. The thesis argues for a principle based rather than rule based approach to designing documents.

The Discursive Generator is presented as a computational framework for implementing specific design systems, and a simple system for arranging blocks according to a set of formal principles is developed by way of illustration. Both shape grammars and constraint based systems are used to contrast current trends in design automation with the discursive approach advocated in the thesis. The Discursive Generator is shown to have some important properties lacking in other types of system, such as dynamism, robustness and the ability to deal with partial designs.

When studied in terms of a search metaphor, the Discursive Generator is shown to exhibit behavior which is radically different from some traditional search techniques, and to avoid some of the well-known difficulties associated with them.

Copyright © Massachusetts Institute of Technology, 1992

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124.

## ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis supervisor, Professor Patrick Winston, for initially giving the benefit of the doubt to an architect with ambitious ideas and for his subsequent help and support in all aspects of the thesis, as well as my three readers, in order of the distance of their respective offices from Professor Winston's at the MIT AI Lab: Professor Timothy Johnson of the MIT Architecture Department, for his constant support from the very beginning and particularly for his help in comparing my work to research done in the early 70's, Professor William Mitchell of the Harvard University Graduate School of Design, for his valuable suggestions, and Professor Alexander Tzonis of the Delft University of Technology Faculty of Architecture for his long-distance comments and his indirect advice to not make deals. Thanks are also due to the Cabot Fund and the ARS Lazarian Scholarship which provided part of the support for my research.

I would like to thank my parents, Hratch and Aroussiag Papazian, for their support and their love.

I would like to thank my friend Josep Maria Fargas for his constant companionship and for discussions about design, computers and the practice of architecture. The projects on "Disposable Metaphors" and "EstheR the Esthetics Replicant," both of which I developed with him, were precursors to the work presented in this thesis. I also owe thanks for companionship and stimulating discussions to Hashem Sarkis, always willing to combine an inspiring intellectual rigor with frequent ice cream breaks.

But above all, I am grateful to my wife Marie Lou. I thank her for expecting and accepting so much, for her love, the books she gave me, our conversations, and for joining me in suspending caution in the face of uncertainty.

## ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis supervisor, Professor Patrick Winston, for initially giving the benefit of the doubt to an architect with ambitious ideas and for his subsequent help and support in all aspects of the thesis, as well as my three readers, in order of the distance of their respective offices from Professor Winston's at the MIT AI Lab: Professor Timothy Johnson of the MIT Architecture Department, for his constant support from the very beginning and particularly for his help in comparing my work to research done in the early 70's, Professor William Mitchell of the Harvard University Graduate School of Design, for his valuable suggestions, and Professor Alexander Tzonis of the Delft University of Technology Faculty of Architecture for his long-distance comments and his indirect advice to not make deals. Thanks are also due to the Cabot Fund and the ARS Lazarian Scholarship which provided part of the support for my research.

I would like to thank my parents, Hratch and Aroussiag Papazian, for their support and their love.

I would like to thank my friend Josep Maria Fargas for his constant companionship and for discussions about design, computers and the practice of architecture. The projects on "Disposable Metaphors" and "EstheR the Esthetics Replicant," both of which I developed with him, were precursors to the work presented in this thesis. I also owe thanks for companionship and stimulating discussions to Hashem Sarkis, always willing to combine an inspiring intellectual rigor with frequent ice cream breaks.

But above all, I am grateful to my wife Marie Lou. I thank her for expecting and accepting so much, for her love, the books she gave me, our conversations, and for joining me in suspending caution in the face of uncertainty.



# Contents

<b>1. Introduction and Preview</b>	<b>1</b>
<b>2. Some Characteristics of Designing</b>	<b>7</b>
2.1 Focus and the Opportunistic Gaze	
2.2 Appraisal and Know-how	
2.3 Multiple Semantics	
2.4 The See-Move-See Cycle	
<b>3. Using Computers in/to Design: A Critique of Current Trends</b>	<b>25</b>
3.1 Shape Grammars	
3.1.1 Reliability vs. Power	
3.1.2 Principle Based vs. Rule Based Design Systems	
3.2 Designing With Constraints	
3.2.1 The Overwhelming Ubiquity of Constraints	
3.2.2 The Manifest Nature of Design Documents	
<b>4. A Discursive Architecture for Design Generation</b>	<b>45</b>
4.1 How the Discursive Generator Works	
4.1.1 The Forum	
4.1.2 The Persona	
4.1.3 The Arbiter	
4.1.4 The Focus Manager	
4.2 A More Subtle Way of Playing with Rectangles: An Application Using the Discursive Generator	
4.2.1 A Sample Run	
4.2.2 Five Modules for Arranging Blocks	
4.2.3 Parametric Next State Candidates	
4.2.4 Consolidation and Simultaneity	
4.2.5 Component-Level Focus	
4.2.6 An Overconstrained Case	
4.3 Why the Discursive Generator Works	
4.3.1 Discursive Generation as Search	
4.3.2 Low-Level Antidotes to Oscillation	
4.2.3 Redundancy for Robustness	
<b>5. Outline of Future Work</b>	<b>93</b>
5.1 Some Principles for Writing Rules	
5.2 Designing Design Personas	
<b>6. Conclusion and Overview</b>	<b>101</b>

"That world does not exist, one has to create it like the phoenix. That world exists in this one, but the way water exists in oxygen and hydrogen, or how pages 78, 457, 3, 271, 688, 75, and 456 of the dictionary of the Spanish Academy have all that is needed for the writing of a hendecasyllable by Garcilaso. Let us say that the world is a figure, it has to be read. By read let us understand generated. Who cares about a dictionary as a dictionary? If from delicate alchemies, osmoses, and mixtures of simples there finally does arise a Beatrice on the riverbank, why not have a marvelous hint of what could be born of her in turn?"

Julio Cortázar

# 1. Introduction and Preview

"The zig-zag of discovery cannot be discerned in the end product."

- Imre Lakatos

*Proofs and Refutations*

It may seem that the way to start the project of building a machine which generates designs is to study the characteristics of those designs. The idea behind such an approach would be to come up with a formal language for describing a design, with the hope of generating it and variations on it by manipulating instances of a descriptive language. One way to study the designs in question would be by identifying their syntactical properties, their components and the formal relationships among them. Another way would proceed by finding the causal relationships and constraints among components, and traces of the intentions behind them. But design objects have enough complexity and variety to defy a generation-by-description approach.

This thesis focuses on designing as an activity, and tries to identify some of its basic characteristics. It introduces some of these characteristics as elements of a theory of designing, and presents a computational framework for design generation based on that theory. The hope, in this case, is that a dynamic substratum of design activity can be developed which, even though it is simple, will be conducive to the complexity associated on the one hand with expertise in designing and, on the other, with design objects themselves.

The thesis introduces the Discursive Generator, a design system based on the following elements of a theory:

1. Opportunism, focus and distraction: Designing is based on a substratum of opportunistic activity. At any given time, the designer focuses on a limited number of components and evaluative criteria. When an opportunity is seen which can be exploited in terms of one of a large number of implicit and explicit values, the designer is distracted by it, and a shift of focus takes place. Focus can be a function of the history of a given

design session, the bias of the designer, higher-level processes such as planning and inference and, of course, external factors.

2. The interdependence of appraisal and know-how: Appraisal is the act of evaluating an evolving design from the point of view of a given criterion, and know-how consists in the repertoire of moves expected to improve the state of the design in terms of a given criterion. The two are closely linked because part of appraisal consists of identifying a potential for improvement based on the knowledge of a possible move. The designer appraises a document not just in order to test it, but mainly to generate the next design move.

3. Multiplicity of semantics and the exclusivity of seeing: The designer can interpret a design document or artifact by attributing to its components and their relations one of many sets of meanings. At any given (arbitrarily brief) period, the designer can be thought of as actively relying on only one of those potential semantics. The equivalent of simultaneous interpretation in terms of two or more sets of meanings can take place when the design moves generated in terms of each of the semantics are equivalent.

4. The see-move-see cycle and the incommensurability of criteria: The above three characteristics can be combined in the see-move-see cycle. The designer "sees" the evolving design in one of many ways of seeing. The design is appraised in terms of *potentially many* criteria associated with the different ways of seeing. A move is then made which is expected to improve the design from the point of view of *one* criterion of appraisal (or one combination of criteria in the case of "simultaneous" moves). The criteria in question are incommensurable. Although two designs can be compared according to the degree to which they satisfy a given criterion, it is not in general meaningful to compare the degree to which a design satisfies criterion A, with the degree to which it satisfies criterion B.

Chapter 2 elaborates these four points. Related to them are the following two methodological considerations for designing design systems:



a. Principle based vs. rule based systems: A design system is principle based rather than rule based, to the extent that it captures design knowledge explicitly and in a modular fashion. This is in contrast with rule based systems which tend to be based on rules capturing purely syntactic characteristics of the end-product and obscuring the knowledge inherent in them. Of course there is no such thing as purely rule based systems, and systems which are predominantly rule based can still capture aspects of design knowledge. But the differences between the two approaches can have important consequences as shown below.

b. The manifest nature of design documents<sup>1</sup>: Not all the intentions and constraints resulting in the creation of components (and their relationships) in a design document are made explicit in it. A computational system which compensates for this apparent shortcoming by extensive annotation and constraint management, runs the risk of losing the indispensable immediacy of the designer's interaction with the document. Due to the overwhelming ubiquity of constraints, the designer needs not only the ambiguity of a document (an intersection of lines can be a cross or two L shapes) but also the arbitrariness inherent in it (a line which could satisfy the relevant constraints by being anywhere within a range of locations, is actually placed in one particular location and the designer's subsequent interactions with it are, at least in part, a function of that particular location).

These two points are covered in chapter 3, in the context of a critique of other approaches to design generation, namely shape grammars and constraint satisfaction or optimization techniques.

Chapter 4 introduces the discursive generator giving its functional details and presents its application to a simple design task of arranging blocks according to some formal principles. For each major decision made in the design of the discursive generator, and for the major characteristics of its behavior, the connection is made with the principles established in chapters 2 and 3.

---

<sup>1</sup> The term "document" is used here and throughout the thesis in a very general sense, referring to any object which is created, transformed and referred to during design activity, from sketches to detailed drawings to design artifacts themselves.

Chapter 5 raises a number of points which can form the basis for future work on the discursive approach to design generation.

By way of preview, let us now look at some excerpts from the output of the block arrangement application presented in chapter 4. The task is stated in terms of arranging three blocks of arbitrary dimensions according to some simple principles of massing having to do with maximizing alignments and abutments, favoring compact arrangements and visibility from a given perspective, as well as trying to maintain a constant volume-to-footprint area ratio. Figure 1.1 shows seven moves made by the discursive generator during a session which will be presented in more detail in chapter 4.

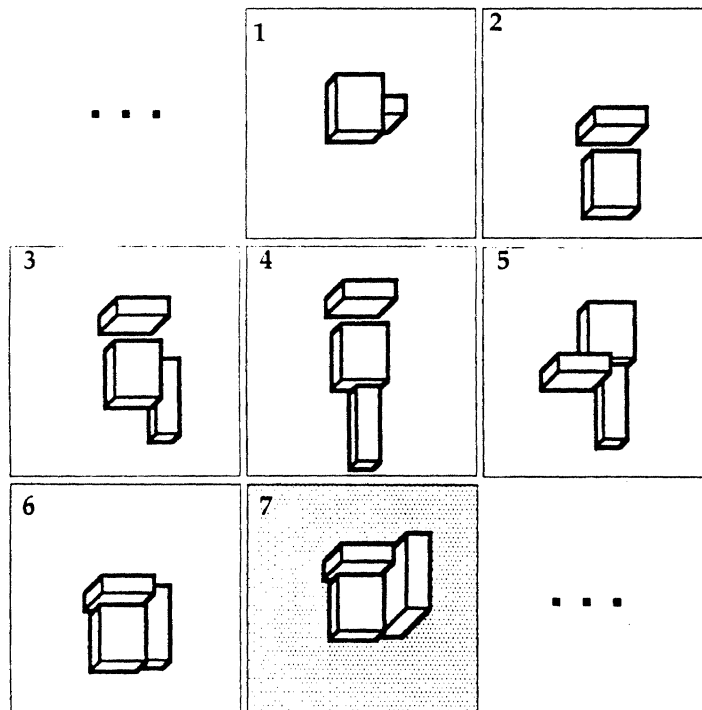


Figure 1.1

In frame 7, after an adjustment of heights, the generator achieves a configuration which can be considered satisfactory in terms of the stated principles. When the desired number of blocks is raised to five, an overconstrained situation results in which a satisfactory configuration is difficult to define. Figure 1.2 shows an excerpt from such a session where the highlighted frames indicate configurations satisfying some reduced requirements.

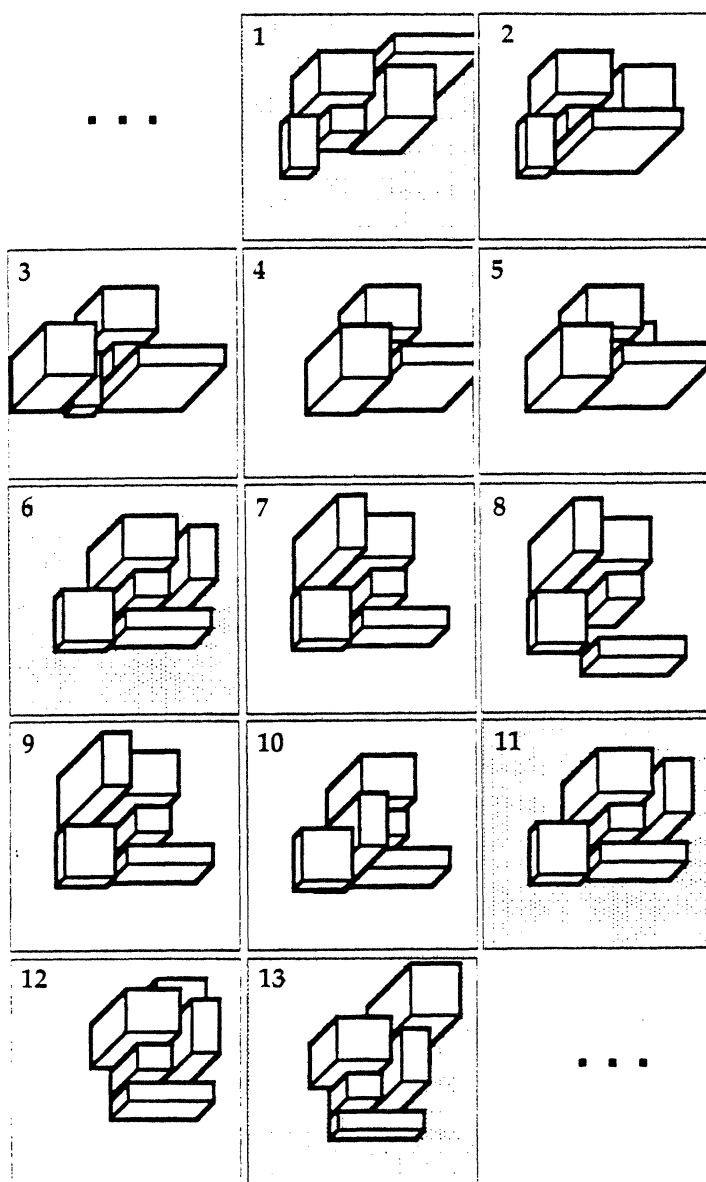


Figure 1.2

Frame 1 represents such a case, as does frame 6 which could be considered an optimal configuration. Two interesting characteristics of the discursive generator's behavior are illustrated in this excerpt. The first is its ability to make a transformation which results in a "worse" configuration which could lead to an improvement in subsequent moves. The transition from the favorable configuration in frame 1 via the one in frame 4 (possibly the worst configuration in this 50-move session) to that in frame 6 (the most favorable in the entire session) is an example of that ability. Another feature is the ability to produce repetitions (which can be considered beneficial for exploration in such overconstrained cases) which do not, however, degenerate into prolonged oscillation. This feature is all the more significant when one considers that no randomness is involved in it. Two examples of this kind of behavior occur in this excerpt. In frame 11 the configuration of frame 6 is reached through a different set of transformations than the ones leading up to frame 6, and the system continues with moves different from the ones which occur following frame 6. In this same sequence a smaller-scale cycle of oscillation is also avoided when the A-B-A repetition of frames 7, 8 and 9 are followed by a new move in frame 10.

The aim of this thesis is to illustrate a new approach rather than to prove some points or to develop a practical application. Accordingly, the theory of design presented in chapters 2 and 3, is not supported by a rigorous process of demonstration, and many points are made with the help of examples thought experiments. As for the application of arranging blocks using the discursive generator, not only is it one of the simplest possible ones which adequately illustrates some important features of the discursive architecture but it is itself based on a set of functionalities which are just sophisticated enough to work for the specific task in question.

Note that although the treatment of design in this thesis is meant to be general -dealing with aspects of design activity which are valid across fields, from graphic design to mechanical engineering design- the specific examples chosen for illustration will almost always be architectural for reasons of convenience.

## 2. Some Characteristics of Designing

In this section I will try to analyze those characteristics of designing which are the basis for the "discursive" model of design activity presented below. I will emphasize the basic opportunistic nature of designing and the role of focus, of distraction, and of multiple ways of seeing a design document or artifact. I will then propose the "see-move-see" cycle as a basis for the proposed dynamic model, contrasting it with other notions of design cycles, such as the generate-and-test cycle and the successive refinements cycle. The idea of the incommensurability of criteria and the evidence for a dynamic interaction between acts of evaluation and generation will be used to argue against more traditional notions of design cycles.

### 2.1 Focus and the Opportunistic Gaze

It may seem, at first glance, that the basic driving force behind a progressing design is a set of goals (long-range or short-range ones) that a designer is trying to achieve at any given time. But I will claim that a fundamental substratum of design activity is an ongoing *opportunistic* quest for situations where a potential value (not necessarily related to explicit criteria) can be brought out, can be exploited, through an intervention on the part of the designer. The apparent goal-directed behavior of the designer is achieved by focusing one's opportunistic activity both in the physical space of the artifact (by working on certain parts of the design at a time) and in the space of intentions, implicit or explicit, associated with the given design task.

To repeat, I have suggested that:

1. An evolving design is evaluated in terms of any number of implicit criteria, in addition to the stated requirements of a design.
2. This evaluation is dynamic in nature and is based on finding, in the evolving design artifact, opportunities to bring out potential values.

This model of design activity can account for several common phenomena which traditional approaches either ignore, or delegate to the realm of exceptions. Consider, for example, the simple fact that with an apparently identical set of requirements, different designers (or the same designer) almost always propose varying designs. In the case of the three proposed plans for the St. Peter Cathedral in Rome shown in figure 2.1 [Fletcher 1975] (by Bramante, Michelangelo and Sangallo, respectively) the variations among the proposals cannot be accounted for by reference to the divine model on which they were based, or on any other explicit criteria. As Tzonis and Lefaivre point out, the classical canon admits some variation in that it is proscriptive, not perscriptive. "By constraining rather than directing, the classical canon allows for a certain degree of freedom and invention in responding to those forces of change that lie outside the world of forms" [Tzonis and Lefaivre 1987]. When one examines traditions of design where this "degree of freedom" is much greater than that afforded by the classical canon, the need for a dynamic model becomes even more evident.

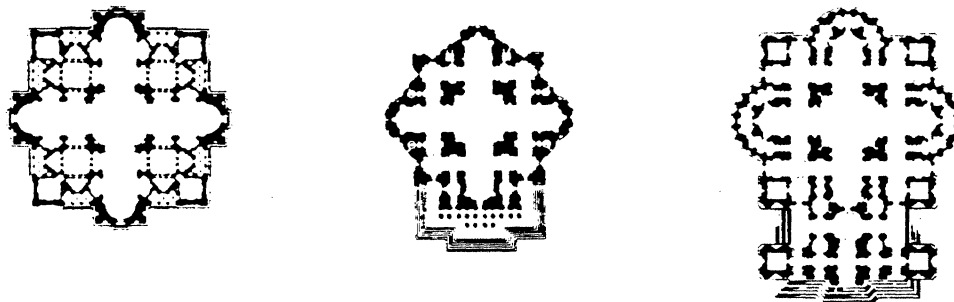


Figure 2.1

Another phenomenon typically avoided by traditional models is that of the violation of explicit criteria. Consider the case of an architect commissioned to design a residence with three bedrooms and one guest bedroom, and who presents to the client a design which omits the guest bedroom. Not only is this a possible scenario, but it is also possible that the client will happily accept the design, for the same reason that the architect proposed it. Not because it was impossible to meet all the client's requirements and the guest bedroom was the "optimal" sacrifice. But because the designer constructed a system of

criteria and an artifact which satisfied them in a coherent and interesting manner. The guest bedroom was sacrificed for the sake of some discovered system or set of values with which it was incompatible. John Archea, in an essay entitled "Puzzle-Making: What Architects Do when No One Is Looking," acknowledges the importance of implicit criteria, stressing the role of originality in the design process:

Instead of specifying what they are trying to accomplish independently of and prior to their attempts to accomplish it, as problem-solvers do, architects treat design as a search for the most appropriate effects that can be attained in specific spatio-temporal contexts which are in virtually all aspects *unique*. A key point of demarcation between puzzle-making and problem-solving is the former's over-riding emphasis on uniqueness. [Archea 1987]

But I claim that this aspect of "puzzle-making" does not apply only to original or unique designs. It applies to all design activity, from the mediocre to the sophisticated, and across fields, from visual arts to mechanical engineering. The self-conscious search for uniqueness, just like the goal directedness of design are higher-level<sup>1</sup> phenomena than what I have called the opportunistic substratum. And not only are explicit requirements not the only ones which are important in designing but, in principle, no constraints (other than metaphysical ones) are *absolutely* important for a designer. Yes, a design for a sky-scraper may be rejected by the building department because it violates zoning laws (typically sited as examples of inviolable rules in architectural design), yet we cannot avoid calling the process which resulted in the proposed building a process of design. And, of course, the design may even be accepted and the building built on the basis of negotiations or oversight. Or, to take an even more extreme case, a mechanical engineer can build a pump which does not work as a pump because it ignores certain laws of physics. Yet the process which led to the (failed) pump being what it is, may

---

<sup>1</sup> By using the term "higher-level" I do not mean to necessarily advocate a hierarchy of models of design activity. I simply mean that the role of opportunism is more basic than particular characteristics of expertise in design, just as a sense of rhythm is a more basic part of dancing than what it takes to tango. A theory of the tango does not *have* to be built strictly "on top of" a theory of rhythm, but it must somehow account for the role of rhythm. (Thus, a purely syntactical account, such as "it takes *two* to tango" cannot be considered a complete theory.) On the other hand, studying rhythm while postponing a treatment of the finer points of doing the tango may be a valid method of research.

well involve much more design than the act of putting together a (working) pump according to some conventional recipe.

Of course the path followed by this dynamic process is typically not arbitrary or guided by the designer's subconscious, and that is where *focus* comes in. At different stages in the history of a design session, the designer can attend to different aspect or components of the design and can analyze the design from a different point of view. What is particularly interesting is the pattern of shifts of focus and the mechanisms behind such shifts. I will not attempt here to analyze, in general terms, patterns of shifts of focus. I will simply point out, that the dynamic model of designing suggested above naturally accounts for an important, but often neglected cause of shifts of focus: distraction. Of course, distraction typically has a pejorative connotation, but it is essential for exploiting unforeseen opportunities which appear in an evolving design as a result of transformations made in the context of some plan of action (from which the designer is now distracted). Just as no constraint can be absolutely binding in a design, similarly, no evaluative criterion can be absolutely dormant, completely "out of focus." Speaking, for a moment, in behaviorist terms, we might say that if a stimulus in the evolving design artifact is important enough for a given evaluative criterion, it will "trigger" that criterion into focus. To give a somewhat simplistic example, imagine an architect who, in his repertoire of principles of formal organization, has that of anthropomorphic shapes. (Figure 2.2 shows a top view of Le Corbusier's project for the Centrosoyus in Moscow [Le Corbusier 1987].)

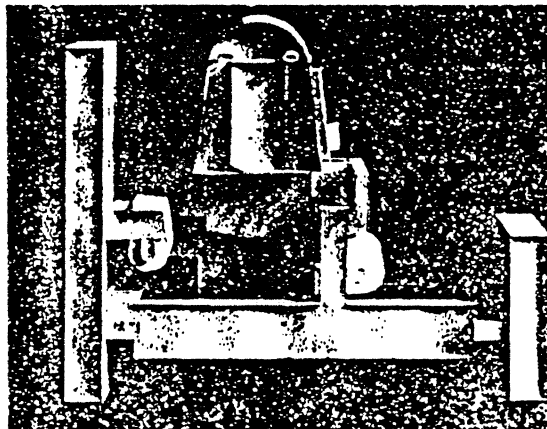


Figure 2.2



While working on the layout of a complex of buildings, paying particular attention to some technical aspect of their arrangement, the architect may "suddenly notice" that the layout would look exactly like a human body if the conference hall is brought into alignment with the gymnasium. It is quite possible that the architect will proceed to make that aligning move, and even to work some more in the context of anthropomorphic shapes.

## 2.2 Appraisal and Know-how

One does not have to be a designer in order to evaluate an artifact in terms of some criteria. And conversely, the ability to evaluate a design does not automatically lead to the ability to design. Hence the inherent strangeness of design education, and the difficulty of automating what designers do. In *The Sciences of the Artificial*, Herbert Simon complains that:

Engineering Schools have become schools of physics and mathematics; medical schools have become schools of biological science; business schools have become schools of finite mathematics. The use of adjectives like "applied" conceals, but does not change, the fact. It simply means that in the professional schools those topics are selected from mathematics and natural sciences for emphasis which are thought to be most nearly relevant to professional practice. It does not mean that design is taught, as distinguished from analysis. [Simon 1981]

Simon attributes this trend to the fact that engineering schools "hanker after respectability" by emulating their more scientific counterparts within universities. But an equally important factor may simply be that analysis is more easily taught, more readily captured in general rules, than synthesis. In architecture studios, for instance, one regularly sees instructors making use of esoteric vocabulary or, in the manner of the zen master, of very indirect discursive techniques to convey the often subtle mapping between proscriptive (evaluative) principles and prescriptive (generative) heuristics.

Obviously, the designer's skill lies more in the ability to find the design strategy which will result in an artifact having a desired quality, rather than in deciding what qualities are the best ones to aim for. Although, in general, the relationship between evaluation and generation is not well defined, there are particular cases where it is. If proscriptive formulations such as zoning regulations or an esthetic canon are highly restrictive, they tend to have a generative effect. The attempt to satisfy the rules becomes a search for possible "solutions" in the space of acceptable designs. Furthermore, the process of search can be replaced with the application of a recipe for achieving an acceptable design. The heuristics outlined in the Ten Books on Architecture by Vitruvius are an example of relatively flexible recipes for generating

designs having certain qualities and conforming to certain typological constraints [Heath 1989]. Another way of softening the proscriptive principles/prescriptive rules dichotomy is by shifting part of the effort of finding appropriate design solutions to the evaluation stage. This can be done by devising criteria for evaluation which are normative enough to suggest partial solutions for the generative phase. In his *Towards a Non-Oppressive Environment*, Tzonis writes the following about church design in the Renaissance:

The rules of architecture were expected to establish the link between the design product and its divine model. Therefore architectural investigations were aimed toward accomplishing two tasks: the identification of the structure of the divine model, and the invention of means of implementing it in the architectural products. A design product is "true" or "harmonic" or "perfect" if it is "according to measure," if it complies with the sacred prototype. [Tzonis 1972]

Of course normative criteria are not necessarily divine in nature. A simplistic functionalist approach to design can achieve the same effect as a metaphysical one by equating the evaluation of a design product with some aspects of its functional performance [Herberg 1983]. If the design of a building is considered as good as the efficiency of its circulation pattern, then it can be hoped that an efficient circulation pattern can dictate formal aspects of the building.

But the application of a recipe or the use of normative criteria are not sufficient for producing good designs in general. A powerful theory of design activity must somehow account for the interaction between *appraisal* -the designer's ability to evaluate an artifact- and *know-how* -the faculty of coming up with appropriate moves for beneficially transforming an artifact. This point, in itself, may seem obvious. After all, even in the trivial case of choosing a finished design out of a catalog -an act which consists almost exclusively of evaluation- there is a vestige of know-how if only in having adopted the catalog-lookup method. But what I will try to show is that a general model of designing must be able to accommodate an arbitrarily fine scale of interaction between appraisal and know-how. Catalog-lookup of finished products may be design, but it cannot be a general model of design

activity, simply because it cannot account for the diversity of design products. Analyzing a good design, or listening to some designers describing their work, one may be led to believe that the final product is simply the successful result of an attempt to satisfy a set of a priori intentions and requirements. But observing a designer in action clearly reveals the opportunistic nature of design processes, and the more subtle interdependence of appraisal and know-how.

## 2.3 Multiple Semantics

In a previous experiment (presented by Josep Maria Fargas and myself at the MIT Design Research Seminar under the title "Disposable Metaphors") designers were shown, on a computer screen, randomly generated pictures consisting of a frame, two lines and two rectangles (see figure 2.3.a). The design task was to make the picture "more stable" by displacing the rectangles. No clarification was given about the meaning of the term "stable"<sup>2</sup>. The computer would record the designers' moves, and later produce a real-time replay or a dynamic record of the process as shown in figure 2.3.

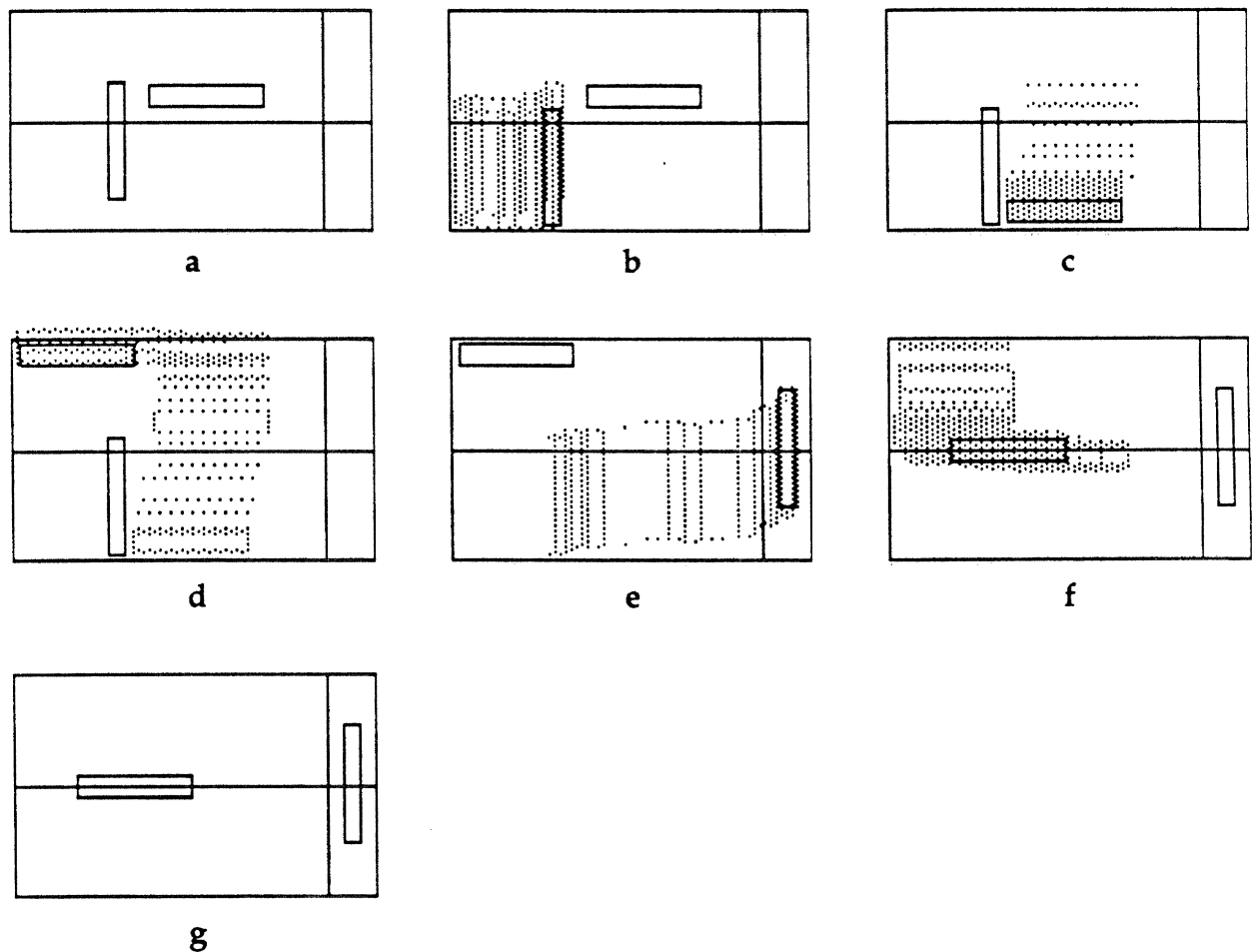


Figure 2.3

<sup>2</sup> Later, a computer program called EstheR (the Esthetics Replicant) was developed to replicate one particular design session of the Disposable Metaphors experiment, using a knowledge-based system.

Protocol analysis revealed that each designer would make use of a set of metaphors in order to attach a meaning both to the elements of the picture and to the term "stable" in the problem statement. Seen through what we called a *gravity* metaphor for instance, the rectangles would become "blocks," the bottom of the frame would represent a "table-top" and "stable" would come to mean "resting in equilibrium." The same designer, in a given session, would often shift metaphors (adopting, for instance, a *perspective depth* metaphor where a smaller rectangle would be considered as being "farther back") and would occasionally return to metaphors which were previously abandoned. These different ways of seeing the design document seemed to be the primary driving force behind the designers' generative moves.

Many designers and researchers have argued that the means of representation (such as line drawing) used by a designer influences the way the progressing design is perceived. In that sense, a representational medium would constitute a kind of cognitive filter through which the potential design artifact is "seen". But ways of seeing cannot simply be a function of the medium used, or even of a given representational system. Stiny has recently redirected the attention of the computer aided design community to a specific instance of the multiplicity of seeing in design, having to do with picking out different shapes from a potentially ambiguous drawing<sup>3</sup> [Stiny 1990]. Furthermore, the designer can explicitly represent an object in more than one way, by producing, for example, different kinds of sketches. Peter Rowe points out the subjective nature of such representations:

Referential sketches, for instance, often have an idiosyncratic, notational quality about them. They are the 'marking' of concepts and conceptual developments, rich in meaning to some but meaningless to others.[Rowe 1987]

But seeing a progressing design in terms of a particular semantics, through one kind of "filter" or another, does not necessitate an explicit representation in those terms. The designer does not have to draw construction lines in

---

<sup>3</sup> See below, in section 3.2.2 ("The Manifest Nature of Design Documents"), a discussion of the importance not just of the *ambiguity* of a document, but of its *arbitrariness* as well.

order to pay attention to the alignments of certain components. In fact, some of the most powerful insights that a designer has are not mediated by explicit representations specific to the type of seeing involved. Of course, the designer may actually make some sketches "marking" the insight for the benefit of subsequent stages of the process (or for posterity) but, formally speaking, these can be considered special annotations of the design document.

Thus the designer, over the history of the development of a design, sees the evolving document in potentially many different terms<sup>4</sup>. The advantages of this multiplicity are quite general, as illustrated by this sentence of Cortazar's: "Sometimes it helps to give a lot of names to a partial vision, at least it prevents the notion from becoming closed and rigid." [Cortazar 1966] or by Valery qualifying the word "drama" which he has just used: "Drama, adventure, agitations, all words of this kind can be used, on condition that they are many and correct one another." [Valery 1919]. The mechanism of multiple ways of seeing which are unmediated by explicit representations involves an arbitrarily fine scale of interaction between evaluation and generation. In that context, the concept of the exclusivity of ways of seeing (over an arbitrarily short period) becomes important. Famous examples (Wittgenstein's Rabbit-Duck drawing, the Necker cube, etc.) illustrate the fact that a given interpretation of an image cannot simultaneously be perceived as one thing and another. Although one can succeed in seeing, in the same drawing, a rabbit, then a duck, then a rabbit, etc., it seems impossible to see both at the same time. This principle of exclusivity can be a powerful tool for abstraction in models of the design process. We can consider that a designer, at any given (arbitrarily brief) period of time is attentive to one aspect of a design or another.

At least two major objections can be raised to this principle of semantic exclusivity. One objection has to do with the fact that designers almost never treat a project from a single point of view. The second objection is based on

---

<sup>4</sup> This potential multiplicity of ways in which the artifact or document is seen as the designer interacts with it should not be confused, on the one hand, with notions of alternative *explicit* representations (sometimes referred to as multiple worlds [Veerkamp 1990]) or, on the other hand, with the idea that a multiplicity of knowledge sources or disciplines cooperate in a design [Pohl et al 1991].

the observation that good designers are particularly adept at making design moves which simultaneously satisfy more than one evaluative criteria. The concerns inherent to the first objection are easily reconciled with the principle of exclusivity by combining a "multiple semantics" model with the concept of "distraction" as defined above. The idea can be illustrated by saying that the designer potentially "sees" the drawing in many different terms, each of which is associated with an evaluative criterion. At any given time, all but one (or a given combination) of these ways of seeing is latent, in the sense that they are not the driving force behind the generation of design moves. However, each of them (or some combination of them) has a chance of coming to the perceptual forefront, so to speak, after every transformation of the design document. Each of them can distract the designer's attention away from the current way of seeing. In such a model, there is no contradiction between the potential power of each way of seeing, and the idea that focus, by definition, involves the exclusion of certain aspects or elements in order to give more importance to others.

A similar reconciliation is possible with the second objection, which has to do with the existence of "simultaneous" moves. Indeed the importance of these kinds of design moves has dictated many of the considerations behind the model of designing proposed here.

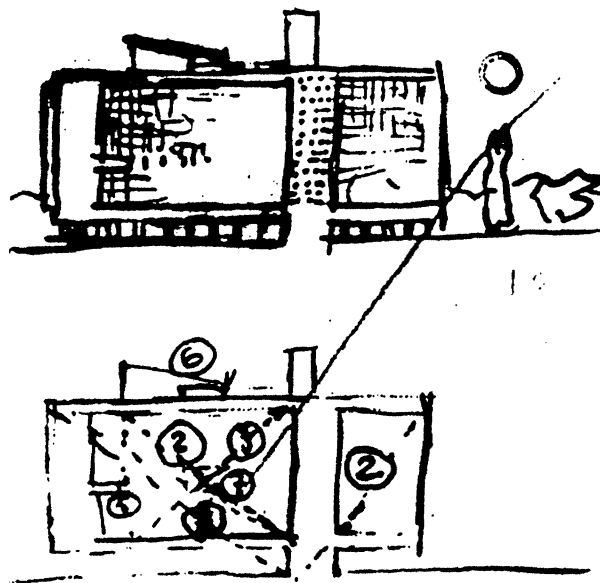


Figure 2.4



Moves having a simultaneous nature are possible when subjective transformations dictated in the context of two (or more) ways of seeing correspond to the same objective transformation at the level of the document or the design artifact. To give a specific example, let us say (at the risk of oversimplification) that at a certain stage of a design session the designer potentially sees a building's facade, on the one hand, as a set of rectangles conforming to some normative system of proportioning and, on the other, as a variety of textures judged from the point of view of some tacit notion of visual balance. If the elongation of one of the rectangles which was too short in the system of proportioning and the increase in area of one of the textures whose effect was not significant enough in the "visual balance", both constitute (or can be achieved by) a certain displacement of one of the bays of the building, then such a displacement would be a simultaneous move. Figure 2.4 shows one of Le Corbusier's sketches for his Unité d'habitation at Marseille. The lower figure is a study in terms of the "tracé régulateur," one of the architect's proportioning systems [Le Corbusier 1987].

How can we insure that the proposed model incorporate a bias towards this type of simultaneity? To answer that question let us first introduce the concept of the "saliency" or the "immediacy" of aspects of a document in terms of each of the ways of seeing (or some combination of them) involved in a design process. The likelihood that some aspect of the design will distract the designer's attention away from current concerns is proportional to the saliency of certain aspects of the document in terms of the distracting consideration. If an attempt to make a facade symmetrical results in the elimination of the entrance to the building, then it is likely that the designer's attention will, at some point, shift to practical concerns of circulation and physical access. In general, the degree of immediacy of a concern is a function of several factors, such as how recently that concern was active, how critical or problematic the current state of the design is from that particular point of view, and so on. Favoring simultaneous moves can be achieved by recognizing that this immediacy is considerably higher for pairs (or groups) of ways of seeing which coincide at the level of their proposed transformations.

Of course the skill of the good designer lies as much in the ability to bring about situations where simultaneous moves are possible, as in seizing the opportunity to make such moves. I will argue that the former ability, and other sophisticated skills, such as ones involved in the related merit of function-sharing [Ulrich 87], analogy, and so on, belong to a higher-level<sup>5</sup> model and that they are designer specific. Function-sharing, for instance is the notion of one component of a design fulfilling more than one function. A cord by which a lamp fixture is hung from a ceiling can exhibit function sharing if, at the same time, it provides support for the lamp, and acts as the electric conduit for it. This can be considered as the component-level equivalent of the simultaneity of moves. But function sharing (like simultaneity of moves), although a powerful design feature, is not a general requirement of good design, and therefore should not be an *inherent* feature of a model of design activity. In fact what I will call *function segregation* is as pervasive a feature of good designs as function sharing. Where the latter provides component-level economy, the former provides an economy of effort and modularity. The functions of support and partitioning which are shared in a bearing wall, are deliberately separated in typical modern building designs. Both function sharing and function segregation are often used as esthetic features in design.

---

<sup>5</sup> See footnote 1 of this chapter.

## 2.4 The See-Move-See Cycle

Every model of the design process seems to incorporate some notion of a cycle. The proposed cycles vary in scale, ranging from that of the interaction of designer and document [Graves 1977], to the scale of product life-cycles [Simon 1981]. Most theories, such as the one outlined in Habraken's *The Appearance of the Form* [Habraken 1985] combine different kinds of cycles, or nested cycles at different scales.

Obviously some notion of iteration is needed if one's aim is to give a temporal or dynamic account of designing. Two questions one can ask, however, when evaluating a particular notion of design cycles, are the following: (1) Is the notion effective?: does it capture a fundamental aspect of designing on which it is possible to build a more detailed account? (2) Is it at the appropriate scale?: Does it avoid the over-generalization which can result from applying a phenomenon which takes place over a long period of time, to the more immediate setting of a design session? The literature on design studies is full of proposals of design cycles which seem to fail on one or both of these two counts. I will argue, in particular, against the tendency of many researchers to model the design activity which takes place at the most intimate scales as a "generate-and-test" cycle, or a cycle of "successive approximations".

In their paper entitled "Kinds of Seeing and Their Function in Designing," Schon and Wiggins describe a very simple but powerful version of the design loop which is particularly compatible with notions of opportunism and multiple semantics. They illustrate their account of this cycle -which I will call the "see-move-see cycle"- using the following statement made by Petra (the name given to the subject of a protocol analysis): "I had six of these classroom units but they were too small in scale to do much with. So I changed them to this more significant [L-shaped] layout". Schon and Wiggins write:

Petra's move begins with a particular way of seeing the first configuration, "six of these classroom units". Her way of seeing them involves a judgment of quality: she finds them "too small in scale to do much with". Hence she changes them to the L-shapes,

which she sees as as "this more significant layout". With her first visual judgment, Petra has set a problem: "too small in scale". She makes her move in order to solve this problem, and with her subsequent description, "this more significant layout", she expresses a second judgment, namely, that the problem she initially set has now been solved. Petra's judgments are embodied in acts of seeing. She sees that the six classroom units are too small in scale to do much with, and sees that the three L-shapes are more significant (clearly, she means to indicate that they are more significant in scale, whatever other significance they may also turn out to have). Her design snippet can be schematized as seeing-moving-seeing. [Schon and Wiggins 1990]

The following scenario, illustrated in figure 2.5, may help clarify the role of opportunism, focus and multiple semantics in the see-move-see cycle.

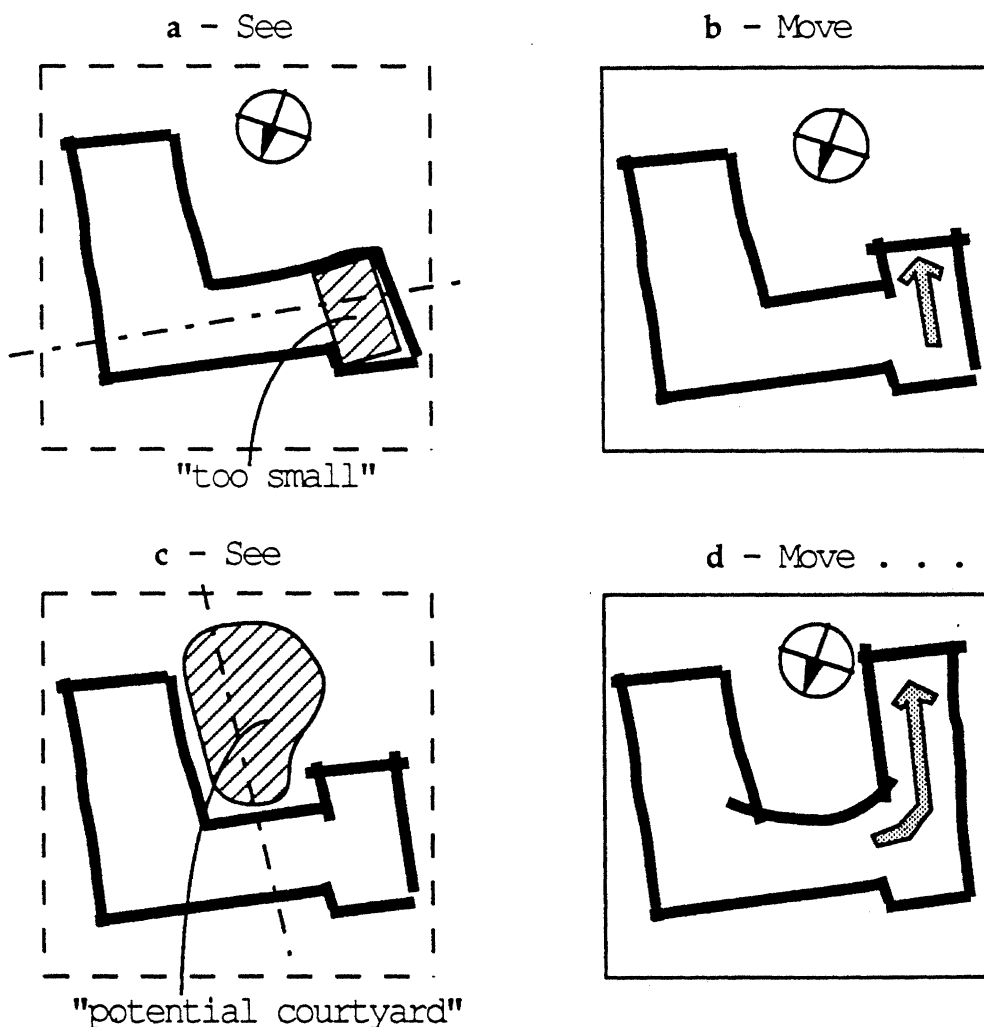


Figure 2.5

In frame a, the designer sees the layout of a building in terms of its programmatic requirements, and decides that one of the rooms is too small. In frame b, the designer extends one edge of the small room in order to enlarge it. As an unexpected result of this extension, an opportunity is seen, in frame c, to create a south-facing U-shaped courtyard by extending the right wing of the building, as shown in frame d. Figure 2.6 is a detail of a drawing by Farkas Molnár made during the design of his Red Cube [Klotz 1989]. Note the coexistence of different ways of seeing the evolving design, traces of which are recorded explicitly in the document.

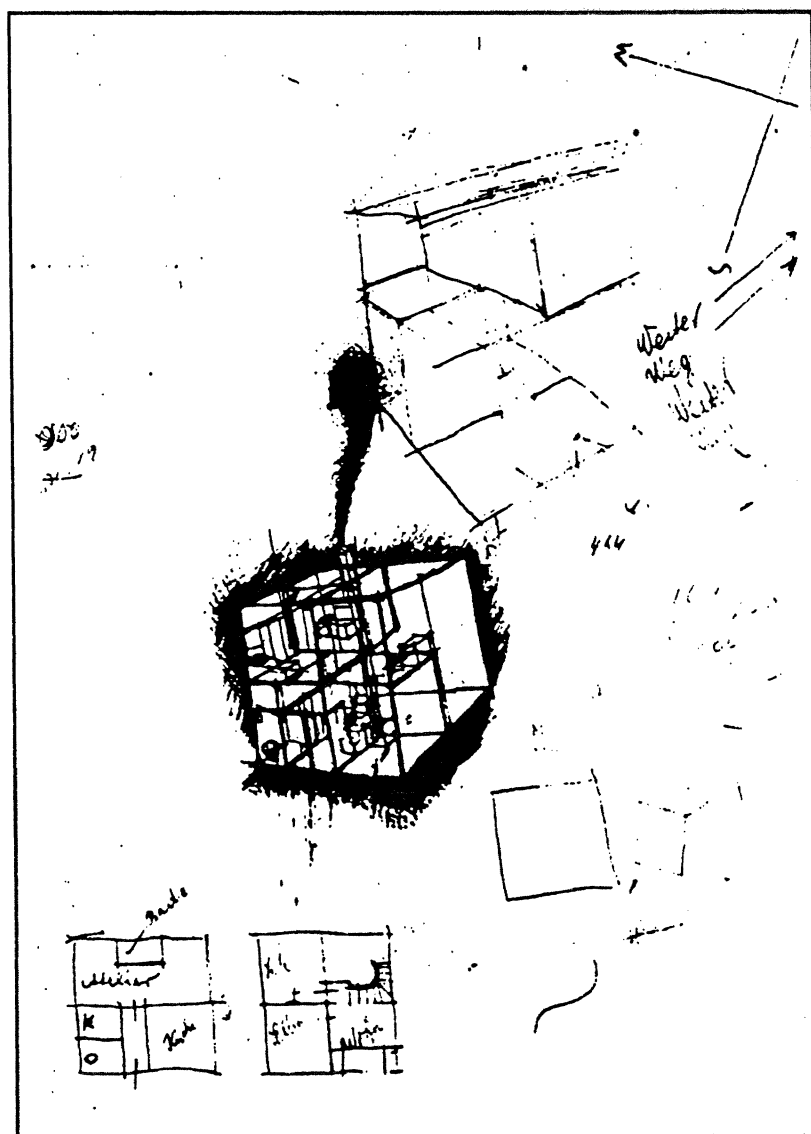


Figure 2.6

Notice how evaluation and generation interact in this model. The *appraisal* of the document in frame c involves a recognition of the potential for a courtyard. Inherent in this recognition is the designer's *know-how*: the knowledge of a move expected to create such a courtyard<sup>6</sup>.

The see-move-see cycle is not a generate-and-test cycle. Although in the see-move-see cycle there is a succession of generation and evaluation, the basic mechanism is not one of generating a design, seeing if it meets all the desired criteria (or to what extent it meets each of the criteria) and accepting or rejecting it on that basis. Rather, the idea is to evaluate the evolving design after each act of generation, *in order to generate the next move*. There is no testing. Although designers do backtrack, abandoning a whole path of exploration which turned out to be unsuccessful, they do not do so at each move. Of course there are tentative moves which are made and immediately retracted, but those are typically not retracted on the basis of a global test, but as part of an evaluation specific to the criteria which led to that move.

The see-move-see cycle is not a successive approximations or incremental refinement cycle. The major indication being that a move made with the expectation of improving some quality of the design, may leave it in worse shape in terms of some (or even every) other quality. Of course one would like to make only moves which result in improvements for every relevant quality of an artifact. But in the typically over-constrained and ill-defined domain of design tasks, requiring that every move be universally positive is counterproductive. Two particular problems associated with design systems based on successive improvements are: (1) the problem of getting stuck in local maxima, and (2) the problem of the incommensurability of criteria. The first is a well known drawback resulting from the situation where things have to get worse before they have a chance of getting better.

---

<sup>6</sup> This idea that know-how is somehow intrinsic in the act of appraisal is in some ways related to Wittgenstein's notion that expectations contain a "picture" of the thing expected: "Expectation is connected with looking for. My looking for something presupposes that I know what I am looking for, without what I am looking for having to exist." [Wittgenstein 1964]

The second is a more fundamental problem common to all design systems based on quantitative procedures for evaluation. It can be illustrated with the following perplexing question: Assuming that symmetry and the inclusion of a certain number of windows are equally important in the design of a facade, which is worse, an asymmetrical facade with the right number of windows or a symmetrical one with too many windows? What will the answer be if we assume, instead, that symmetry is actually twice as important as having the right number of windows, but that in the second case the number of windows is three times what it should be? Any system which must computationally evaluate the global current state of a design and compare it to some proposed state must answer those questions.





### 3. Using Computers in/to Design: A Critique of Current Trends

In the early 70's Bill Hillier and Adrian Leaman, in the introduction to an important paper entitled "How is Design Possible? A Sketch for a Theory", wrote the following:

The theories of design developed over the past decade in connection with 'systematic design', 'design method' and subsequently 'computer-aided design' do not in general have the merit of rendering the evidence about design 'nearly obvious'. On the contrary, they make it appear mysterious. For example, the syncretic generation of outline solutions in the earliest stages of design is made to appear illegitimate and undesirable on the grounds that any 'rational' approach to design must seek to generate the solution as far as possible from an analysis and synthesis of problem information and constraints. [Hillier and Leaman 1973]

Unfortunately, the leading trends in the use of computers in design still fail to account in a convincing way for much of "the evidence about design". In what follows, I will try to evaluate leading trends in computational approaches to design. I will avoid, as much as possible, basing this evaluation on a comparison with some notion of conventional design activity. Instead I will borrow, from the literature on epistemology, the concepts of *reliability*, *power* and *dynamism*<sup>1</sup>, as general measures of performance [Goldman 1986]. I will take these terms to mean the following:

- . **Reliability:** The degree to which a system produces "good" (or "acceptable") designs. Note that a system which always produces the same (good) design is completely reliable.
- . **Power:** The range of different (good) designs that a system can produce. A system which is capable of producing many different kinds of designs none of which are good designs is powerful, but not reliable at all.
- . **Dynamism:** Dynamism is a measure of responsiveness or spontaneity. A system may internally enumerate all possible designs and eliminate ones which are unsatisfactory. Such a system is reliable and powerful but it loses out on dynamism.

---

<sup>1</sup> I will use the term "dynamism" in place of Goldman's "speed" in order to avoid confusion with the more usual meaning of "speed" as used in a computational context.

### 3.1 Shape Grammars

"The *spirit* of his vows he made no scruple of setting at naught,  
but the *letter* was a bond inviolable."

-Edgar Allan Poe  
*Three Sundays in a Week*

Shape Grammars make use of "substitution rules" to build forms. Starting with a "Hall", we can substitute in its place either a "Hall with a Room behind it" or a "Hall with a Room to its right". For each of these, we can substitute a more complex figure, and so on. Figure 3.1 shows such an approach as it is illustrated in a paper on shape grammars by Ulrich Flemming [Flemming 1987].

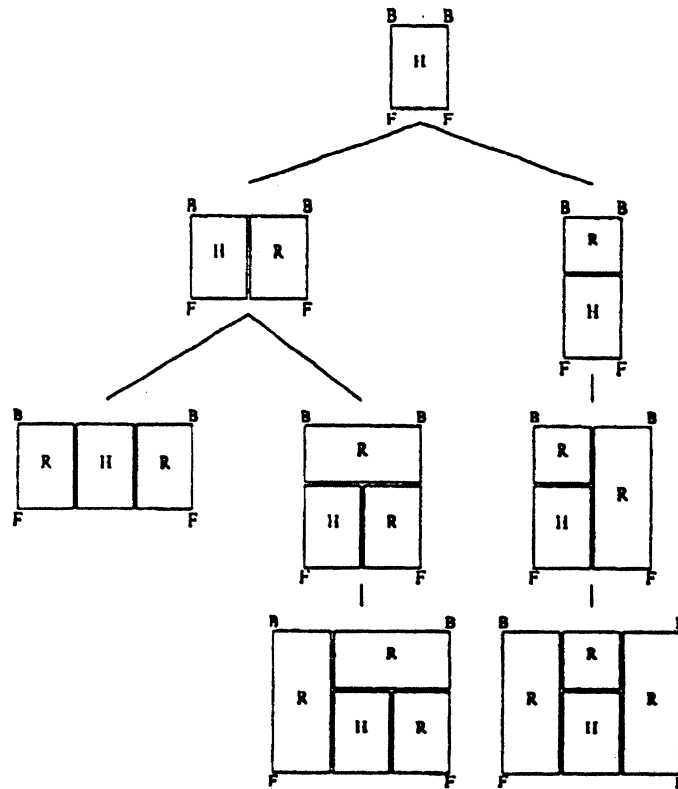


Figure 3.1

This apparently simple technique, when used with an in-depth understanding of particular types or families of designs, or when applied to designs based on a well-defined canon, can yield very impressive results. In

1978, Stiny and Mitchell presented a shape grammar which generated the ground plans of Palladian villas [Stiny and Mitchell 1978]. Many other grammars have been formulated for designs ranging from window lattices to Queen Anne houses. But in a critical paper on shape grammars, Aaron Fleisher questions whether their introduction into design is an occasion "on which miscellany is raised to systemics." [Fleisher 1990]

### 3.1.1 Reliability vs. Power

One benefit of writing a shape grammar, as anyone who has done it knows, is that the writer gains a very intimate understanding of the structure of the design in question. In an authoritative review of shape grammars, Flemming also makes that point: "While developing our grammars in both parts of the study, we were forced to look at examples with a degree of closeness that is hardly needed if the analysis proceeds in the traditional, intuitive way." [Flemming 1987]. Unfortunately the knowledge gained from designing a shape grammar is not captured in it. Once completed, the grammar may be able to produce a large number of artifacts, but it has no room for variations which, although unforeseen, could easily have been derived from the knowledge that went into building them. In that sense, shape grammars are *reliable* (they can consistently produce acceptable designs) but they are not *powerful* (the range of designs they can produce is very limited). In general, shape grammars rate well on *dynamism*. In principle, Their computations will yield legal configurations every time. Of course one can fail to capture all the motivations behind a design in purely syntactic generative rules (a necessary requirement in shape grammars). In such cases the practice has been to generate a more-than-complete set of alternatives and, in a subsequent step, evaluate them [Stiny and Gips 1978] possibly eliminating the unsuccessful ones [Galle 1981]. In *The Logic of Architecture* William Mitchell points out the importance of style in design.

An architect's knowledge of the shapes and materials of available elements and how to use them establishes a characteristic architectural language -a personal style . . . Without this, an architect attempting to design is like the scholars Gulliver encountered at the Academy of Lagado, who tried to write books

randomly combining words. That way, one would never get to the end. [Mitchell 1990]

This statement is a powerful reminder that knowledge of syntactic composition, whether it is captured in precedents, grammatical rules or in some tacit form is indispensable in design. Shape grammars are particularly efficient for capturing certain types of syntactic knowledge. But it is a bit too much to hope, as one might be tempted to do when observing the performance of some of the better grammars, that substitution rules alone can structure a design. An intuitive grasp of this fact can result from contrasting the history of almost any design session with the succession of steps that the execution of a shape grammar results in. But to put this evaluation on a more pragmatic footing, let us introduce the notion of "principle based" systems borrowed -like generative shape grammars- from linguistics [Berwick 1987].

### 3.1.2 Principle Based vs. Rule Based Design Systems

Many contemporary texts meant to be used by designers take the form of a compendium of functional standards [Ramsey and Sleeper 1956], examples of the use of different ordering principles of form and space [Ching 1979], or collections of prototype solutions [Markus 1968].

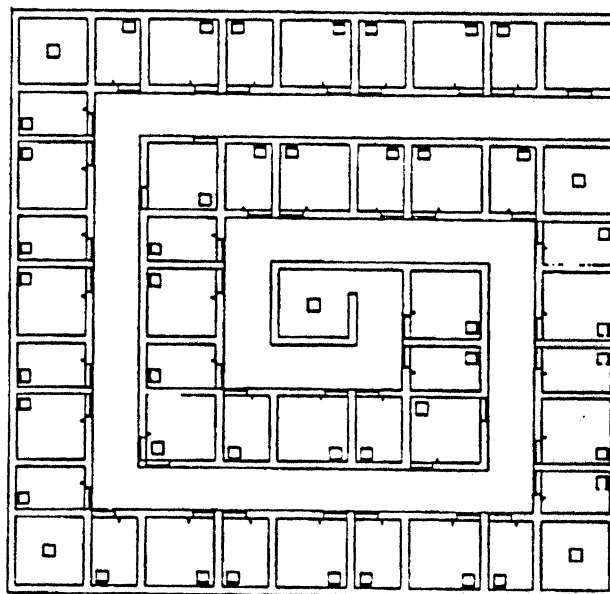


Figure 3.2

It is clear that these types of text, as opposed to ones advocating a specific methodology of design, are attempting to stress different *principles* to be abstracted from the given examples. An illustration of how the spiral can be used as a formal ordering principle will typically stress the abstract properties of a spiral, rather than the particular syntactic content of the example. Figure 3.2 shows the plan of a project by John Hejduk called "Solist-Labyrinth" [Hejduk 1986] which employs a spiral scheme. A design system which captures principles of appraisal and know-how at this level of abstraction will exhibit several concrete advantages over purely syntactic systems.

Consider a very simple illustration of style in the task of making a composition using rectangles. Figure 3.3 shows a few examples of such compositions which should be sufficient for forming some hypothesis of what the style consists of.

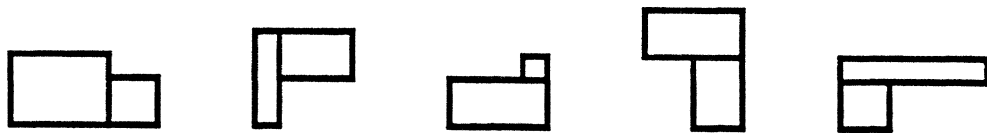


Figure 3.3

Of course, more than one hypothesis is possible. But let us assume that the size of the rectangles is arbitrary, and that the rectangles are always orthogonal, and let us propose a "grammar" by way of capturing this style of composition. The following are substitution rules for generating such compositions. At each step, the rectangle which is introduced is assumed to be of random dimension.

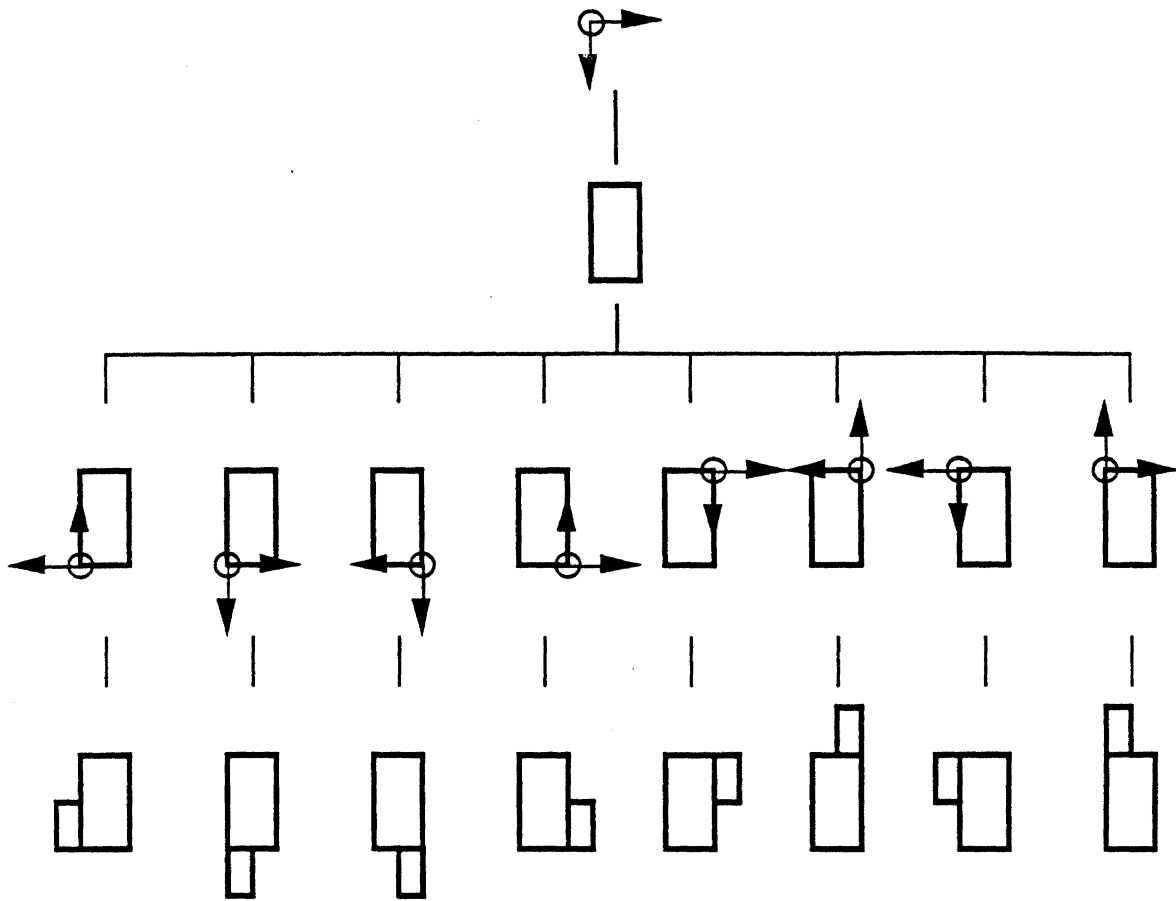


Figure 3.4

Note that this grammar is complete (i.e. it produces all possible compositions) if we assume that the rules for a syntactic evaluation of these rectangles are:

- (r1) The rectangles should abut at one (and only one) edge.
- (r2) One edge of one rectangle should align with an edge of the other.
- (r3) There should be one pair of rectangles.

The diagram in figure 3.4 represents what I will call the *rule based* approach. It is based on substitution rules which can be captured in the form of "If ... then ..." statements. A principle based approach would somehow capture some underlying principles, in this case principles of form and order, which would fit the examples given above. These principles might be the following:

- (p1) The rectangles should align and abut as much as possible.
- (p2) The rectangles should not overlap.
- (p3) The number of rectangles should be two.

Notice that this formulation modularizes the stylistic requirements which come into play. Alignment and abutment are grouped because they are related (abutment = alignment + adjacency), and the notion of overlap replaces the restrictions on the number of edges which should align or abut. The "incorrect" composition in figure 3.5.a does not respect the alignment/abutment principle. The one in figure 3.5.b has maximum abutment and alignment, but violates the no-overlap principle. As for the composition in figure 3.5.c, it has maximum alignment but minimum abutment.

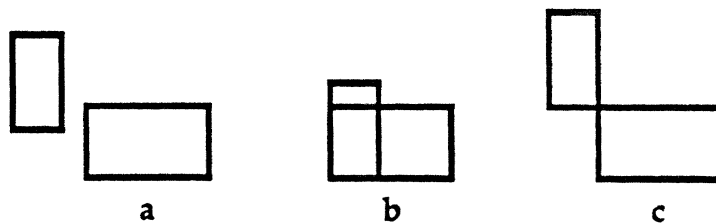


Figure 3.5

The apparent advantage of the rule based approach is that it has a built-in mechanism for generating all possible configurations. Elsewhere in this thesis (in chapter 4) there is an explanation of how to imbed principles such as p1 through p3 above, in a dynamic mechanism for generation, using appraisal and know-how. But let us simply assume for now that such a mechanism is possible. What are its advantages over the rule based approach?

1. The principle based approach is more *powerful* than the rule based one in the technical sense of the term "power" defined above. It can handle a larger family of designs than the rule based approach. For example if we replace rectangles with triangles as shown in figure 3.6.a, or if we increase the number of rectangles from two to three or more (figure 3.6.b), or if we

combined these two modifications (figure 3.6.c), then principles p1 through p3 can remain virtually unchanged, whereas the grammar illustrated in figure 3.4 will need substantial modifications.

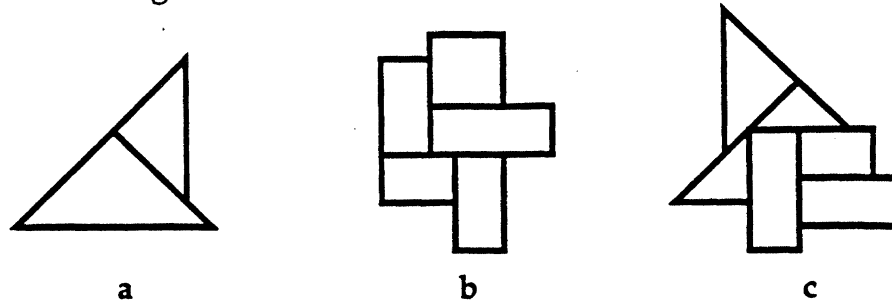


Figure 3.6

2. The principle based approach can handle the addition of new principles gracefully. That is, adding new principles or replacing or removing current ones will not require a substantial reformulation. Consider, for example, an additional requirement that the center of mass of the group of rectangles coincide with a given point, as shown in figure 3.7.

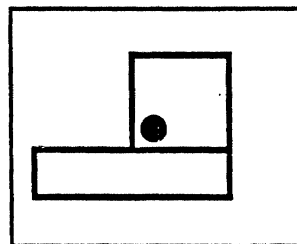


Figure 3.7

This addition would require a complete revision of the shape grammar above. For the principle based approach, it would just mean adding the faculty of appraisal which will recognize off-center configurations, and the know-how to simply move the rectangles in the appropriate direction to correct the problem. The principle based approach can even include conflicting principles. A simple example being the conflict between maximum abutment and overlap as illustrated by figure 3.5.b.

3. A principle based approach can account for exceptions and partially correct configurations without additional machinery. In the case of more than two rectangles, as shown in figure 3.6.b, not every two rectangles is



in a relationship of maximum abutment or alignment. Similarly the principle based system will recognize, for instance, that a configuration has good alignment properties, even though it exhibits some overlap. This is only possible in rule based systems if every possible type of exception is specifically accounted for. For example, the tree of substitution rules for the grammar would have to have a large number of additional branches corresponding to the generation of exceptions or partially correct configurations.

4. A principle Based approach, because it can deal with partially correct configurations, is appropriate for a dynamic model of design. Syntactic approaches such as shape grammars operate in terms of a rigid hierarchy of substitutions and transformations. Although their steps may sometimes correspond to the design moves in a strictly top-down process, they are generally too normative to capture design activity.

## 3.2 Designing with Constraints

"I am aware, on the other hand, that the case is by no means common, in which an author is at all in condition to retrace the steps by which his conclusions have been attained. In general, suggestions, having arisen pell-mell, are pursued and forgotten in a similar manner."

-Edgar Allan Poe  
*The Philosophy of Composition*

Problem-solving has often been used as a metaphor for designing. Design tasks are referred to as design "problems", and designs are sometimes called "solutions". Figure 3.8 shows an illustration from a paper on constraint-based layout design by Luis Moniz Pereira in which the problem-solving approach is adopted [Pereira 1978]. In the taxonomy of types of problems, "design problems" are ill-defined. There is no direct mapping from the problem statement to possible solutions. The least one can say is that the knowledge, the assumptions and the arbitrariness that a designer brings to a design problem are indispensable assets. But is problem-solving a good metaphor to use in such cases? Is it a good metaphor from the point of view of actually "solving" those kinds of "problems"?

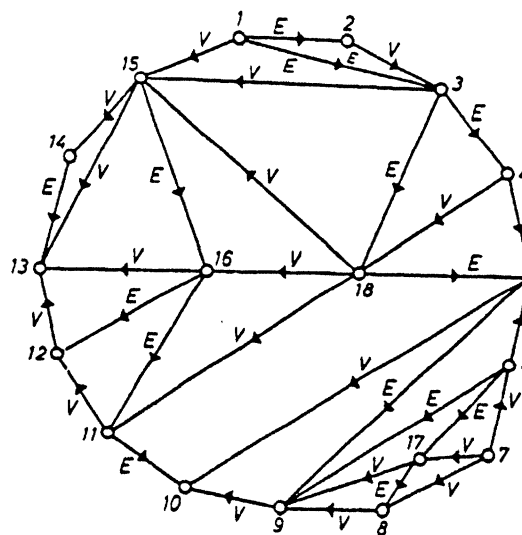


Figure 3.8

### 3.2.1 The Overwhelming Ubiquity of Constraints

Gross, Ervin, Anderson and Fleisher, in a paper entitled *Designing with Constraints*, point out that even if design is not problem solving, problem solving often occurs during design. The model of designing which they propose is meant to capture the process of defining a design task, proposing alternatives and, at some level, testing them.

We can describe a design problem or task as a collection of constraints and relations on attributes of the object to be designed. Then, to design is to describe constraints and to specify an object that satisfies all these constraints. [Gross et al. 1987]

One may argue that designing (even good designing) is not limited only to acts which lead to the satisfaction of all the constraints initially stated in a problem. But even if we assume that the final product of a design satisfies the explicit initial constraints, we are still left with the problem of characterizing design activity in a useful way. Certainly designing can be *described* in terms of constraints. We can track (or let us assume that we can track) all the constraining relationships among the components of a design artifact, registering, at each move, the transformations that they undergo [Rossignac, Borrel and Nackman 1989]. We can describe the designer's moves in terms of their effect on this constraint space, checking the resulting changes against our list of explicit requirements. But this kind of description is not a particularly effective one because design tasks are overconstrained (it is impossible to satisfy all the stated constraints at the same time) and also because constraints, as the authors themselves point out, are ubiquitous. What is interesting to observe in the way a designer approaches a problem is the selectivity with which constraints are considered relevant, ignored, and interpreted. Consider the following account by Rem Koolhaas transcribed in *The Chicago Tapes*, concerning the design of a villa at St. Cloud (see figure 3.9) [UIC 1986]:

Because the site was so small, it was very difficult to occupy it without ruining it. And as I said, the clients wanted a glass house, while at the same time imposing a lot of conditions that made a glass house near impossible. It was a difficult issue to resolve: the incredible weight of the swimming pool resting on this glass

pavilion. We did it by creating a three-story concrete wall, which had a cut-out representing the glass building. Inside the glass pavilion is the structure to support the weight of the pool.



Figure 3.9

In an underconstrained situation it may be a good idea to "solve for" the constraints and to find some configuration which satisfies them all. But what can one do in the usual overconstrained situation? One answer has been to delegate to a computer the task of "managing constraints", and to a designer the task of "designing with constraints". In other words, the designer is forced to work in an explicit constraint based paradigm and, at the same time, asked to unravel the tangle of constraints which typically appear after a dozen moves. More traditional approaches have tried to arbitrarily relax constraints, or find optimal (or pareto optimal) solutions which somehow satisfy all the constraints "as much as possible". Almost always, these approaches have failed to be very *reliable* or particularly *dynamic*.

In 1970 a team of researchers at the Massachusetts Institute of Technology headed by Timothy Johnson published a report on a system for "multi-constrained spatial synthesis" called IMAGE [Johnson et al. 1970]. IMAGE used an optimization algorithm to lay out rectangular spaces according to a set of constraints of varying degrees of importance. These constraints specified the relationships among rectangles from the point of view of overlap, proximity, alignment, relative position, visual access, and so on. The system would try to find the configuration for which the sum of the degrees of satisfaction for all the constraints (taking into account their relative

importance) was as high as possible. It was the first time that such a constraint satisfaction paradigm was taken to its logical conclusion. Although the IMAGE project explored many issues related to computer assistance in design, its main role (in retrospect) was that of an experiment in generating acceptable layouts based on constraints. In the underconstrained case which involved "the exploration of a large relatively unlimited solution space", the system sometimes produced suggestive layouts (see figure 3.10) and responded in an interesting way to changes in the relative importance of constraints.

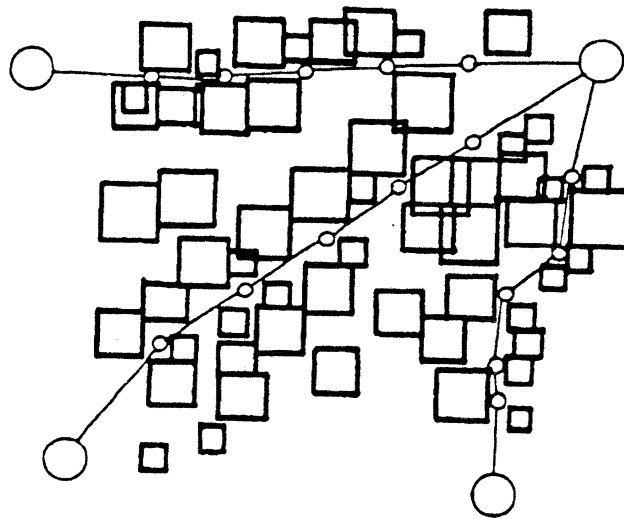


Figure 3.10

But in the more typical case where the solution space was "apparently small or nonexistent", its output was disappointing. The problems encountered by IMAGE are characteristic of constraint based systems. They are:

- . **Oscillation:** If a certain transformation of a design which takes it from state A to state B does not change its over-all value, then the system may follow up with a reverse transformation which brings it back to A. This cycle may be repeated indefinitely. Harder to control oscillations are ones which involve more complex cycles (A-B-C-D-A...).

- . **Getting stuck in local maxima:** Let us represent the different possible configurations of a design as different locations on a terrain where

altitude corresponds to the value of a configuration. If, through successive improvements, the system reaches a "hill" in such a terrain, then it will have to somehow make negative transformations to the design in order to come down the hill and have a chance of finding the "mountain". Many tricks can be used to deal with this phenomenon, but the fundamental problems behind it are: a) The impossibility of having a *global* view of the terrain (due to the nature of design tasks). And b) The fact that conventional constraint based systems operate in a *continuum*, meaning that discreet but principled jumps across the terrain (a common occurrence in design activity) are impossible for such systems.

. **Being overwhelmed by all those constraints:** The intuitive notion that there are too many constraints which have to be simultaneously satisfied is probably a valid objection to systems (even computational ones) which try to do just that. As already mentioned above, constraints are ubiquitous, and the skill of the designer includes the ability to concentrate on some at the expense of others at any given time. Many constraint based optimization systems unsuccessfully try to replace focus and opportunism with tolerance and compromise.

Despite the problems encountered by IMAGE, there was a great deal of enthusiasm throughout the 70's for the prospect of automated space planning, and the general approach to design automation associated with it<sup>2</sup>. Charles Eastman concludes one of his early papers on the subject with the hope that "As greater capabilities are developed for processing spatial

---

<sup>2</sup> In 1975, Guy Weinzapfel and Steven Handel presented a paper documenting an augmented version of IMAGE [Weinzapfel and Handel 1975]. but none of the fundamental shortcomings of the earlier system seemed to have been solved in the new version. Interestingly, a comment made by the authors in an appendix concerning the particular optimization algorithm used in IMAGE (namely the Least Mean Square Fit Relaxation method) provides a clue to the lack of basic progress over the five years since the creation of the program. While evaluating the algorithm the authors site, as one of its advantages, that "the method operates in a continuum." Ironically, in the same collection of papers, there is also an important one by Herbert Simon entitled "Style in Design" in which he argues for "satisficing" techniques as a more reasonable alternative to optimization algorithms working in a continuum, for particularly complex problems. (Strangely, however, Simon refers to Weinzapfel and Handel's work -among other authors' in the collection of papers- as an example of a satisficing technique!)

arrangement tasks, we may expect better analyzed, better resolved, and possibly even more beautiful physical environments to result" [Eastman 1973]. But in 1978 Max Henrion expressed the subsequent pessimism of researchers in the field in an illuminating overview of preceding work on space planning entitled "Automated Space Planning: A Postmortem?" [Henrion 1978]. Henrion concluded that existing systems failed to "fit happily into the design process", and that the incorporation of domain-specific knowledge was needed. The question remained, however, of how this design knowledge would be brought into play as part of a dynamic design system.

### 3.2.2 The Manifest Nature of Design Documents

The terms "intelligent drawings" and "relational modeling" have recently been used to express the idea of drawings with attached databases and constraints. Imbedded in such drawings would be not just the location of a wall, for instance, but also some trace of the intentions which placed it in that particular location. These traces can most readily take the form of relations between the wall and other objects in the design document. Although this is a very ingenious and practical idea for computer aided design documents, it goes against the spirit of how documents are used in the process of designing.



Figure 3.11

This may seem, at first, like a facetious statement to make. But what I will call the manifest nature of the design document and the unmediated character of the designer's interaction with it are essential ingredients, at least at some level, of any design process. Figure 3.11, which shows a site plan with a sketch superimposed on it by Tadao Ando [Klotz 1989], bears witness to this kind of interaction.

Christopher Alexander gives some amusing examples of what he calls "unselfconscious" designers reacting in an unmediated and direct way to their perception of the design artifact, and not having to manipulate it through some representation. He writes of the Eskimo redesigning his igloo in real time, as it were, as a result of perceived changes in temperature [Alexander 1964]. In light of this information, one may be led to think of the conventional ("self-conscious") designer as manipulating the thing being designed indirectly, *through* the design document, the drawing. But, as several writers on the subject have pointed out, the designer manipulates the design document, and not the projected artifact. The distinction between directly manipulating a design document and indirectly manipulating the projected artifact through the document may be subtle, but it has important consequences. Robin Evans highlights this distinction as follows:

It would be possible, I think, to write a history of Western architecture that would have little to do with either style or signification, concentrating instead on the manner of working. A large part of this history would be concerned with the gap between drawing and building. In it the drawing would be considered not so much a work of art or a truck for pushing ideas from place to place, but as the locale of subterfuge and evasions that one way or another get round the enormous weight of convention that has always been architecture's greatest security and at the same time its greatest liability [Evans 1986].

In the context of design constraints, the "enormous weight" Evans refers to can be interpreted as what I have called the overwhelming nature of simultaneous constraints. As for the "subterfuge and evasion", they are based on the necessary (but hopefully temporary) ignorance of certain commitments not captured in a drawing. This controlled ignorance is made



possible by the unconstrained immediacy of the document. The document is *ambiguous*, in the sense that it can be interpreted in different ways, as discussed in the section on multiple semantics above. But more importantly, and perhaps less obviously, the document is also *arbitrary*: a component may be able to satisfy some relevant constraints in any of several states, but it is captured in the document in only one of those states (or a limited family of them). When the designer appraises a document, the component in question contributes to this appraisal in the state in which it was represented, and not as a representative of its (overwhelmingly) many potential incarnations. To give a specific example, consider a line representing some boundary in an architectural drawing. The position of the line might have been anywhere within a given range. But in the context of a line-drawing representation, it can only be drawn in some default position. Although the designer can take into account the flexibility of this positioning, any immediate appraisal based on the drawing will take the line to be in the particular location where it is drawn.

The above distinction also accounts for the power of sketches in design. Even a non-annotated document, if it is drawn precisely, can entail more commitments than are desirable in the early stages of a design process. A sketch, with its valuable ambiguity leaves room for the designer to maneuver. Figure 3.12 shows a sketch by Mies van der Rohe [Bleau and Kaufman 1989]. Notice the potentially ambiguous articulation of the overlapping rectangles at the lower right.

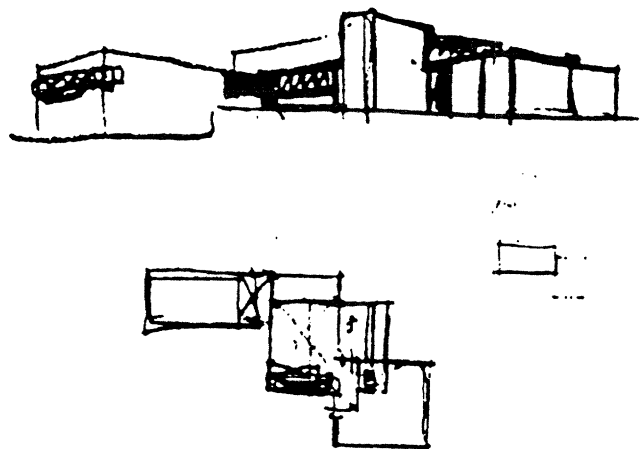


Figure 3.12

a sketch can also be conducive to discovery and invention in a very direct way. William Hubbard speculates about the role of incidental pencil marks in the inclusion of a certain ornament on a building's facade:

Some of these marks would be added "to make a better sketch," but others would be happy accidents -unintentional marks that were kept and carried forward because they made the sketch a better sketch. For example, it is interesting to speculate that the draftsman's practice of ending each line in a blob or tick mark might have been the source for this decoration on the extrados of the arch of the Crane Library [Hubbard 1986].

This issue is also important from the point of view of capturing design knowledge. Koutamanis argues for considering architectural drawings as the most appropriate representation for such knowledge, without having recourse to other more general means such as formal logic, semantic nets and the like [Koutamanis 1990]. It is interesting to note that the motivation behind this idea of the manifest character of design documents is analogous to that of Rodney Brooks in his insistence on using the world "as its own model" for the purpose of making autonomous robots interact with the real world [Brooks 1990]. The burden of symbolic representation can sometimes be too overwhelming at a fine scale of interaction.

## 4. A Discursive Architecture for Design Generation

In this chapter I will introduce a computational framework called the "discursive generator". This framework accommodates both the characteristics of design activity and the methodological principles of using computers in design which were discussed above. The characteristics in question are the following:

- . **Opportunism, focus and distraction:** Designing is based on a substratum of opportunistic activity. At any given time, the designer focuses on a limited number of components and evaluative criteria. When an opportunity is seen which can be exploited in terms of one of a large number of implicit and explicit values, the designer is distracted by it, and a shift of focus takes place. Focus can be a function of the history of a given design session, the bias of the designer, higher-level processes such as planning and inference and, of course, external factors.
- . **The interdependence of appraisal and know-how:** Appraisal is the act of evaluating an evolving design from the point of view of a given criterion, and know-how consists in the repertoire of moves expected to improve the state of the design in terms of a given criterion. The two are closely linked because part of appraisal consists of identifying a potential for improvement based on the knowledge of a possible move. The designer appraises a document not just in order to test it, but mainly to generate the next design move.
- . **Multiplicity of semantics and the exclusivity of seeing:** The designer can interpret a design document or artifact by attributing to its components and their relations one of many sets of meanings. At any given (arbitrarily brief) period, the designer can be thought of as actively relying on only one of those potential semantics. The equivalent of simultaneous interpretation in terms of two or more sets of meanings can take place when the design moves generated in terms of each of the semantics are equivalent.
- . **The see-move-see cycle and the incommensurability of criteria:** The above three characteristics can be combined in the see-move-see cycle. The designer "sees" the evolving design document or artifact in one of many ways of seeing. The design is appraised in terms of *potentially many* criteria associated with the different ways of seeing. A move is then made which is expected to improve the design from the point of view of *one* criterion of appraisal (or one combination of criteria in the case of "simultaneous" moves). The criteria in question are incommensurable. Although two

designs can be compared according to the degree to which they satisfy a given criterion, it is not in general meaningful to compare the degree to which a design satisfies criterion A, with the degree to which it satisfies criterion B.

As for the two methodological principles introduced in chapter 3, in the context of a critique of "shape grammars" and "designing with constraints", they are the following:

. **Principle based vs. rule based systems:** A design system is principle based rather than rule based, to the extent that it captures design knowledge explicitly and in a modular fashion. This is in contrast with rule based systems which tend to be based on rules capturing purely syntactic characteristics of the end-product, and obscuring the knowledge inherent in them. Of course there is no such thing as purely rule based systems, and systems which are predominantly rule based can still capture aspects of design knowledge. But the differences between the two approaches can have important consequences as discussed above.

. **The manifest nature of design documents:** Not all the intentions and constraints resulting in the creation of components (and their relationships) in a design document are made explicit in it. A computational system which compensates for this apparent shortcoming by extensive annotation and constraint management, runs the risk of losing the indispensable immediacy of the designer's interaction with the document. Due to the overwhelming ubiquity of constraints, the designer needs not only the ambiguity of a document (an intersection of lines can be a cross or two L shapes) but also the arbitrariness inherent in it (a line which could satisfy the relevant constraints by being anywhere within a range of locations, is actually placed in one particular location and the designer's subsequent interactions with it are a function of that particular location).

In what follows, I will give a detailed description of the discursive generator. For each aspect of its design, I will point out the ways in which it takes into account one or more of the six points above and what consequences this has on its behavior.

## 4.1 How the Discursive Generator Works

The discursive generator is composed of four main components: (1) the *forum*, (2) the *persona*, (3) the *arbiter* and (4) the *focus manager*. These

components interact in a see-move-see cycle: The persona "sees" the design document resident in the forum and produces a number of potential transformations. The arbiter chooses one of these transformations (or a composite of some of them) and applies it to the document as a design "move". At each cycle, the focus manager tunes the attention of the persona, both in terms of the document and in the space of current intentions or tendencies<sup>1</sup>.

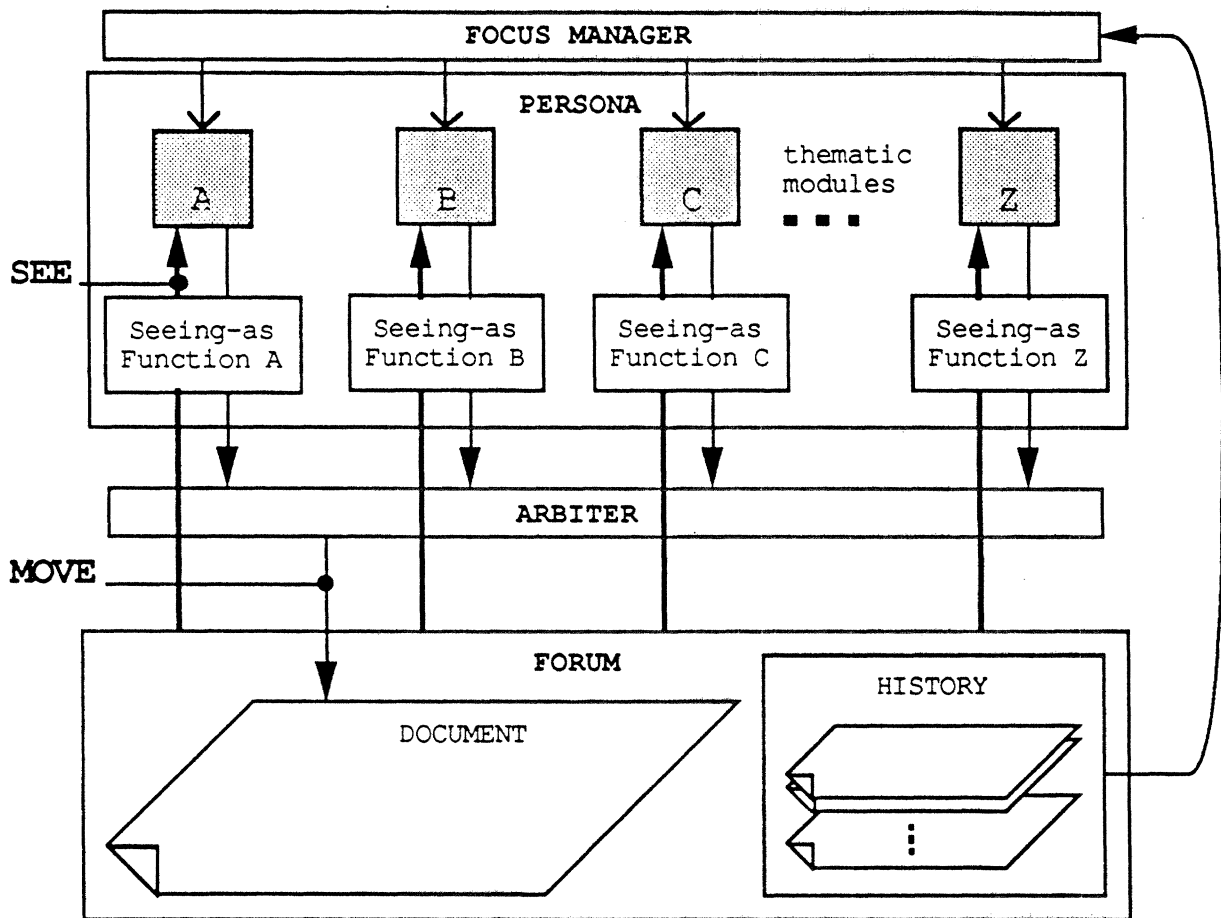


Figure 4.1

<sup>1</sup> In describing the architecture of the discursive generator and in naming its parts, I will make use of a mixture of two metaphors, the "animate agent" metaphor ("attention", "seeing" ...) and the "society" metaphor ("arbiter", "forum" ...). Both of these are quite common in the computer science literature. However, since this thesis does not attempt to model human cognitive or social processes but, at the same time, relies on such processes to validate and guide the theory on which it is based, the use of these metaphors may be confusing at times. Having said that, I will proceed to use them because of their convenience.

### 4.1.1 The Forum

One part of the forum is simply the collection of all the drawings, databases, annotations, artifacts, etc., which the designer refers to or acts upon in the context of designing. This part is called the "document". It is the locus of all the (formalized) interactions between the designer and the outside world. The other part of the forum is simply a convenient (but optional) abstraction on the document, namely its history. As a matter of convention, the persona only refers to the document (including any traces of its history which it has explicitly annotated) while the focus manager refers to the history proper in order to implement shifts of attention. The term "history" does not necessarily mean a complete record of the transformations that the document has undergone, but some special aspects and subsets of them (such as the most recent ones for instance). The history can also contain information which would not otherwise be recorded in the forum, such as some measure of the degree of "conviction" with which a transformation was made.

It is important to emphasize the following two aspects of the document:

1. *The document is the unique locus of interaction between the design system and the outside world.* It contains not only specifications of the thing being designed but any requirements associated with it as well. For instance the number of bedrooms which must be included in an apartment to be designed can be recorded only in the document. Furthermore, *any object in the document can be transformed or deleted by the design system.* The number of required bedrooms can be changed by the persona, just as easily -but hopefully not with the same likelihood as the shape of a bedroom. (Note that this is another difference between the document and the history, because the history cannot be transformed by the system). This feature is based on the principle that there are no absolute constraints in design activity, as discussed in section 2.1 ("Focus and the Opportunistic Gaze") above. Its result is to provide the flexibility needed at the level of the persona, and to transfer to it the responsibility of coherence and competence.

2. *No constraints recorded in the document are guaranteed to be binding across the system.* This is an extension of the first feature. It implies that not only are initial or explicit constraints not absolute, but that the persona itself cannot imbed constraints in the document which it will have to respect in subsequent moves. Thus one aspect of the persona concerned with functional features of "circulation" within an apartment, cannot "fix" the width of a door in such a way that it will not be altered by its "cost accounting" faculty for instance. This feature corresponds to the principle of the *manifest nature of design documents* discussed above.

There is not much to say about the system of representation used in the forum, because it is completely application-specific. The way(s) this representation is interpreted at the level of the persona is specific to the persona (see below). In practical terms however, given the fact that the representation used in the document is some formal abstraction, there is a need for an a priori general characterization of systems of representation which could be used. In other words the system needs to operate in some well-defined microworld<sup>2</sup>.

#### 4.1.2 The Persona

The persona is primarily a collection of modules, each of which corresponds to a potential quality of a design and a set of heuristics expected to effect improvements in terms of that quality. I have called these modules "thematic modules", but not every one of them necessarily corresponds to a "theme", and the sense in which they are modular does not exclude redundancies among them. In fact, redundancy can be a source of robustness, and makes it possible to design different modules without extensive coordination. Associated with each module is a bias factor. This factor indicates the relative importance of the different modules for a given persona.

---

<sup>2</sup> Note that in any design system, there are many constraints which can be considered as inherent in the medium used. For instance, if a CAD system only allows one to draw orthogonal lines, an inherent constraint can be said to prevent the creation of triangles. These types of restrictions are elevated to the level of metaphysical constraints of the type "a line cannot be two places at the same time". They are not to be confused with constraints which are not inherent to medium or method, nor is it a good idea to implement requirements of a design (such as "the building must not be higher than x meters") as metaphysical constraints.

The choice of which qualities should be lumped together in a module is very much a design issue which I will touch upon only briefly in chapter 5 ("Outline of Future Work"). The components of a thematic module, however, are well defined. They are:

1. **The seeing-as function:** This function takes the current or proposed state of the document as input, and returns a module-specific description (called the "subjective description") as output. It also returns a data structure called the "objective/subjective mapping" documenting the correspondence between the components of the document and those of the subjective description. For example, one module's seeing-as function can represent the drawing of a building facade as a set of shades and shadows, while another module's seeing-as function represents the same facade as a number (the number of windows, for instance). The fact that each module has a different seeing-as function corresponds to the principle of *multiple semantics*. The role of this multiplicity is very important in determining the characteristic behavior of the discursive architecture.
2. **The appraisal function:** Each module applies its appraisal function to its own subjective description of the document. The appraisal function returns a value which can be used to compare the given subjective description to one of its other states. This function is used to evaluate both the document and next states of the document proposed by the module itself or other modules. The score returned by the appraisal function should be used within its module. Due to the principle of the *incommensurability of criteria*, it cannot be compared to the appraisal scores of other modules for the same document. (In the current implementation of the discursive generator appraisal scores are also used outside the module. This will be changed in subsequent versions in order to conform more fully with the incommensurability principle.)
3. **The know-how:** The know-how is a collection of move rules. Each move rule has a condition predicate which tests the subjective description of the document. If the condition is met, the move rule is applied to the



subjective description as well as to the objective/subjective mapping produced by the seeing-as function. The move rule returns an "objective transformation function" which can, in turn be applied to the actual document to yield a proposed next state (see figures 4.2 and 4.3). Associated with each move rule (and with the corresponding proposed next state of the document) is a factor, called the "eagerness factor" corresponding to a measure of conviction. Eagerness factors are adjusted by the focus manager at every cycle. They are the means of controlling the systems "internal" focus (as opposed to the document-level focus). Each move rule also has a "thematic index", which can be used to associate with it some key-words expressing its area of relevance, and which is used by the focus manager as one of the factors considered in adjusting of focus. The kinds of thematic indexes used will depend very much on the application.

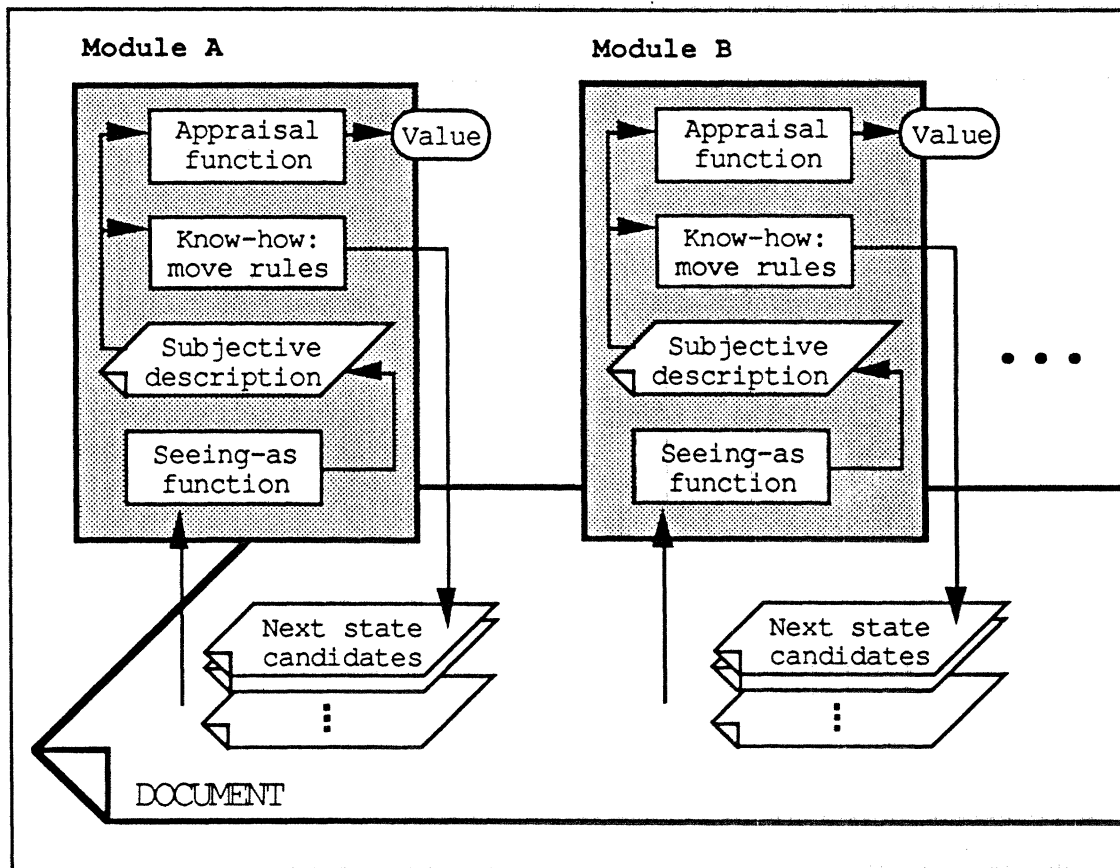


Figure 4.2

It is important to note that the transformations proposed by the move rules are as general (or parametric) as possible. If a move consists of aligning a rectangle with a line, then the destination of the rectangle is not specified as a point, but as a set of points along the line, with one of them acting as a default position. This feature is exploited by the arbiter (see below).

Moves in the discursive generator can have one of three types<sup>3</sup>:

- . Exploratory
- . Proactive
- . Remedial

Consider the following three statements about the need or the desire which results in a move (my emphasis):

I had better be more precise about this state of desire business: a state of desire takes place when from a state of satisfaction one passes to a state of mounting dissatisfaction and then, immediately thereafter, to a state of dissatisfying satisfaction, namely, of desire. It isn't true that the state of desire takes place when something is missing. [Calvino 1969]

But behind all action there was a protest, because all doing meant leaving *from* in order to arrive *at*, or moving something because it would be here and not there, or going into a house instead of not going in or instead of going into the one next door; in other words, every act entailed the admission of a lack, of something not yet done and which could have been done, the tacit protest in the face of continuous evidence of a lack, of a reduction, of the inadequacy of the present moment. [Cortazar 1966]

Unselfconscious cultures contain, as a feature of their form-producing systems, a certain built-in fixity -patterns of myth, tradition, and taboo which resist willful change. Form-builders will only introduce changes under strong compulsion where there are powerful (and obvious) irritations in the existing form which demand correction. [Alexander 1964]

---

<sup>3</sup> See chapter 5 for a brief discussion about a special fourth type of move, the "status quo" move.

The first statement attributes desire -the desire to act- to a sense of "mounting satisfaction" as opposed to the realization that "something is missing". What I call *exploratory* moves are the result of this type of motivation. The second attributes action to a sense of unexploited potential: the admission of "something not yet done which could have been done". This is the basis of what I call *proactive* moves. The third is concerned with action guided by the perception of problems "which demand correction". This type of reaction leads to what I call *remedial* moves. The significance of a move's type is in determining changes in the eagerness factor of the move in function of the preceding design history (see the section on the focus manager below). The spectrum of types of motivations is, to be sure, more of a continuum ranging from the purely exploratory to the strictly remedial, but I think that for the purposes of this thesis, the coarse subdivision into three types is an appropriate idealization.

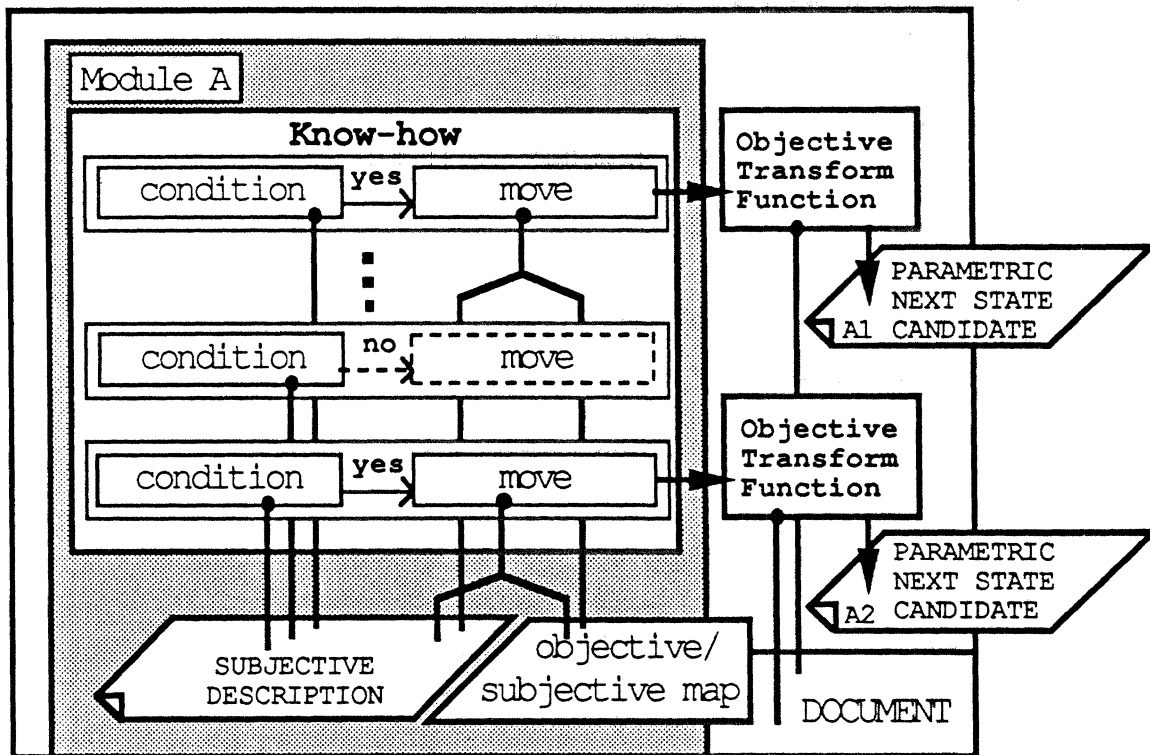
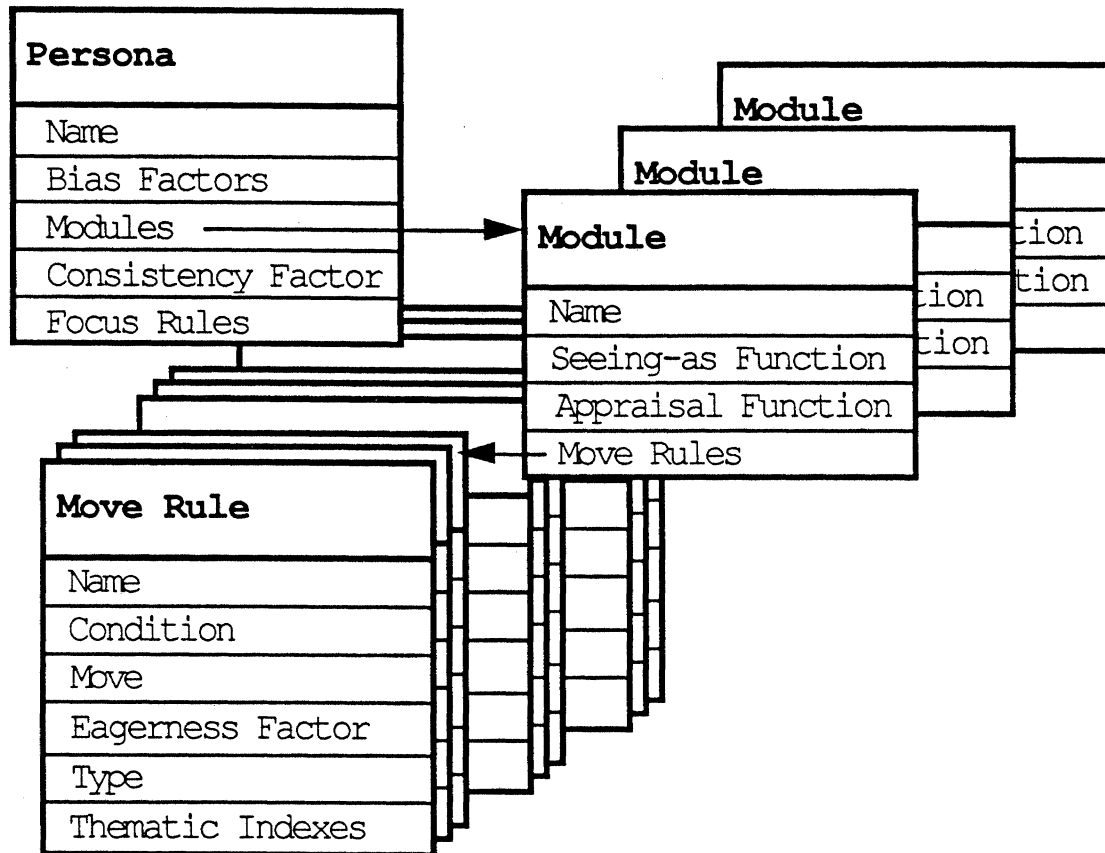


Figure 4.3

Note that there is no room in this subdivision for arbitrary moves, ones which are made, for instance, out of boredom. Rather, the proposed model accounts for the effect of states such as boredom at the level of their effect on

focus, eagerness, and so on. These can increase the likelihood of some move being made, but the source of the particular move is accounted for by one of the three motivations above.



#### 4.1.3 The Arbiter

The arbiter's job is to take all the next state candidates proposed by the different modules and choose one of them as the new state of the document. First the arbiter tries to "consolidate" the different candidates. This means that it takes every two or more parametric proposed next states and tries to find their "intersection". This potentially yields proposed next states which are the result of simultaneous moves (see figure 4.4). The number of moves associated with a candidate is called its "simultaneity index".

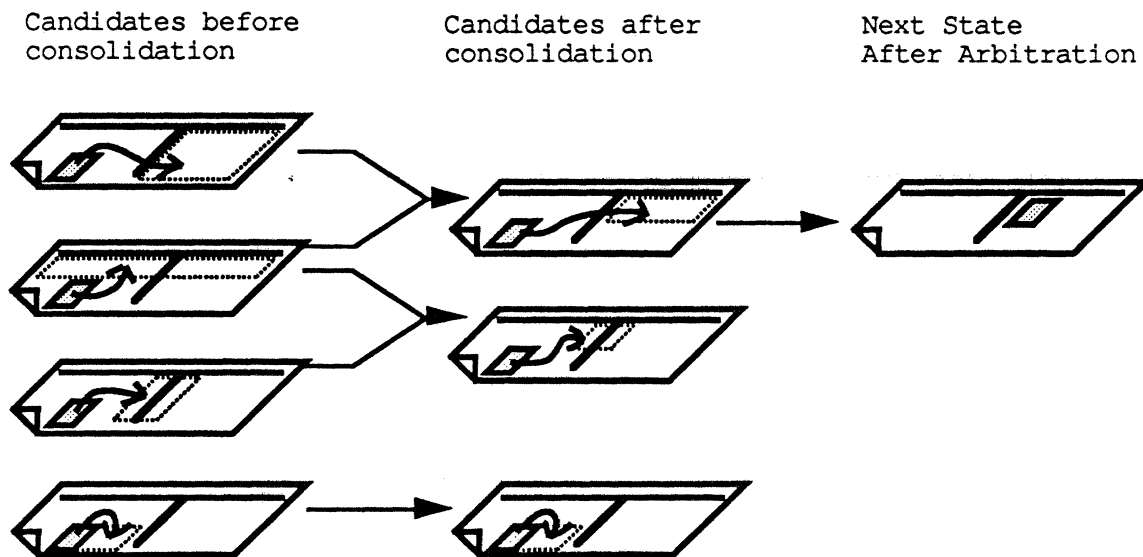


Figure 4.4

After trying to consolidate the candidates, the arbiter polls each module by having it appraise each candidate to determine its degree of (positive or negative) improvement. A set of initial scores are thus produced for the candidates by taking an average of these degrees of improvement. Note that this averaging of values produced according to different modules is a violation of the principle of the incommensurability of criteria. It dates from a stage in the development of the discursive generator when the incommensurability principle was not fully developed. In fact the artificial nature of the effort to somehow normalize the values produced by different criteria so that their effects could simulate a unified system of measure provided the motivation for developing the incommensurability principle. In the next version of the generator, the scores given by modules to different next state candidates could be replaced by an indication of "better", "worse" or "neutral". But ideally even these ratings should be eliminated, thus systematizing the process which is (even in the present version) the driving force behind the generator's behavior: The module's appraisals effect the system's focus, and focus plays the decisive role in the choice of a next state candidate. The focus manager, described below, does not rely on any assumption of commensurability of evaluative criteria.

The scores assigned to each candidate are weighted using the following factors:

- . The (maximum) *eagerness factor* of the move(s) associated with a candidate.
- . The *bias* of the persona for the thematic module associated with a candidate
- . The *simultaneity index* of the candidate

The candidate with the highest score becomes the next state of the document<sup>4</sup>.

#### 4.1.4 The Focus Manager

At each cycle the focus manager arranges the document such that the components involved in the latest move are the most likely ones (the first ones) to be considered by the different move rules. Each move rule can, however, propose a transformation of some other components due to the evaluation of its condition predicate. In order to promote coherence, the focus manager also arranges the components according to criteria specific to the modules. Even before the modules are given a chance to propose moves, their criteria for choosing a component for modification and their eagerness factors are used to re-arrange the components a second time. Thus, if one module is very eager to transform a certain component, this component is

---

<sup>4</sup> The scores returned by the appraisal functions (and, of course, those produce by the averaging operation) range between -1 and 1. The three factors applied as weights (eagerness, bias, simultaneity) range between 0 and 1. Given a score  $S$  and a weight factor  $W$ , the weighting function returns the weighted score  $S'$  such that:

$$\begin{array}{ll} \text{for } S > 0 & S' = S + (1 - S)W \\ S = 0 & S' = 0 \\ S < 0 & S' = S + (0 - S)W \end{array}$$

The effect of this function is to raise the score  $S$  towards a ceiling of 1 (if  $S$  is positive) or a ceiling of 0 (if  $S$  is negative) by a ratio of  $W$ . This means, for example, that if a weight of  $1/2$  is applied to a score of  $(1 - d)$ , where  $d$  is between 0 and 1, the score will increase by  $d/2$  (i.e. it will go  $1/2$  of the way towards its ceiling). Thus negative scores remain negative and positive ones positive, and the increase is smaller for scores closer to their ceiling than for scores farther from their ceiling. Other features of this function are that the order in which weights are applied does not matter and a weight of 0 has no effect while a weight of 1 turns negative scores into neutral ones (0) and positive scores into the maximum (1).

favored by the system with the hope of having all the modules focus on it. Again, there is no guarantee that the other modules will actually focus on that component, but those modules which do not have any specific preferences at that time will do so. This results in a simple model of component-level focus and distraction. Internally, another kind of focus is implemented by the manipulation of eagerness factors at each cycle. This manipulation is done according to "focus rules" specific to the persona. The following is an example of a standard set of focus rules, specifying the way each move rule's eagerness is to be altered:

- . Decrease the factor when the move was just made on the previous cycle.
- . Increase it if the move rule's associated module was just used anew (that is, after not having been used for n moves).
- . Increase it when the move's *thematic index* matches (by some measure) the current index.
- . Increase it if the move's type is "remedial" and the latest average evaluation of the document was low.
- . Increase it if the move's type is "exploratory" and the latest average evaluation of the document was high.

All eagerness factors have some initial setting. When a factor is increased or decreased, it gradually returns to its initial setting.

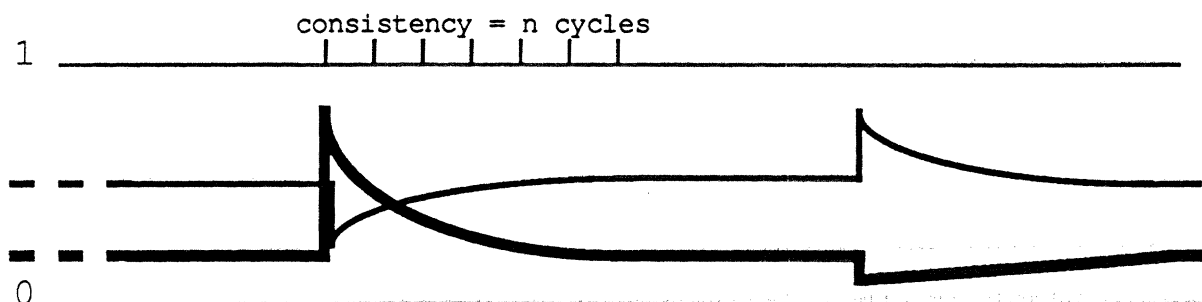


Figure 4.5

The number of cycles it takes for a factor to return to its value prior to being altered is called the "consistency factor". This factor is one of the characteristics associated with the persona. The lower it is, the more sporadic the document history is likely to be.

Eagernesses can be increased or decreased by a factor<sup>5</sup> ranging from 0 to 1. To be more precise about these changes, we must point out that a relative increase of an eagerness by a factor of  $W$ , for instance, is actually executed as an increase of  $W/2$  for that factor, and a decrease of  $W/2$  for all other factors. This trick avoids the rapid desensitization of eagerness factors which can result from quickly pushing them to their ceiling.

As we have just seen, the concept of focus is used in the discursive generator at two different levels. The selectivity of the system at the level of components comes closest to the traditional notion of cognitive focus. But at the level of the modules, the role of focus is to activate some current intentions or goals as well as the ways of seeing associated with them. It is this second level of focus which provides the opportunity for implementing sophisticated patterns of behavior in the context of the discursive generator. If it is appropriate for a given implementation of the generator to proceed according to a certain plan of action (favoring some modules at given stages of a design session, for instance) then focus rules can be used to favor the adherence to such a plan. And, like the move rules which make up each module's know-how, focus rules can go beyond simple "if ... then ..." statements and make use of precedents to find patterns of action which are likely to have favorable results for the current design task.

The focus manager is also the appropriate component of the generator in which to implement any stopping criteria. A simplistic mechanism for deciding when to stop the design session could rely on a time limit. Since the focus manager keeps track of the history of the design session, it could gradually "wind down" the module's eagerness factors until no more design moves are suggested. I do not mean to suggest that such a strategy for stopping is a good one, or that the discursive generator should have a

---

<sup>5</sup> This factor is applied as a weight, using the weighting function described in footnote 4 above.



mechanism for stopping. But any explicit manipulation of the history of design sessions produced by the generator should happen at the level of the focus manager. Such an approach could have the advantage of easily maintaining the discursive generator's dynamism, but more importantly, it would avoid imbedding in specific modules global strategies which can control other modules' behavior.

## 4.2 A More Subtle Way of Playing with Rectangles: An Application of the Discursive Generator

The preceding section established some correspondences between the architecture of the discursive generator and the theory of design activity on which it is based. But these correspondences are still purely formal ones and, as such, they do not necessarily provide any assurances of good performance. In this section I will present a simple application of the discursive generator by way of illustration, and in order to be able to discuss with specific examples certain characteristics of its behavior.

The "design task" for this demonstration is to arrange three blocks of variable height and rectangular footprints of arbitrary dimensions, according to some formal principles of massing. These principles are quite simple. They favor:

1. Maximum alignment and abutment of the blocks
2. Compactness (smallest possible maximum length and width)
3. Some constant footprint-to-volume ratio
4. Visibility of all the blocks from a given view point

Depending on one's reading of these principles, the task of arranging three blocks according to them can be considered underconstrained. Let us say, for instance, that what we mean by "maximum abutment" is that there be a condition of adjacency among every pair of blocks in the configuration. And let us take "maximum alignment" to mean that, in addition, there be as many conditions of alignment as possible among the footprints of the blocks, given the particular blocks in question. In that case, given any three (or fewer) blocks, a satisfactory arrangement can be found. First I will present, below, an example of the block arrangement application for this underconstrained case. Section 4.2.6 shows the more interesting example of the discursive generator's behavior in an over-constrained case of five blocks. It is this second example which shows the generator's advantages over more traditional methods.

### 4.2.1 A Sample Run

Figure 4.6 shows a sample run of the discursive generator working on this task. In frames a and b, the generator adds rectangles of random dimensions.

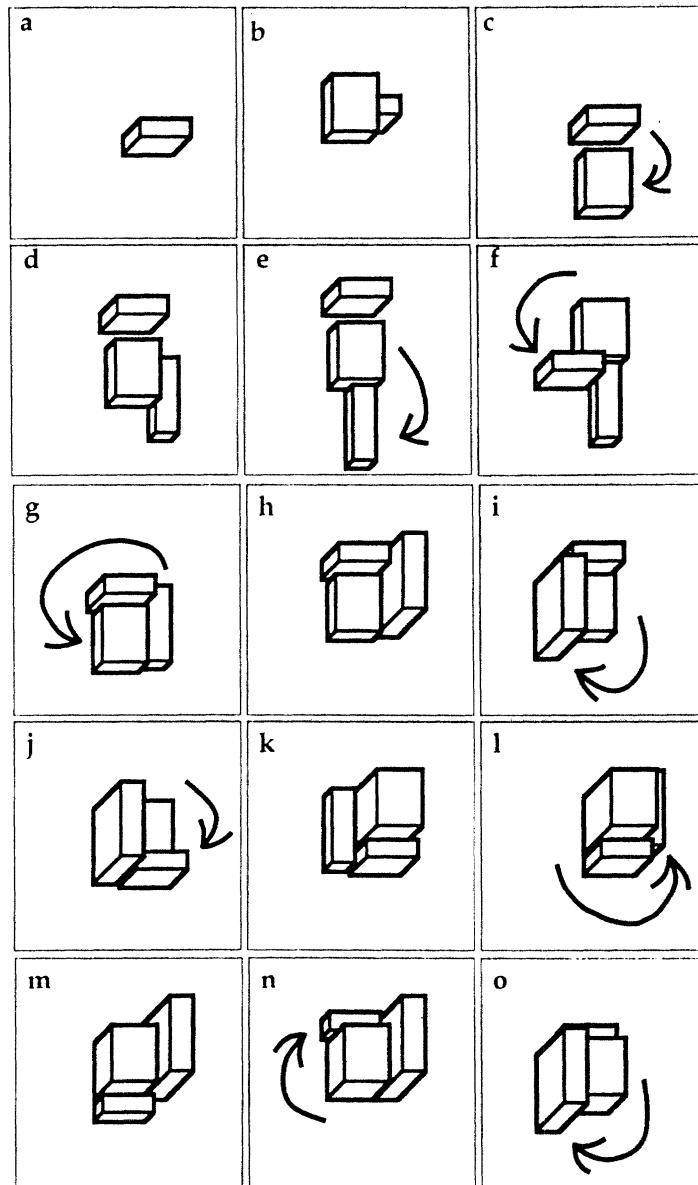


Figure 4.6

In frame c it undoes the undesirable overlap of the two rectangles before adding a third one in d. (It might have been just as likely for the third rectangle to be added before the overlap of the first two was undone, depending on the *eagerness* and *bias* factors associated with the different modules, as described above.) Note that the move in frame c which undoes the overlap also manages to align the right sides of the blocks. Similarly, in e, the generator moves the third block out of its overlapping position and places it in abutment with the larger block.

In frame f the remaining loose block is placed in a position where it abuts with the two others. Although the configuration in f provides the maximum number of alignments and abutments possible with these three blocks, it is particularly unsatisfactory in terms of heights, compactness and the visibility of all the blocks from the perspective of a given point, namely a point at "ground" level, at the lower left corner of the frame, looking towards the upper right corner. In frame g, the larger block is rotated to the opposite side of the other blocks, enhancing the compactness of the configuration. But one of the smaller blocks is still not visible from the lower left corner. This is corrected in h by lowering the block which is closer to that corner and raising the one which is farther back. These changes in height are done in such a way as to preserve a given ratio between the surface area covered by the footprints of the blocks, and their total volume.

The configuration in frame h, as well as those in k and m, satisfy the massing principles stated above to the extent that it is possible to do so given these three blocks. But the system does not necessarily consider the design complete whenever (or only when) some conditions are satisfied to a certain (or even maximum) degree. The fact is that the system, as presented in this example, does not have explicit criteria for stopping. Although the issue of when to consider a design task accomplished (and how to begin designing) is beyond the scope of this thesis, the architecture of the discursive generator does imply some partial answers which are discussed in section 4.3 below. Suffice it to indicate for now, that according to this principle of "good does not mean done," the generator produces, in frames i to o (and beyond) some additional arrangements of the blocks.

Before giving further samples of the discursive generator's output I will explain the details of this block arrangement application.

#### 4.2.2 Five Modules for Arranging Blocks

The *persona*, which embodies the faculties of *appraisal* and *know-how* used in the application, is composed of five *modules*. The window at the top left corner of Figure 4.7 shows the contents of the document at a certain stage of development. Note that the desired number of rectangles and the particular area-to-volume ratio used are part of the document, and could be changed by the *persona* during the course of the design session (even though the present modules do not include that possibility). The other five windows represent the document as *seen* by each of the five modules.

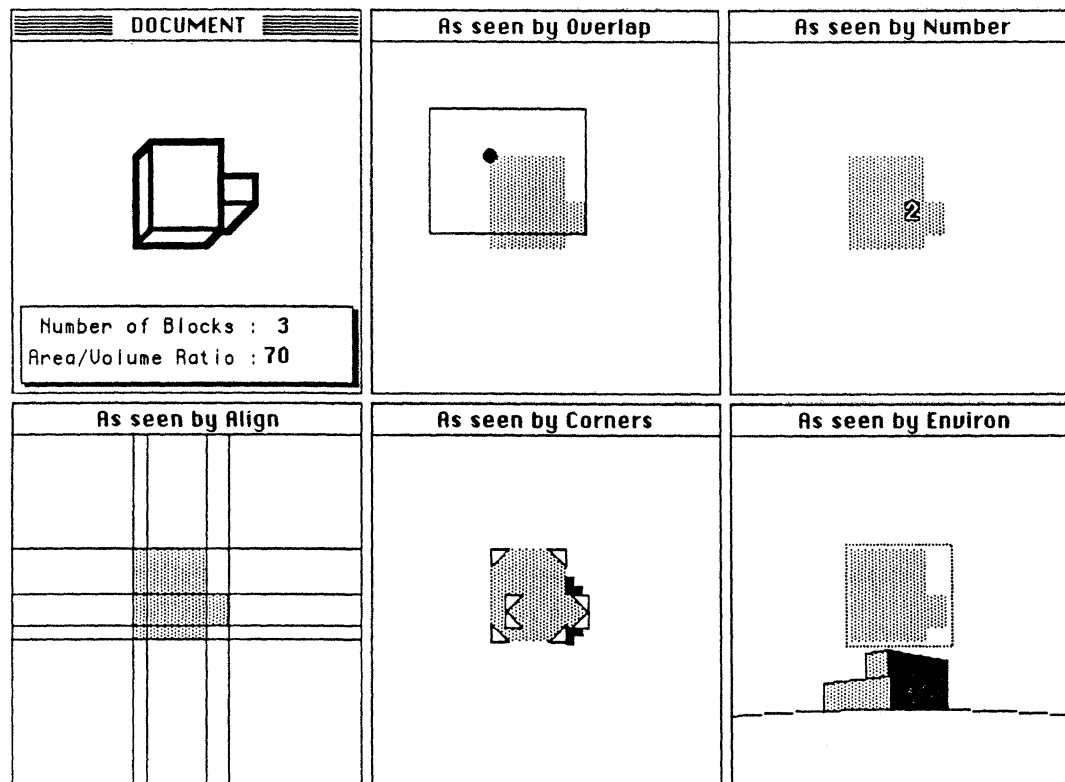


Figure 4.7

The **Overlap** module sees the blocks in the document as the configuration space<sup>6</sup> of each of the other blocks. It represents the document as a set of points (the origins of the blocks' footprints) and rectangles ("obstacles" in the configuration space), and favors situations where points are located outside their corresponding rectangles.

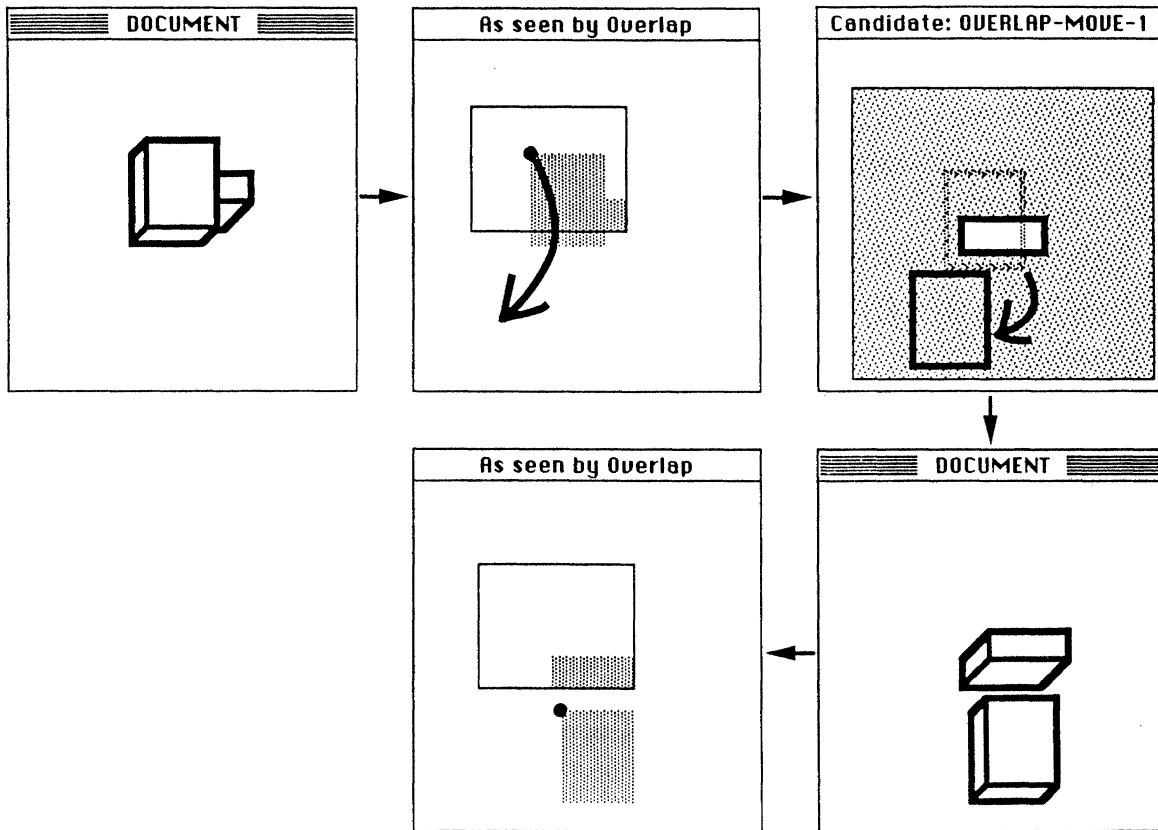
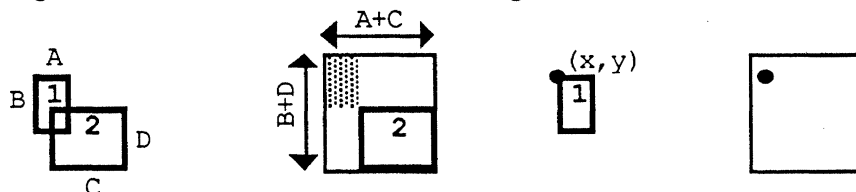


Figure 4.8

<sup>6</sup> A configuration space representation is used to reduce the task of checking the intersection of two rectangles to one of checking whether a point is inside a rectangular region. Given two rectangles, where rectangle 1 has sides of length  $A$  and  $B$ , and rectangle 2 has sides of length  $C$  and  $D$ , the configuration space of rectangle 1 is a rectangular region of sides  $A+C$  and  $B+D$ , whose lower right corner coincides with that of rectangle 2.



Rectangle 2 is thus replaced by this (larger) rectangular region, and rectangle 1 is replaced by a point  $(x,y)$  at its upper left corner. It is not difficult to see that the rectangles intersect if and only if the point is inside the rectangular region.

This module *appraises* the document by calculating the ratio of overlapped area to total footprint area. Its know-how consists simply of moving the points outside their corresponding rectangles. The first two windows of Figure 4.8 show a given state of the document, the way it is seen by the Overlap module and the "subjective transformation" proposed by the module (moving the point outside the rectangle).

This subjective transformation corresponds to a move at the level of the document (namely, displacing one of the rectangles). The move is specified parametrically: the gray area in the third window of figure 4.8 is the set of potential destinations of the displaced rectangle, whereas the black rectangle is a default position.

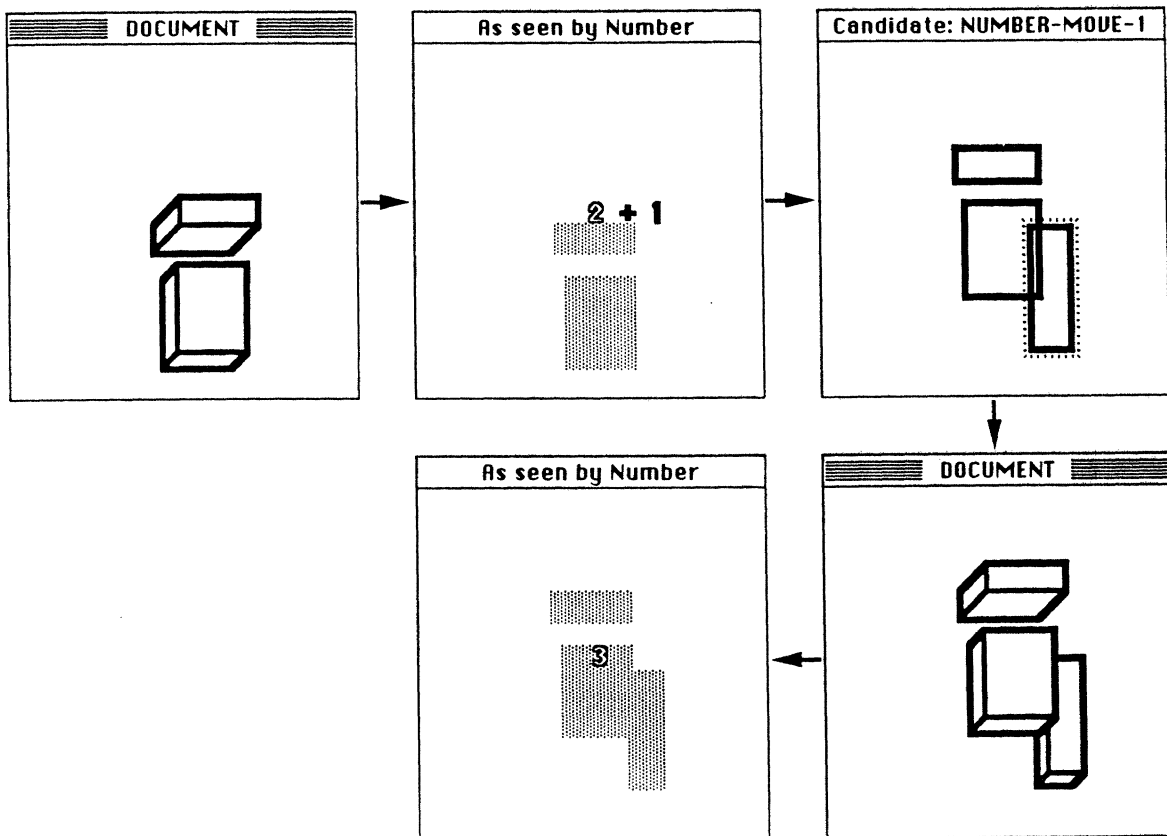


Figure 4.9

If executed, this move will produce a new state of the document which is *expected* to have a better value in terms of this module's appraisal. A move

proposed by a module can, in principle, have a negative effect on the document from the point of view of the module which proposed it. Such a move could even be chosen as the one which will actually be executed if, for instance, it is particularly favorable from the point of view of some other modules. It could even be the case that a move which is unfavorable in general is chosen as the next move because of the excessive eagerness of a given module.

The **Number** module's behavior is illustrated in figure 4.9. This module sees the document as a number, namely the number of blocks in it. It can propose one of two moves: increasing the number by one or decreasing it by one. Its appraisal function counts the number of blocks and compares the result to the desired number of blocks as recorded in the document (see figure 4.7).

A future enhancement of this module might add to its know-how the ability to change the number of desired blocks. If the initial eagerness factor associated with this ability is sufficiently low, then its effect might be to execute such a move only in cases where an extremely overconstrained situation eventually results in the reduction of the eagerness factors of all the alternative move rules.

The **Align** module, as shown in figure 4.10, sees the document as pairs of vertical and horizontal lines. For each block it sees two pairs of lines corresponding to each pair of parallel sides of the block. This module favors coincident lines in its subjective description. The moves it proposes displace one line to place it over another. At the level of the document, this results in moving a block to one of a set of possible destinations, all of them having the property that they create new alignments. The gray area in the window entitled "Candidate: ALIGN-MOVE-2" of figure 4.10 is one set of possible destinations proposed by the Align module.



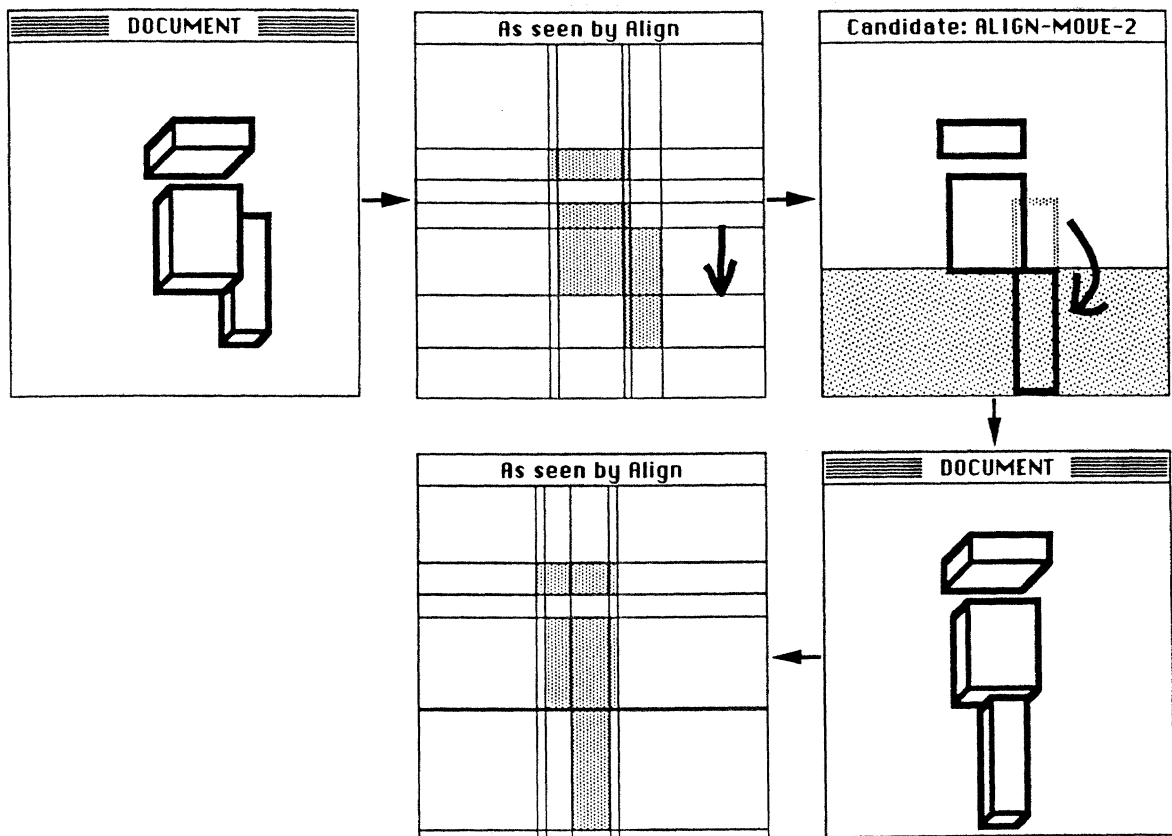


Figure 4.10

The **Corners** module deals with convex and concave corners, and favors situations where convex corners are "tucked into" concave ones. The second window of Figure 4.11 shows how this way of seeing is represented. There are four white triangles for the four (convex) corners of each block's footprint, and a black L for any concave corner formed by the abutment or intersection of any two blocks.

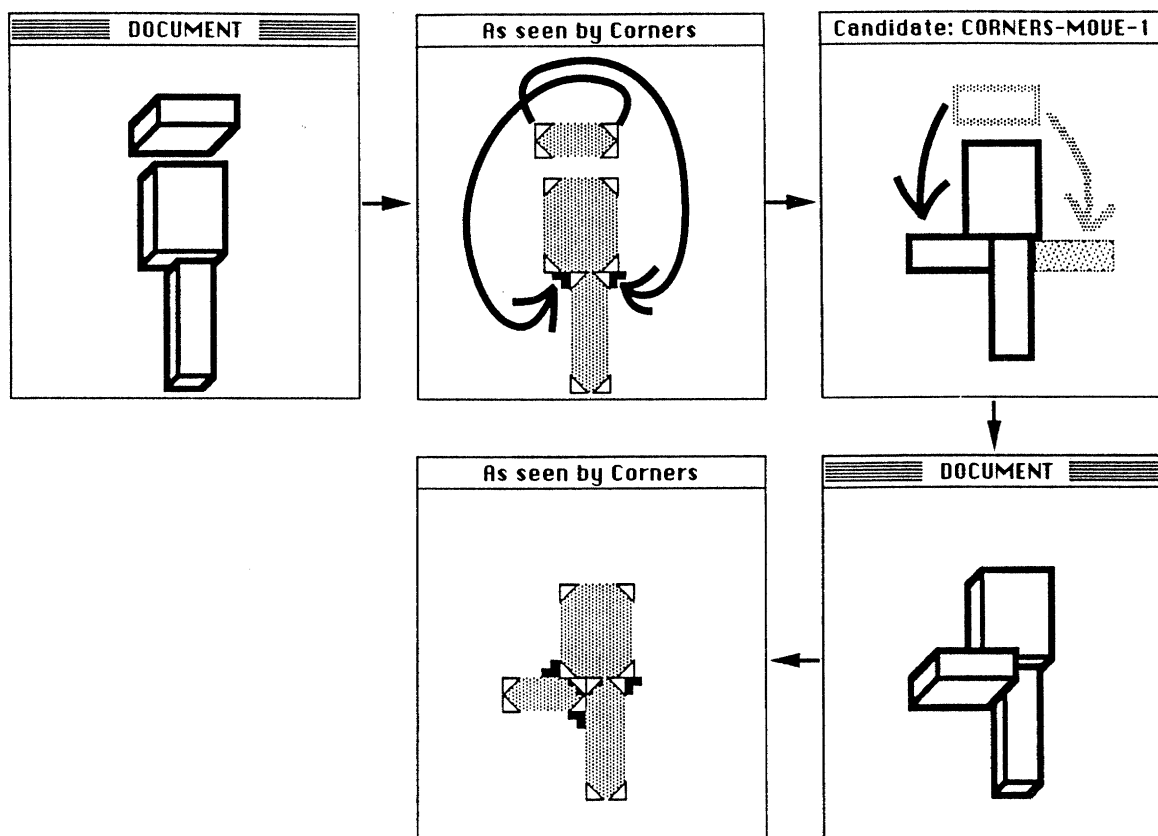


Figure 4.11

The Corners module evaluates its subjective description of the document according to the number of convex corners which are not coincident with some concave corner. Thus it is never quite satisfied, because there will always be some (at least a minimum of four, but possibly a higher minimum depending on the given set of blocks) free convex corners. This module proposes moves which consist in placing a white triangle (a convex corner) over one of any number of black L's (concave corners). In its default choice of concave corner, the module favors ones which are not contiguous to the set of convex corners corresponding to one block.

The **Environ** module sees the document as a perspective view from the lower left corner of the document's frame looking towards the opposite corner. Its appraisal is based on the visibility of two faces of each block in that perspective, as well as the compactness of the configuration (the compactness

being simply a ratio of the area of the bounding rectangle drawn around the footprints to the area of the footprints themselves). The know-how of the Environ module, however, deals only with the first of these issues. Note that this module has global appraisal functions, while the other modules check local relationships between pairs and triples of blocks. The module knows how to adjust heights so as to raise blocks in the back and lower ones in front. These adjustments are done in such a way as to keep the desired ratio of area to volume recorded in the document. Figure 4.12 illustrates a typical height adjustment proposed by the Environ module.

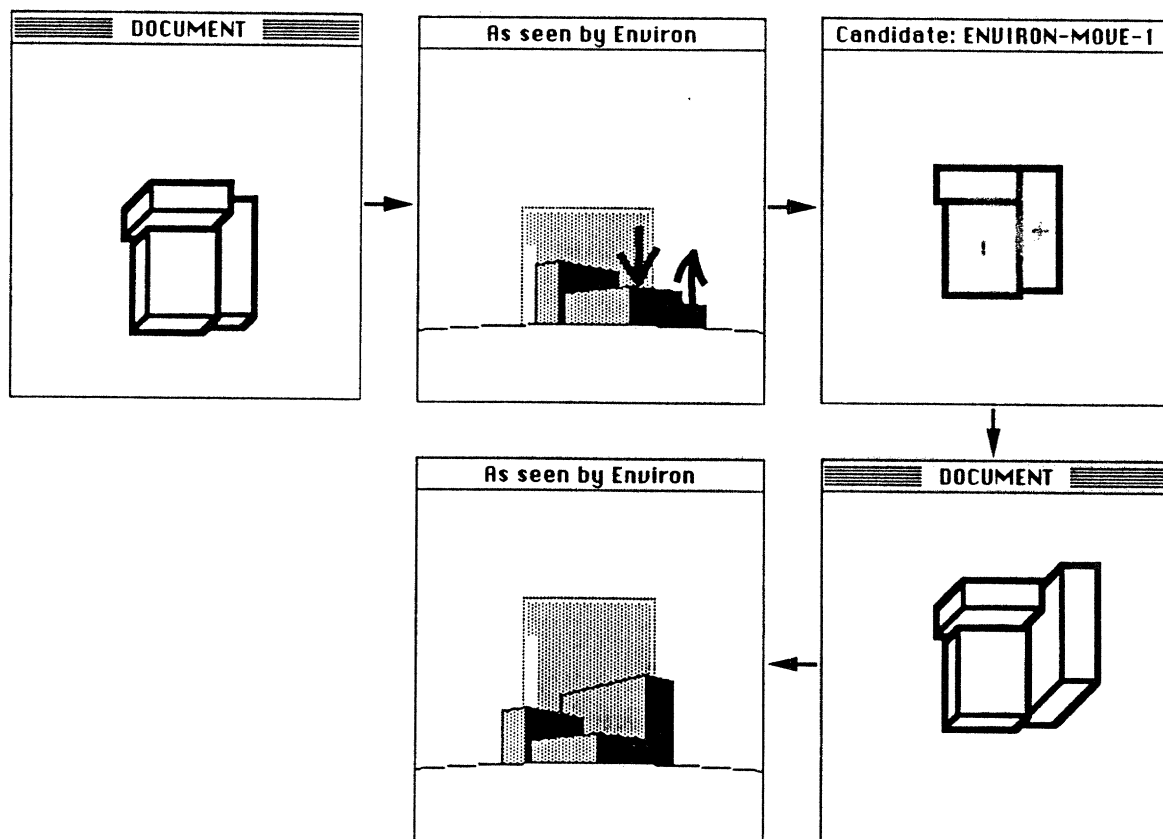


Figure 4.12

### 4.2.3 Parametric Next State Candidates

At each of the see-move-see cycles a number of moves are proposed by the five modules. Figure 4.13 shows a set of such proposals suggesting transformation to the document in frame b of figure 4.2 above. Each of these corresponds to a candidate for the next state of the document. For instance, the top right window of figure 4.13 (entitled "Candidate: NUMBER-MOVE-1") shows a suggestion by one of the move rules in the Number module to add a third block. Note that the Environ module (lower right window) has no proposals to make, because the condition for the applicability of its move rule (namely that all the blocks which are present in the document be in a "cheek-to-cheek" position with at least one other block) is not met.

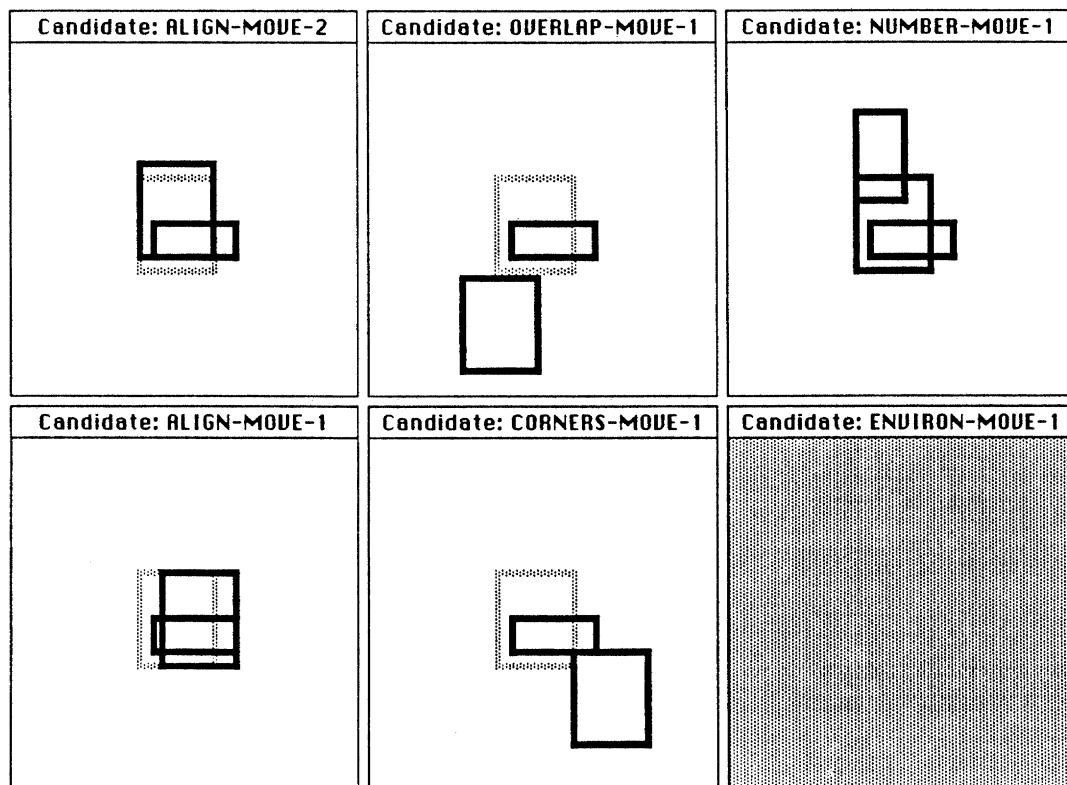


Figure 4.13

The proposed candidates show their default suggestions in these figures. Their actual proposals, however, are often more general. For example, one of the two moves proposed by the Align module -the one entitles "Candidate: ALIGN-MOVE-1", is shown in its full parametric form in figure 4.14. This parametric proposed next state is a disjunction of four sets of possible destinations for the chosen block. Move 1 of the Align module is concerned only with aligning vertical sides. The gray area in figure 4.14.a corresponds to all positions of the block in question which would align its right side with the right side of the other block.

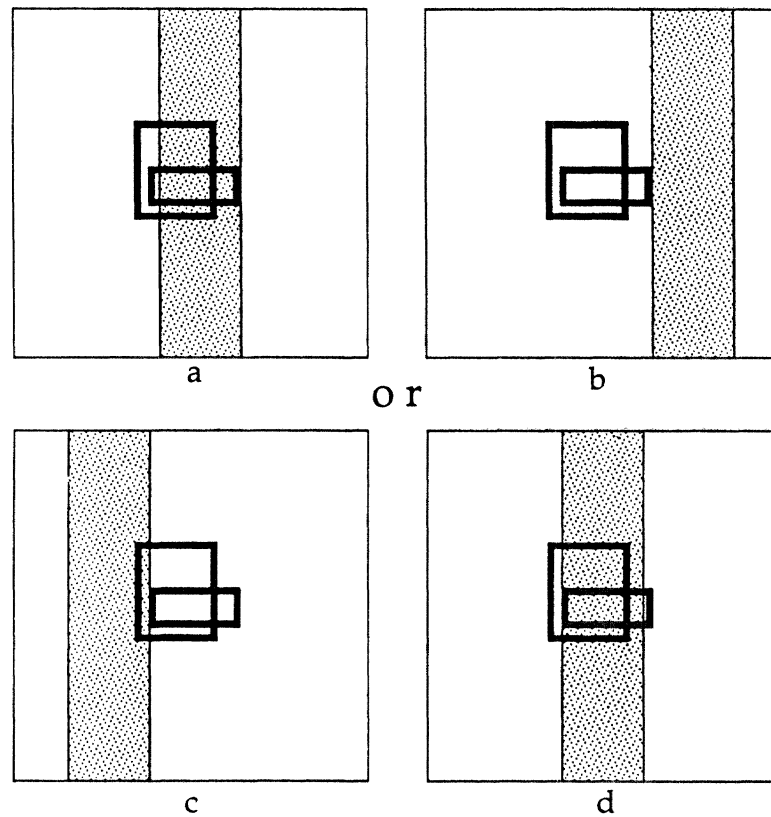
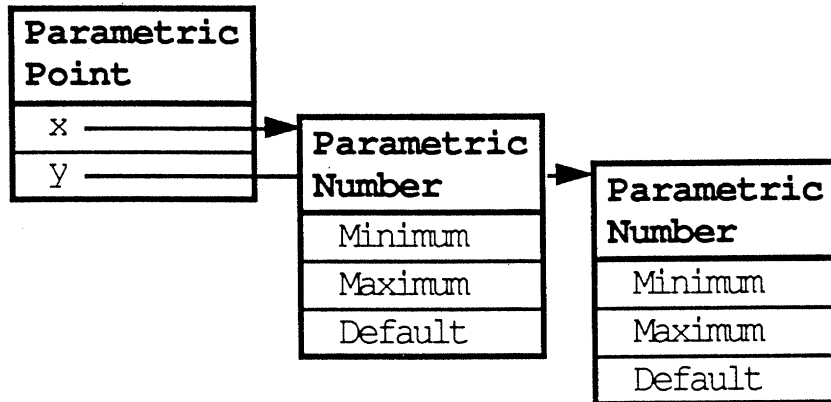


Figure 4.14

The gray area in b, shows all positions (proposed by the module) for which the right side of one block would align with the left side of the other, and so on. Such a disjunction is implemented as a list of four "parametric-points", one for each of a, b, c and d of the figure. A parametric point is a point whose

x and y coordinates are not necessarily single values but "parametric-numbers", that is inequalities plus a default value.



When all the modules propose their candidates, the arbiter consolidates them by trying to find intersections among the different parametric proposals (see the next sub-section for details of consolidation).

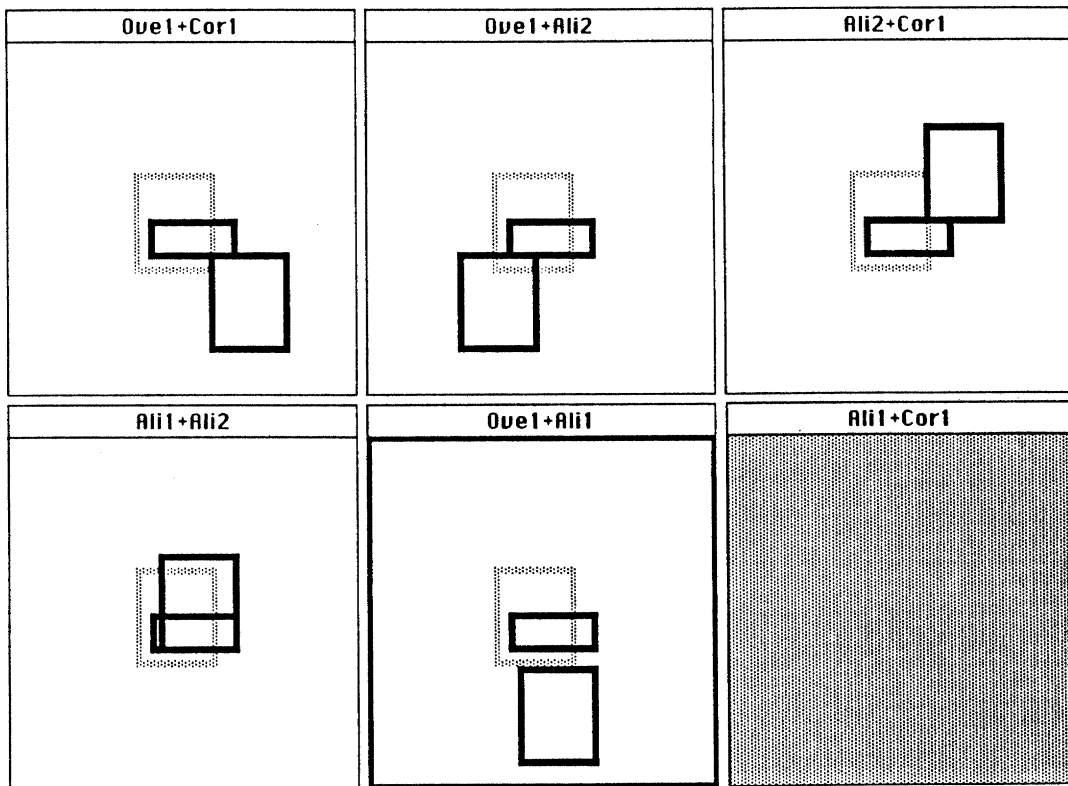


Figure 4.15

Figure 4.15 shows all the combinations of moves produced by consolidating the proposals shown in figure 4.13 above.

Note that some of the modules, such as the Number module, are incompatible with all others. Also notice how the window entitled "Ali1 + Cor1" is empty because the proposal of move rule 1 of the alignment module does not intersect that of the Corner module. In this case, the next state turns out to be the one proposed by the combination of Overlap1 and Alignment1, shown in the window entitled "Ove1 + Ali1".

#### 4.2.4 Consolidation and Simultaneity

The current implementation of the discursive generator can display, at each cycle, the eagerness factors and appraisal scores associated with each move rule. Figure 4.16 shows such a display. Each module has one or two move rules which constitute its know-how.

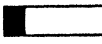
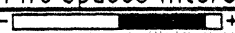


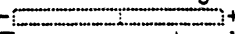






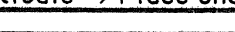


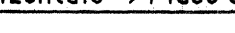




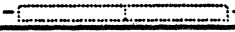

Overlap move rules	
 >	Two spaces intersecting -> Move one outside other -  + 
Number move rules	
 >	Too few rectangles -> Add a rectangle -  + 
 >	Too many rectangles -> Remove a rectangle -  + 
Align move rules	
 >	Non-coincident verticals -> Place one over other -  + 
 >	Non-coincident horizontals -> Place one over other -  + 
Corners move rules	
 >	Concave corner exists -> Plug it -  + 
Environ move rules	
 <	Top occluded -> Raise it -  + 

Figure 4.16

The thick bar to the left of the name of the move rules indicates the eagerness factor (between 0 and 1) of that rule. The arrow next to it shows whether the factor is currently rising or falling. The thin bar under the move rules, indicates the average score (between -1 and 1) given to the next state candidate proposed by that move rule.

If a move is active (that is, if its condition for proposing a move is met) a thin rectangle is drawn around it in the display shown in figure 4.16. The moves whose proposal is adopted as the next state of the document are indicated by a heavier outline. In this case Overlap1 and Align1 are consolidated, and their proposal is chosen. As mentioned above, simultaneous moves are favored over single moves. Figure 4.17 shows the mechanism by which the two candidates are consolidated.

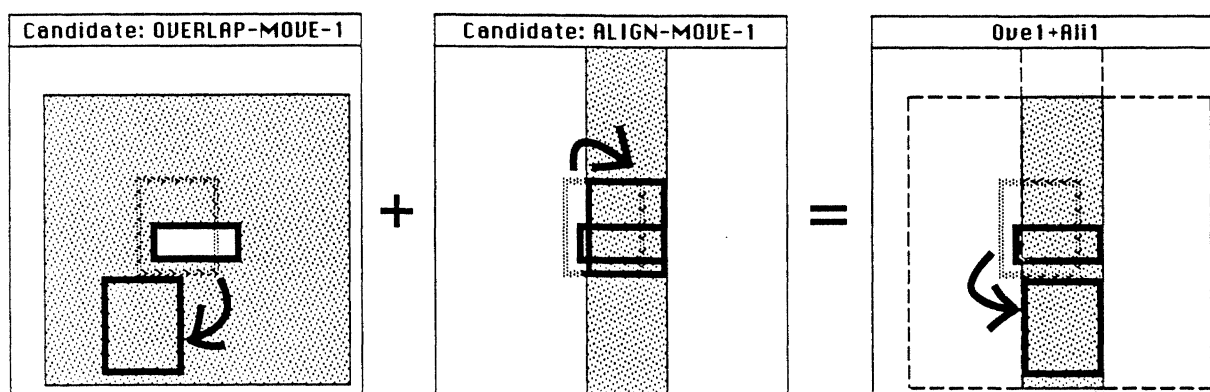


Figure 4.17

An interesting issue which is not dealt with thoroughly in this thesis is that of deciding the default position in a consolidated candidate. If the default of one (or both) of the proposals falls in the region of intersection which is retained in the consolidated version, then it (or one of them) becomes the new default. But if both proposed defaults remain outside the common area, then this implementation arbitrarily picks some limit position as default. In some cases this approach can fail to capture the knowledge inherent in one or both of the proposed candidates. A better solution might be for each candidate to order all the positions it proposes in terms of priority. Then the highest



order position remaining after consolidation might be chosen. In the consolidation shown in figure 4.17, the y coordinate of the Overlap module's default is retained as the new default's y coordinate. The x coordinate is fixed by the Align module.

It is important to make the following two points about this process of consolidation:

- . Consolidation is not an averaging operation. It involves no compromise among the different proposals.
- . The parametric nature of the final proposal is abandoned once it is chosen as the next state of the document. Only its default position is retained.

The second of these points is important in the context of the discussion about the manifest nature of design documents, above, whereas the first follows from the principle of the incommensurability of criteria.

#### 4.2.5 Component-Level Focus

In the preceding sections the mechanism for implementing focus at the level of the modules (using variable eagerness factors) was discussed. Let us now turn our attention to component-level focus, which is the system's tendency to favor certain components of the document (in this case certain blocks) over others at any given time. Note that in figure 4.13 above, most of the proposals involve some transformation of the larger of the two blocks. This is not entirely a coincidence, nor is it due to the system's preference for large components. It is due to the fact that the modules which did choose the larger block did not have a particular reason to prefer one component over the other. because the larger block was the center of focus by virtue of being the latest addition, it was chosen by many of the modules. The component-level focus is not always the result of such passive processes. Figure 4.18 shows the evolution of this focus over frames a through f of the session shown in figure 4.1. Note how in frame f focus shifts from the latest addition to another block. This shift, for instance, is due to the eagerness of the Corners module to

concentrate on the loosest block, despite the Align module's tendency to favor the block which was previously in focus.

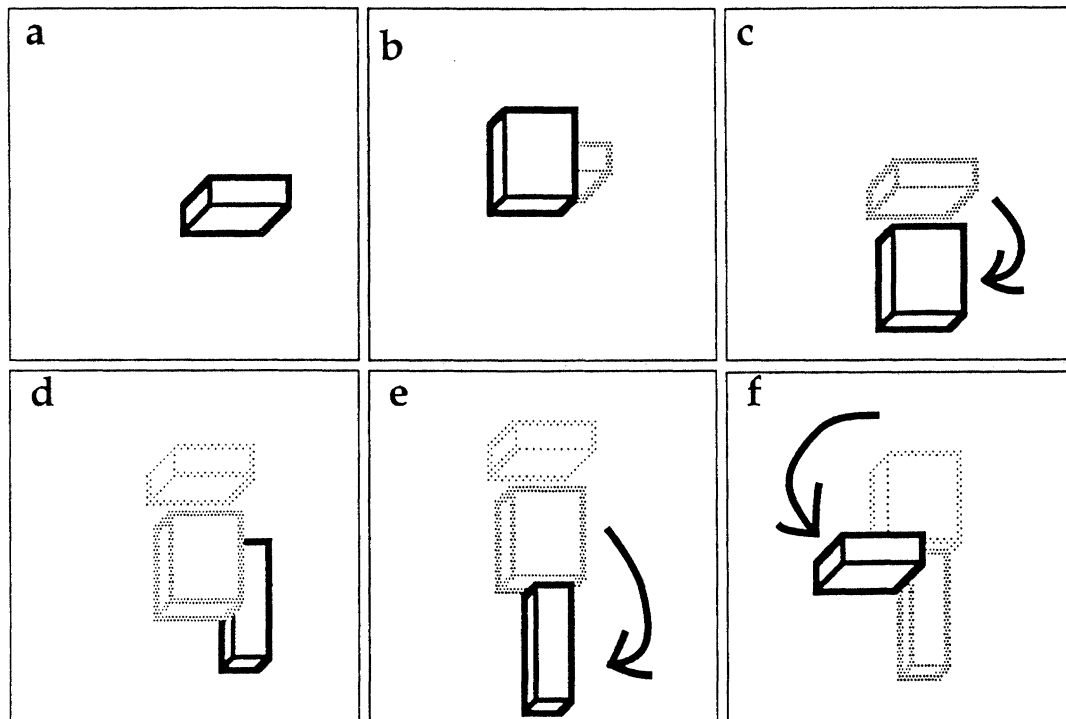


Figure 4.18

It would be interesting to speculate about the relationship of shifts of focus and the question of what constitutes a design move. In frame b of figure 4.18 a rectangle is added on top of another, and in frame c it is moved so as to avoid the overlap and to create an alignment. But what one would tend to call a move might be better characterized by the act of introducing a rectangle in a non-overlapping and aligned position (*b and c*), as opposed to introducing it, then fixing its position (*b then c*). Note that although the process depicted in figure 4.18 separates the transformation in b from that shown in c, b and c form a unit in the sense that they are bounded by shifts of focus.

## 4.2.6 An Overconstrained Case

The brief record of the discursive generator's behavior with three blocks, which was shown in figure 4.6, gives some hints of its potential for handling the more challenging task of generating reasonably satisfactory configurations in an overconstrained case where there are more than three blocks. Let us now observe such a case where the desired number of blocks is five.

Figure 4.19 shows the initial condition of the document (in the upper left window<sup>7</sup>) which contains seven blocks of arbitrary dimensions and location. The other windows show the document as seen by each of the five modules.

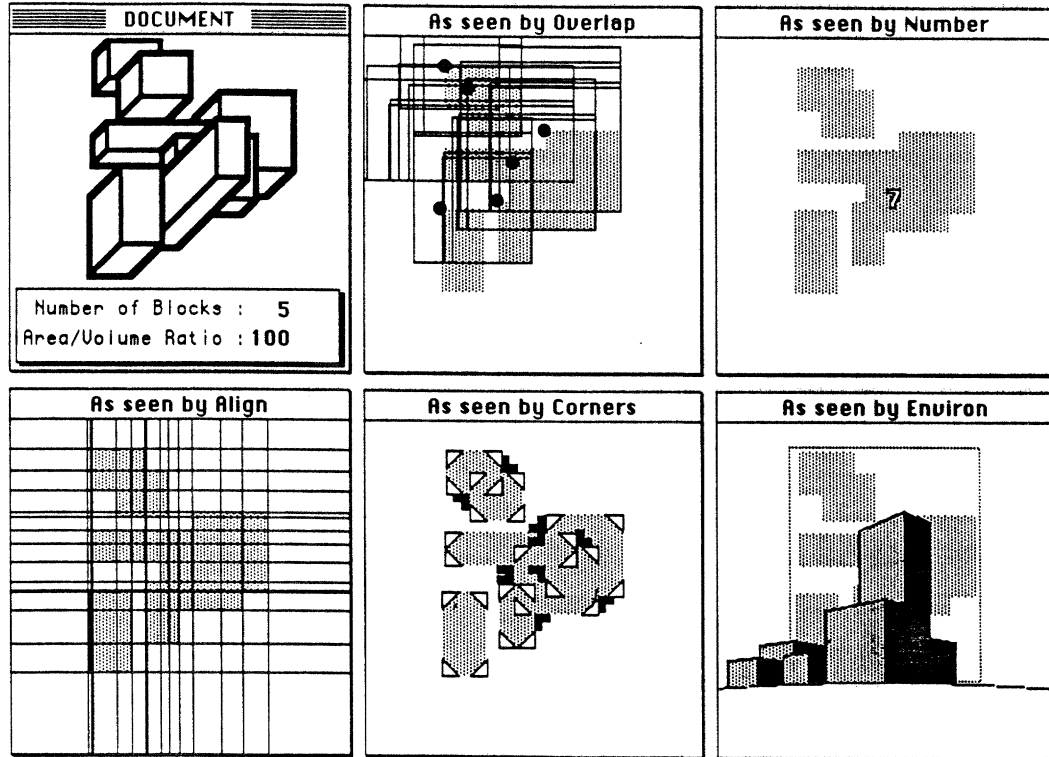


Figure 4.19

<sup>7</sup> The DOCUMENT window in this application of the discursive generator displays blocks in front as occluding ones further back. But for overlapping blocks it arbitrarily makes one occlude the other (in order to avoid constructing complex polygonal faces) occasionally resulting in an impression that some of the blocks are floating. The reader should try to ignore such impressions as much as possible.

Figures 4.20 and 4.20' show the first fifty moves produced by the generator. Before examining these moves, let us reinterpret the first of the four principles presented at the beginning of this section, namely that there should be "maximum alignment and abutment of the blocks", to simply mean that the blocks should form one cluster where any block can be reached from any other through a series of adjacencies.

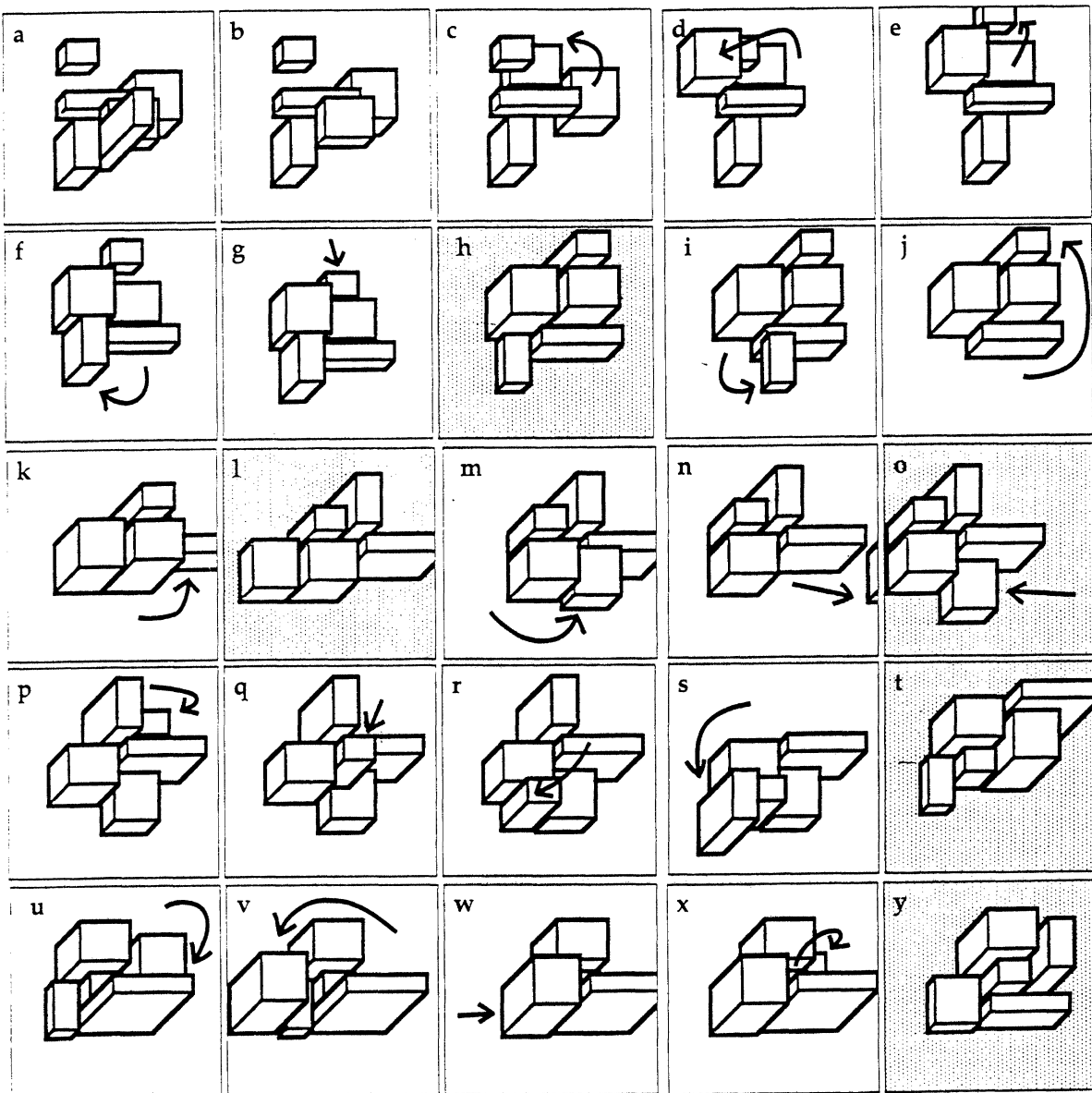


Figure 4.20

The highlighted frames in figure 4.20 and 4.20' correspond to configurations which meet this criterion and also satisfy the principle of visibility and area-to-volume ratio. The generator begins by reducing, in frames a and b, the number of rectangles. By frame h it has a configuration which conforms to our reinterpreted criteria.

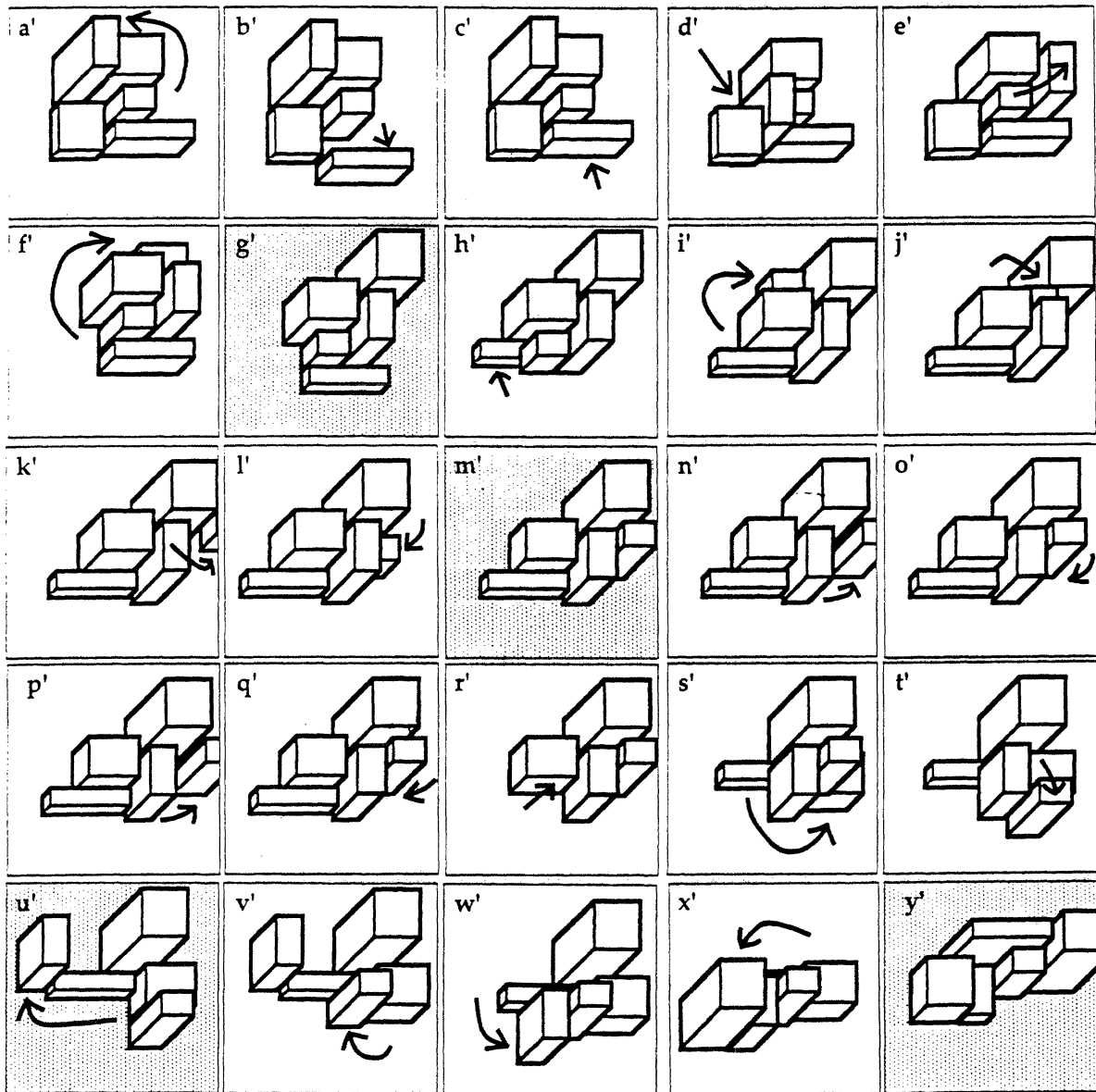


Figure 4.20'

But interestingly, after 25 moves it reaches, in frame y, a configuration which can be considered optimal for the full set of criteria. The larger blocks are arranged in a pinwheel pattern around the smallest one, thus achieving compactness (due to the pinwheel pattern) and maximum abutments and alignments (due to the fact that the smallest block is in the middle). Had a larger block been in the middle, the configuration could have been more compact, but the number of abutments would have been reduced because the smallest block would not have reached the next block in the pinwheel.

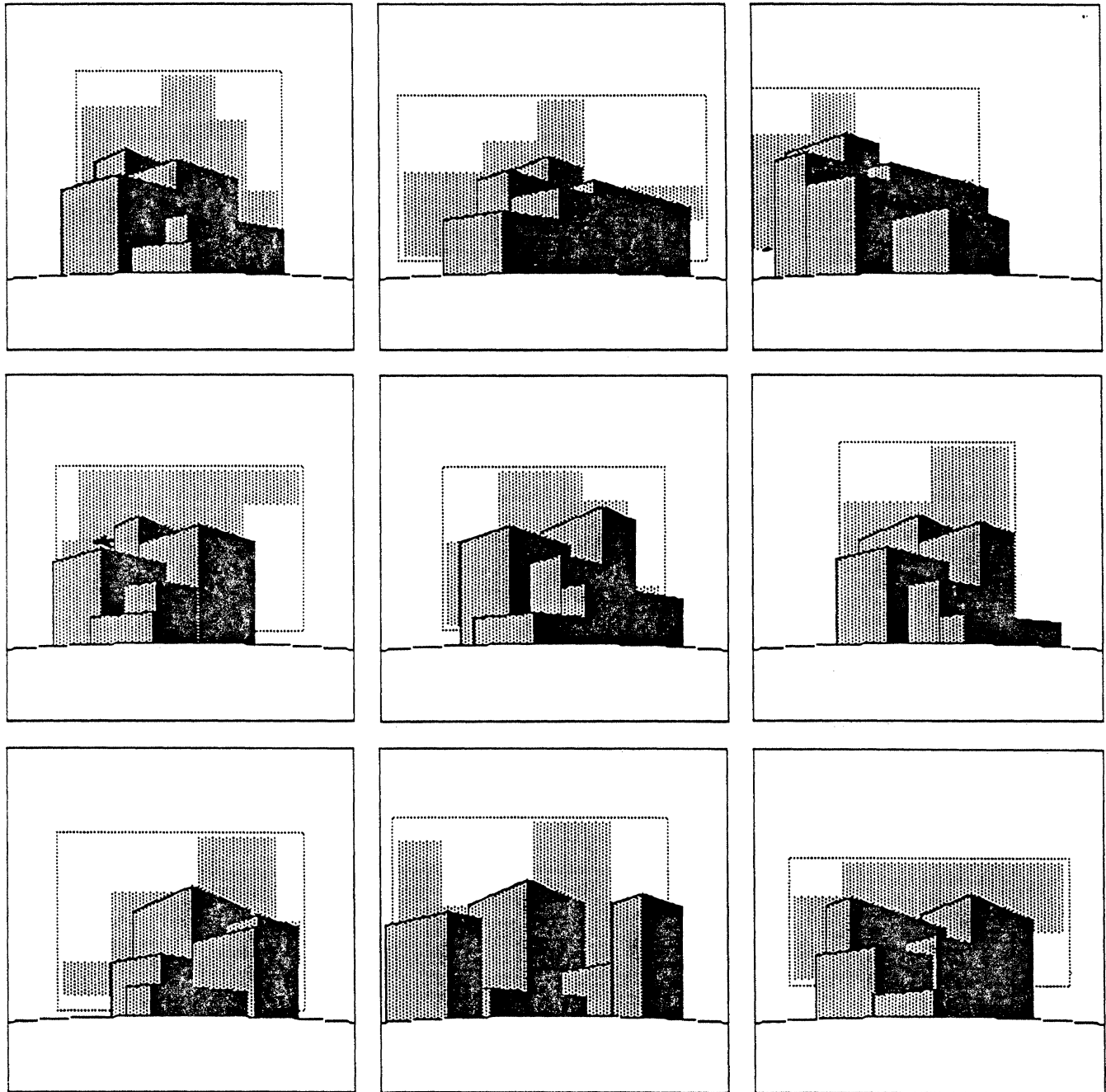


Figure 4.21

Figure 4.21 shows the nine satisfactory configurations (according to the reinterpreted criteria) as seen by the environ module. But the number of satisfactory configurations or the fact that an optimal one was found in this example are not, in themselves, valid criteria for evaluation. The next section will make use of the example shown in figures 4.20 and 4.20' to evaluate the generator in terms of its ability to avoid oscillation and to go beyond local maxima, as well as in terms of intuitive notions of versatility and robustness.

### 4.3 Why the Discursive Generator Works

Evaluating the performance of the discursive generator in a systematic way is difficult, because it is more of an illustration of an approach to designing design systems, than an attempt to solve a particular problem. Thus, if one takes the *reliability*<sup>8</sup> of the generator for granted, that is if one assumes that the generator is built in such a way as to guarantee an acceptable solution to the block arrangement problem, then its value would be in its *power* and its *dynamism*. In intuitive terms, power and dynamism would correspond to the versatility and the apparent spontaneity of the generator's behavior. On the other hand, if the architecture of the generator is seen to inherently favor a versatile and apparently spontaneous behavior, then the degree to which it converges on an acceptable "solution", instead of simply shuffling blocks around in a versatile and spontaneous (but apparently random) way, would be a measure of its worth.

Speaking of the "apparent" behavior of the discursive generator in these terms is probably no more useful than simply showing its output without an attempt at explicit evaluation. On the other hand, basing our evaluation on some ratio of "successful solutions" to "failures" would not be very significant because this particular block arrangement problem is probably better handled by some other, more algorithmic method if one's aim is simply to find solutions. What is interesting about the discursive generator is the way in which it reaches its "solutions", rather than their validity. Its performance is based on the idea that guarantees of dynamism are important at a more fundamental level than guarantees of optimum solutions<sup>9</sup>. Accordingly, I will now turn to an examination of the generator's behavior, pointing out its advantages over other approaches based on the kinds of moves it makes, and the typical problems which it avoids.

---

<sup>8</sup> See chapter 3 for definitions of the terms *reliability*, *power* and *dynamism* in the technical sense in which they are used in this thesis.

<sup>9</sup> The importance of a dynamic characteristic of activity goes beyond the field of design and has been the focus of recent efforts in other fields such as the study of autonomous robots [Agre 1988].



### 4.3.1 Discursive Generation as Search

It is interesting to think about the discursive generator's behavior as a process of searching through a solution space. This metaphor may not be a particularly appropriate one for several reasons, including the fact that it is not necessarily clear what the goal of the search is in this case. But looking at the discursive generator as a method of search can be quite revealing, specially in terms of a comparison with traditional search techniques.

Of course, a fundamental assumption in problems of search is that a complete and global view of the terrain to be searched (that is, of the solution space) is impossible. Another frequent assumption is that it is possible at all points in the solution space to have at least a relative idea of the value of the corresponding solution.

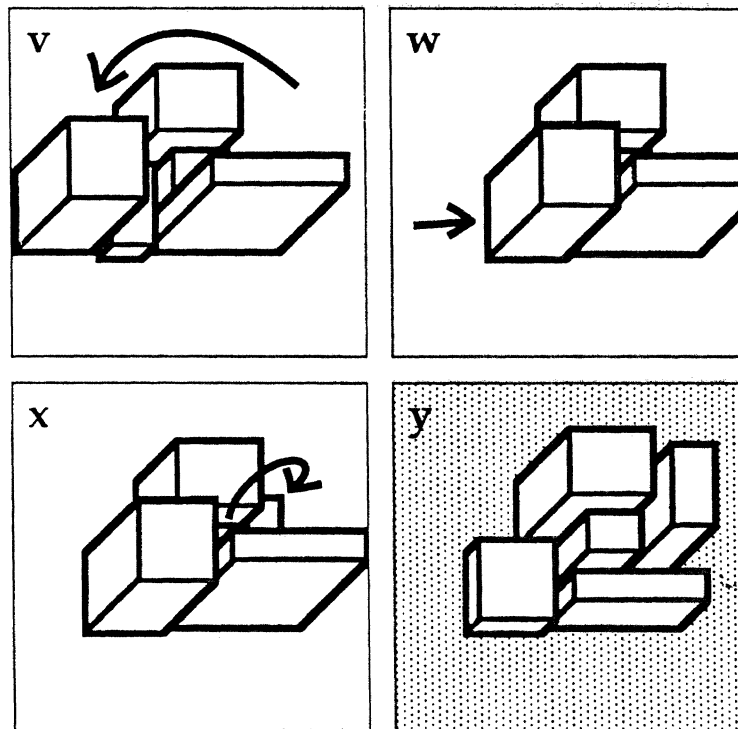


Figure 4.22

Many search techniques are based on gradient ascent, proceeding from each point in the solution space in the direction of greatest improvement, and consequently stopping at points where progress in any direction leads to a worse solution than the one corresponding to that point [Winston 1984]. Such points may be local maxima, and in order to reach a better solution the search must have a principled way of letting things get worse before they get better.

The discursive generator does not suffer from this problem because its progress is made with the aim of improving one criterion at a time and because it will take steps even when they lead to worse solutions. Some of the optimization techniques discussed in chapter 3 make moves in a continuum at the level of the document. This leads to the phenomenon of getting stuck in local maxima because of the impossibility of going from certain states of a document to other better states with continuous transformations, without having to travel through intermediate worse states. But gradient ascent algorithms which avoid this particular problem by including in their repertoire of moves ones which effect noncontinuous transformations, may still suffer from the problem of getting stuck in local maxima.

Imagine a solution space where each dimension of the space represents the value of a given configuration of blocks in terms of one criterion. Adjacent points in this space do not represent configurations which are physically similar, but ones which have equal or adjacent values for all the criteria. A gradient ascent algorithm will use some function to assign an over-all score to each point. These scores are not necessarily a continuous function in the solution space, and we can assume that the gradient ascent algorithm in question has the necessary repertoire of moves to trace a discontinuous path through it. The critical observation to make at this point is that a reasonable repertoire of moves will make it possible to visit only some of the points from any given point. If the gradient ascent algorithm, through a succession of better states, gets to a point from where it only knows how to reach worse states, it will be stuck. Note that this is in spite of the fact that its repertoire of moves may have allowed it to reach a more favorable state if it had taken a different path at certain points in its progress on condition of tolerating worse states. But it may even be the case that the only way of reaching a more favorable state is by going "downhill" from the local maximum where the

algorithm is stuck. Given the same repertoire of moves, the discursive generator will move to one of the points it can move to without the condition of over-all improvement. However, it will do so in what I have called a "principled" manner, meaning that it will try to achieve an improvement in terms of a given criterion chosen according to principles of focus. Through consolidation, the generator increases the likelihood that the move will be a favorable one, but it never requires a guarantee of improvement, even from the point of view of the criteria which dictated the move.

As an illustration of this phenomenon, consider frames v through y of figure 4.20, which are also shown in figure 4.22. In v, a block is moved to a new position where it is adjacent to only one other block. In w, in order to maximize the abutments of the configuration this same block is pushed into a position where it abuts with three blocks instead of one. On the other hand, it now completely overlaps with a fourth block. Comparing frames v and w from the point of view of all the modules would find v to be a better solution, yet the move in w is made. This move was proposed by both the Align and Corners modules, neither of which takes overlap into consideration. The overlap module, even though it gave a very negative evaluation to the proposal, has no counter-proposals because it sees no overlap to be undone<sup>10</sup>. Therefore the proposal which diminishes the current value of the design is adopted as the next state of the document. In x, the Overlap module, having missed a turn and having given a very low value to the current state of the design makes its comeback with a high "eagerness factor". (The fact that the result of evaluating the current state is negative, increases the eagerness factor of the Overlap module's move, because it is of type "remedial". See the section on the focus manager above.) The component-level focus switches from the bigger block to the one which is now covered by it, because it has the largest proportion of overlapping footprint area (again, as a result of the Overlap module's eagerness). Therefore, in x, this smaller rectangle is moved by the overlap module to a nearby corner, in collaboration with the Corners module. After heights are adjusted, the configuration in y proves to be the best one in the entire session.

---

<sup>10</sup> See the discussion of "status quo" moves in chapter 5.

Note that this feature of "allowing" a deterioration of the document's state which creates possibilities for subsequent improvement is much more likely to occur in overconstrained cases than in underconstrained ones (where it would be much less desirable) as evidenced by the two sample runs above: the underconstrained one of figure 4.6 and the overconstrained one of figures 4.20 and 4.20'.

Each move made in the discursive generator is concerned with a given quality of the design. As a result of consolidation, two or more modules may collaborate, making a move beneficial for more than one quality. But as already pointed out, this collaboration never involves an averaging operation, where two different criteria somehow "meet each other half way." A basic prerequisite of consolidation is that the resulting move be fully compatible with the parametric moves proposed by all the participants in the consolidation. As a result of the exclusivity of one module at each move, not only is the complexity the generator has to deal with reduced, but the generator's progress over the solution space becomes a succession of discreet steps along one dimension at a time, thus eliminating the problems associated with gradient ascent.

### 4.3.2 Low-Level Antidotes to Oscillation

A danger associated with optimization approaches, and with generative systems such as the discursive generator (as opposed to more syntactical approaches) is the possibility of oscillating. In chapter 3, I gave an example of how oscillation can effect algorithms using optimization techniques. In the case of the discursive generator, the danger is in encountering a situation where the arbitration process always leads to the same series of moves being repeated in a cycle.

Although explicit checks against oscillation may be necessary and valid for any systems of this type, the discursive generator has some inherent features which greatly reduce the likelihood of short-term oscillation without any explicit checks. The process recorded in figures 4.20 and 4.20' shows instances of repetition which do not degenerate into oscillation, thereby giving some

experimental evidence for the system's resistance to such phenomena. There are two main reasons why the discursive generator does not tend to oscillate. The first is that shifts of focus both at the level of components and at the level of the modules themselves reduce the likelihood of the same moves, involving the same components, occurring repeatedly. The second reason is due to consolidation. Moves which are the result of a cooperation between more than one module have inverses which are much less likely to be formed than the inverse of a single move (which is often the move itself applied in an inverse fashion.)

Let us take a closer look at two instances where oscillation is voided in the session depicted above.

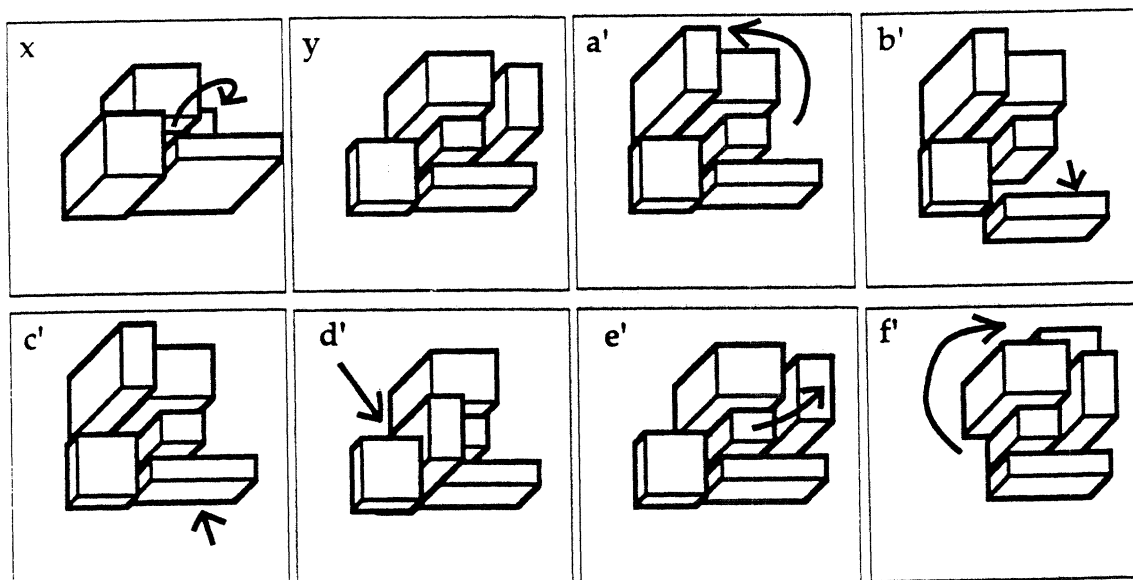


Figure 4.23

Note that frames y and e' show identical states of the document (see figure 4.23). This similarity is not a negative feature in itself (it could even be considered a positive feature due to the characteristics of exploration which it

implies). The danger it raises, however, is that of giving rise to an endlessly repeating cycle which would prevent further exploration. But it is clear that in this case, not only are the two similar configurations reached via different transformations, but they also give rise to different paths of exploration, even though they are identical.

It is important to point out that this divergence is not due to randomness. It is due to the fact that although there was a repetition in the solution space, that repetition did not coincide with one occurring in the space of focus.

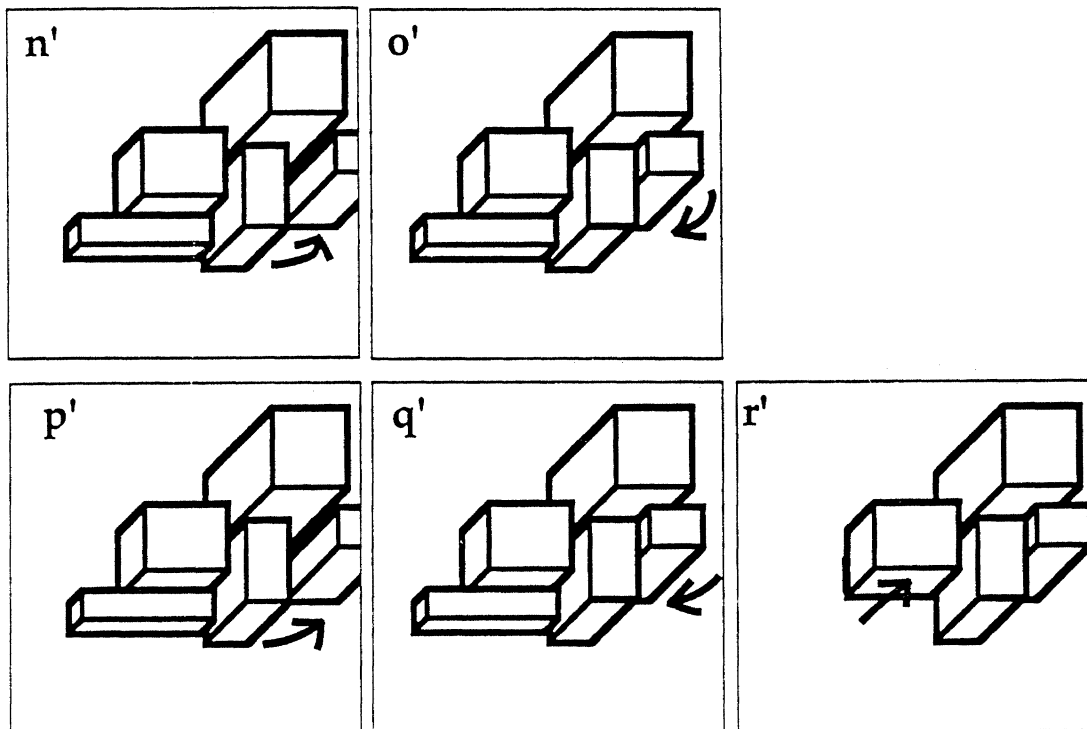


Figure 4.24

In other words, when in frame e' the state of the document was the same as in y, the focus of the system was not the same, and hence the paths diverged. This kind of repetition can be called "long-term" repetition, because it involves sequence of more than one or two transformations. Nested inside that sequence is an example of "short-term" repetition which is more

common in overconstrained cases such as this. Frames a', b' and c' constitute an A-B-A pattern which is immediately followed by a different state of the document. Here the reason oscillation is avoided is due to a component-level shift of focus. A more critical example of such a repetition occurs in frames n' through q' (Figure 4.24), where an A-B-A-B repetition takes place. This pattern is also followed by a new state in r'. A closer look at the history of proposed moves for frames p' and r' reveals that a shift of focus at the level of eagerness factors caused the divergence. Figure 4.25 shows the chosen proposal at p', made by the consolidation of two moves in the Align module (the one for vertical alignment and the one for horizontal alignment), and the chosen proposal at r' (which broke the potential cycle) made by the Corners module in collaboration with the vertical alignment move of the Align module.

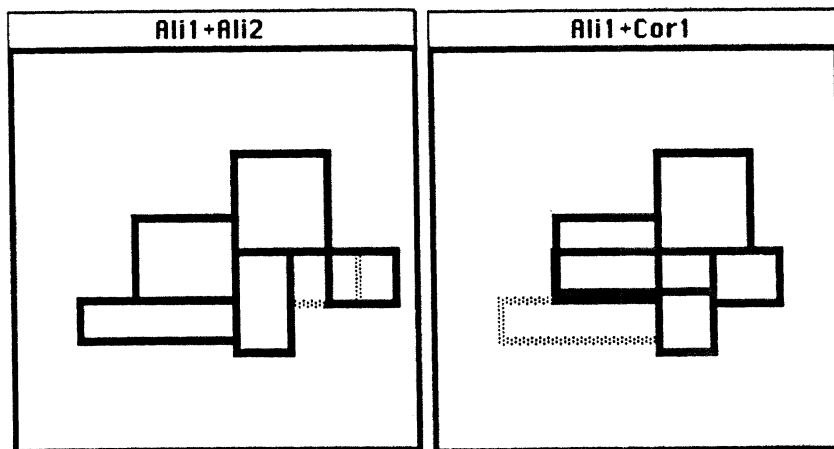


Figure 4.25

What I have called the discursive generator's "resistance to oscillation" can be stated more precisely as follows: the generator makes it more likely that cycles which do occur can be expected to be relatively long-term. This may seem problematic because long-term cycles are more difficult to detect than short-term ones. But because there are only a finite number of configurations, the ideal exploratory behavior would be one which cycles over a large number of favorable ones. The shorter such a cycle is, the more limited the exploration will be.

Once again, a comparison of the underconstrained sample run (figure 4.6) with the overconstrained one (figures 4.20 and 4.20') bears evidence to the fact that repetition is much less likely to occur in underconstrained cases than overconstrained ones where it can be useful as a means of exploration.

### 4.3.3 Redundancy for Robustness

The fact that different kinds of appraisal, know-how and seeing-as functions are grouped in modules, does not mean that their capabilities should be mutually exclusive. In the block arrangement application, for instance, the transformations proposed by the Corners module always have alignment as a side-effect. This kind of redundancy has several advantages. One of these is the practical advantage of making the job of creating modules independently of each other somewhat easier. A more significant advantage is that of adding robustness to the system.

If two modules share the capability of enhancing the quality of a design for a given criterion, then if one of them is not in a position to propose an appropriate transformation due to a low eagerness factor, or an unsatisfied condition, then the effect of the other module will partially take its place. Related to this is the fact that the know-how components of the modules are based on heuristics which are expected to improve the state of a design, and not on rules proven to do so. Therefore it is important to have a certain degree of robustness in order to increase the likelihood of good results. Consider, in this case, the redundancy between the Overlap rule (which consists in taking an overlapping block outside its area of intersection) and the Corners rule (which moves convex corners over concave ones). Note that the Overlap rule in this implementation is written in a very simplistic manner: it just moves an overlapping block to a different location where it might intersect another block. The move does not involve any checking for favorable locations to place the block being moved. On the other hand, simply by moving rectangles into corner locations, the Corners module often undoes intersections, inadvertently assisting the Overlap module. It too can actually create overlaps as well and, as illustrated in frames x and y above, this can



have a beneficial effect because it tends to take the configuration into a new region of the solution space.

But there are also obvious disadvantages in redundancy. I pointed out above that the thematic modules in the discursive generator are not modular in a strict sense. As a result they forfeit some of the advantages of modularity. One problem associated with redundancy is that by eliminating a module, one cannot simply eliminate the characteristics associated with it. These characteristics may also exist, in a marginal way, in other modules. If a decision is made to allow overlaps, then one would think that the elimination of the Overlap module would be sufficient. The fact is, however, that the Corners module will undo overlaps to a certain extent. And it is not always the case that redundancy can be eliminated easily (even if we were willing to sacrifice the advantages gained from redundancy by eliminating it). It is difficult to see how the notion of intersection can be extracted out of the corners module, because it is there as a side-effect. Chapter five contains some general suggestions about possible enhancements to the thematic modules which have some bearing on the issue of redundancy. But whatever the general framework one adopts, using it to implement concrete applications is always a design task in itself, and choosing an appropriate subdivision into modules can be something of an art.



## 5. Outline of Future Work

The relationship between developing elements of a theory of design activity and designing the discursive generator is very similar to the relationship between designing the generator and applying it to a concrete task such as block arrangement. The construction of one provides the inspiration for -and modifies- the other. During the course of this process of development some ideas came up which were not pursued, due to the limited scope and time-frame of the thesis. Some of these have to do with the methodology involved in using the discursive generator for a given task. The behavior of the generator depends very much on how its components (the appraisal function, the move rules, etc.) are written. On the other hand, there are phenomena which are not easily handled by the discursive generator as it now stands, but which are quite compatible with it and could conceivably be built on top of it. In this section I will briefly touch on ideas related to the method of using and the possibilities for expanding the discursive generator.

### 5.1 Some Principles for Writing Rules

The discursive generator was presented as a computational framework, a design activity "shell" or template. As such, it is by definition empty until used. By careful coordination of the activity of the different modules, it is possible to use the discursive generator in a manner which is completely counter to the theory on which it is based. But what would constitute using the generator as it was meant to be used?

Some of the important principles for the intended style of using the generator are captured in its architecture. For example, the fact that each module has its own seeing-as function implies that the subjective descriptions used by the modules should be different from each other in some significant way. In the block arrangement application, each way of seeing involved a loss of information which had the effect of a filter intervening between the appraisal function within the module and the document. This is most obvious in the Number module which only retained the number of blocks in the document. Another feature inherent in the architecture is the independence of appraisal

from the condition under which a move is suggested. This is not contradictory to the idea of the interdependence of appraisal and know-how as it might seem at first glance. That interdependence is manifested at another level of abstraction, namely within the arbiter where the appraisal of proposed next states produced by the know-how (that is by the move rules) influences the application of the know how, and vice versa. The idea of the independence of a move rule's condition for action and the module's appraisal function makes it possible, for instance, for the Environ module to be unhappy with the current state of the document, yet be unable to suggest a move because the document is not in a state (all blocks adjacent) where an adjustment of heights would be fruitful.

As mentioned in previous chapters, the idea of the incommensurability of criteria is not taken to its logical conclusion in the current implementation of the generator. If it were, then the evaluation of proposed next states would not involve any averaging function combining scores from different modules. This would also effect the criteria for knowing when to stop, and for choosing the "best" alternative of all the ones produced. In light of the incommensurability principle, the comparison of different alternatives would involve an independent criterion which might include some considerations related to the individual appraisal functions, but which would not just be a function of them. Therefore, the fact that the discursive generator does not have a built-in mechanism for knowing when to stop is one of its positive feature.

In the design of the generator, a decision was made to not imbed specific focus rules (the rules according to which eagerness factors are manipulated) in the system. In other words, just as different modules can be used for different tasks, different rules for shifts of focus could also be used to vary certain characteristics of behavior. A similar choice was made in a less systematic way regarding component-level focus. Each module contributes to a process by which the focus manager arranges the components of the document in order of their over-all focus, and therefore the process is independent of the architecture of the generator itself. On the other hand, some features were imbedded in the generator, and further study may reveal that they should be converted into empty slots to be filled according to the requirements of

specific applications. An example of such a case is in the choice of one of three types for moves: exploratory, proactive and remedial. By way of illustration, consider the following argument concerning the need, in certain cases, for a fourth type of move, the "status quo" move.

In the block arrangement application, the Overlap module only suggests moves when it detects an overlapping condition. Let us assume that at a certain cycle, the Align module suggests moving block A which happens to be currently intersecting block B, to any one of a set of points which would bring it into alignment with B. Some of these positions would still have A intersecting B, and others not. Similarly, the Overlap module might be proposing a move which would move A out of its overlapping position, to any one of a set of points, some of which would have it aligned with B, and others not. Through consolidation, these two moves could join in suggesting a position favorable for both modules. But if there is no overlap initially, then the overlap module will not propose moves, making it just as likely that the alignment proposed by the Align module also results in a new overlap. We could add a regular move rule to the know-how of the overlap module which would suggest destinations for block A which preserve the favorable current condition. The hope is that these proposals would be consolidated with other moves, and prevent an unnecessary deterioration of the current configuration. The problem with such an approach could be that these new move rules, when they are not consolidated with others, would result in chaotic moves, at the expense of more principled (but, in the short term, less favorable moves). Hence the need for "status quo" moves which have the property of having an effect only when consolidated with another move, and never on their own.

The fact that the discursive generator favors the simultaneity of moves is a feature which contributes to some of the positive aspects of its behavior, such as its resistance to oscillation. This bias for simultaneity is a particularly difficult feature to abstract out of the generator without necessitating either a fundamental revision or accepting a loss of functionality. This does not imply, of course, that simultaneity is fundamental to designing. Taking the position that it is would contradict the broad view of design activity adopted in this thesis. Rather, the observation illustrates how phenomena such as

simultaneity which I have (somewhat reluctantly) qualified as "higher level" would be imbedded in the context of the discursive architecture to achieve levels of sophistication which are taken for granted in different traditions of design. Taking this line of inquiry one step further would seem to lead us to question the fundamental role which this thesis assigns, for instance, to opportunism. But that would imply a confusion of different levels of abstraction. Opportunism can be a feature specific to a certain tradition or style of designing, and as such its status is similar to that of simultaneity, function segregation, analogy, and so on. But I have used concepts like opportunism at another level, as tools for the study and the production of design activity rather than as techniques and features of designing. Designing is opportunistic in nature in the sense that it involves a certain mode of interaction between many evaluative criteria, regardless of how that interaction is guided, manipulated or even suppressed over time. The inclusion of this last possibility, the possibility that phenomena like opportunism could fail to manifest themselves because they are suppressed, illustrates the point that such concepts have a formal role: We model the absence of manifest opportunism as the occurrence of processes causing its neutralization. The claim, then, is that the opportunistic model is a particularly powerful and generalizable one, rather than that it is a particularly "true" one<sup>1</sup>.

---

<sup>1</sup> The choice of the faculty of simultaneity over other sophisticated faculties in the design of the discursive generator was not arbitrary. It was chosen partly because it applies directly to characteristics of design moves rather than design artifacts. But more importantly it was chosen in order to show how the idea of the exclusivity of ways of seeing over an arbitrarily brief period does not necessarily exclude desirable faculties related to perception and representation. This last point seems to have a counterpart at a more practical level in that the bias for simultaneity counteracts the unidimensional nature of the generator's moves to the extent that it is possible to do so in a principled manner.

## 5.2 Designing Design Personas

At certain points in the thesis I argued (or implied) that some characteristics of good design which are typically considered universal are actually specific to certain traditions of design or to individual designers. One such case had to do with the idea of function sharing. Another related one might be the tendency, in architectural design, to favor analogical correspondences between different readings of a design component: An assembly hall is the central element in a building for ease of access, but also in virtue of symbolizing its central role in some organizational structure. The argument that all of these types of considerations are in a way optional, has the consequence of keeping them out of the architecture of the discursive generator itself, and delegating them to the level of specific modules.

Although they are not an integral part of designing as such, there are certain kinds of functionalities which are fundamental enough to be shared by many (or all) modules. Among these would be things like the faculty of recognizing (or aiming for) a relationship of the form "A is to B as C is to D", or other relationships such as inversion, equality and so on.

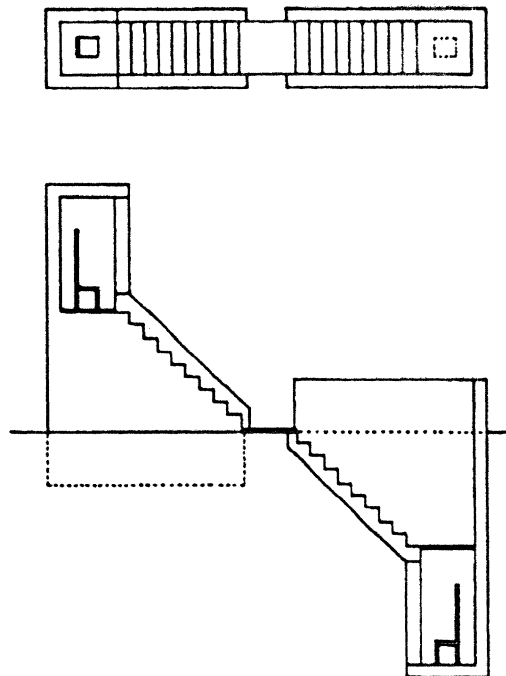


Figure 5.1

Figure 5.1 shows a particularly abstract and subtle example of inversion in a project by John Hejduk called "Judge-Stairs" [Hejduk 1986].

Perhaps a more advanced architecture for design generation should allow the building of functional objects shared by different modules. In fact there could be hierarchies of modules, where the highest-level ones would be modular in a sense that excludes redundancy, and only the lowest-level ones, having inherited their functionality from several more abstract modules would actually interact with the document.

A similar hierarchy seems appropriate for the kinds of seeing used as an interface between the document and the modules. This hierarchy would be operating in the opposite direction. For the hierarchy proposed at the level of know-how and appraisal functions of modules, inheritance would proceed from some mutually exclusive abstract faculties (inversion, etc.) towards the document. But for seeing-as functions the more basic different ways of seeing would provide a first level of filters applied to the document, and successive abstractions over those might result in higher-level filters corresponding to the seeing-as functions as they were used throughout the thesis. Note that this process does not imply the modeling of any "extraction" or "recovery" of information. Given the fact that the computational framework generated the document (supposedly in a computational medium), it has full knowledge of its contents.

The know-how used in the block arrangement application presented in this thesis was the minimum needed for the purposes of illustrating the ideas behind the discursive generator and characteristics of its behavior. But a more important reduction, again for the sake of being able to concentrate on illustration and demonstration of other points, occurred in the choice of "move rules" in the manner of production systems. These "if ... then ..." rules fail to capture in an elegant manner some important processes such as, for instance, reference to precedents. In a hypothetical example in chapter 2, the designer was said to see the potential for a courtyard in the plan view of a building. Typically this type of recognition involves knowledge of other buildings with courtyards. Capturing that in the form of simple rules is impractical. On the other hand, the discursive generator does contain a



strategy for bringing any knowledge obtained through precedents into play in the design activity. Rules were used to simulate more sophisticated implementations of know how in order to illustrate just such a strategy.



## 6. Conclusion and Overview

I have tried to illustrate in this thesis an approach to combining design principles in a generative framework. I have argued that neither rules of syntax nor constraint satisfaction techniques are powerful enough for the task of building a computational framework for design activity. If that is the case, then one is left with the question of how to bring "mixtures of simples" into play in a dynamic manner, with the aim of generating more complex artifacts. The discursive generator was presented as an answer to just that question. It was shown to have several positive features both in terms of its compatibility with the theory of design activity on which it is based and in the sense that it avoids many problems typical of other systems. In particular, the discursive generator was shown to avoid difficulties associated with search techniques based on gradient ascent, as well as problems related to optimization algorithms.

At one level, the contribution of this thesis is in the fact that it illustrates how an opportunistic approach to transforming a current state according to one of many criteria at a time, can be combined with the idea of focus (that is of a selective filtering of objects, intentions and systems of representation) in order to maintain dynamic and principled behavior in the face of complexity. At another level, the thesis suggests a new mode of creativity, where the designer would construct "design personas" as a collection of principles of appraisal and know-how.

It is tempting, at this point, to propose the discursive generator as a "designer's assistant". But I will yield instead to a temptation which is greater for me, that of suggesting that a designer might use the discursive generator (or rather a more advanced reincarnation of it) to build machines which design. Lucien Kroll writes: "Even the label 'Computer Aided Design' is misleading; to call it Computer Use in Design (CUD) would be more appropriate. For it is the architect (among others) who creates and not his pencil: a good pencil can help, but then so can a holiday in the mountains." [Kroll 1987]. Although it is easy to sympathize with this statement, specially if one has heard some of the more naive claims made about computers as

design aids, the position I will take is different from Kroll's. I will assume that design happens, and that the holiday in the mountain and the pencil are as much part of it as the architect. Imagine a process of indirect design, where the designer creates not a final artifact, but a "design persona", a set of principles and syntactical systems which, when brought into play in an environment such as the discursive generator, yield designs. Such systems would somehow be less than the artifacts they would produce (in that they could not be adequately substituted for them) but also, somehow, more (in that they contain many potential artifacts and, more importantly, another level of design). This duality is analogous to the duality characteristic of the term "weak" as used in mathematics. A weaker theory somehow states less but implies more.

The act of building a machine that generates a family of novel artifacts is a creative one. A computational framework such as the discursive generator can provide a level of guidance and organization, but using it in specific instances involves a special kind of designing. It is important to stress that the design task in question would not (typically) be that of replicating existing designs or mimicking designers, but the creation of new designs. As such it is a constructive task, involving some skills which are closely related to ones traditionally considered important in the field in question, as well as other special design abilities. This approach involves an unprecedented interaction between conceptual/analytical issues and systems of production. It implies, on the one hand a high degree of explicitness in generative principles and hence of control over the process of production and, on the other, a loss of direct control due to the mediation of the computational framework. In that sense a radical transformation of the current tradition of design is involved in such a process, just as the current tradition of architectural design, for example, involved a radical (though gradual) break with that of the master mason.

Kroll says: "Some enthusiasts have become deeply involved in one of these directions and, seduced by the game, have come almost to believe that their creation has taken on a life of its own, like the mythical sculptor who became so engrossed in his paternal fantasy that he asked his carved figure, 'But why don't you speak?'" But the problem of the designer using the discursive

generator is the inverse one. Because of its guarantees of dynamism and because of the nature of the computational framework, the generator will speak. The challenge, of course, would be to have it say things worth being engrossed in.

The discursive generator represents an approach to building design systems based on fundamental properties of dynamism and spontaneity as opposed to guarantees of optimization or constraint satisfaction. But this approach initially entails certain necessary setbacks. Calvino expresses the uncertainty resulting from dynamism for the primordial organism:

So the characteristics that determine my interior and exterior form, when they are not the sum or the average of the orders received from father and mother together, are orders denied in the depth of the cell, counterbalanced by different orders which have remained latent, sapped by the suspicion that perhaps the other orders were better. So at times I'm seized with uncertainty as to whether I am really the sum of the dominant characteristics of the past, the result of a series of operations that produced always a number bigger than zero, or whether instead my true essence isn't rather what descends from the succession of defeated characteristics, the total of the terms with the minus sign, of everything that in the tree of derivations has remained excluded, stifled, interrupted: the weight of what hasn't been weighs on me, no less crushing than what has been and couldn't not be. [Calvino 1969]

With the adoption of a dynamic approach to design generation comes a sacrifice, at least at some primitive level, of the deterministic and rational order associated with formal rules. But in terms of the current use of computers to design, this sense of order is still an illusion resulting from the act of confusing the systematic analysis of ordered artifacts with synthesis from principles of order. The discursive approach to design generation provides a principled and dynamic basis for design activity. In its context an order can eventually be constructed which would effect a closure of the cycle of sophistication and spontaneity.



## Bibliography

Agre, Philip 1988, "The Dynamic Structure of Everyday Life," PhD Dissertation Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science.

Alexander, Christopher 1964, *Notes on the Synthesis of Form*. Cambridge: Harvard University Press.

Archea, John 1987, "Puzzle-Making: What Architects Do when No One Is Looking," in Kalay, Yahuda Ed., *Computability of Design*. New York: John Wiley & Sons.

Berwick, Robert 1987, *Principle-Based Parsing*. Massachusetts Institute of Technology Artificial Intelligence Laboratory Technical Report 972.

Bleau, Eve and Edward Kaufman Eds. 1989, *Architecture and its Image*. Montreal: Canadian Center for Architecture.

Brooks, Rodney 1990, "Elephants Don't Play Chess," in Maes, P. Ed., *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*. Cambridge: The MIT Press.

Calvino, Italo 1969, *t zero*. New York: Harcourt Brace Jovanovich.

Ching, Francis 1979, *Architecture: Form, Space and Order*. New York: Van Nostrand Reinhold.

Cortazar, Julio 1966, *Hopscotch*. New York: Pantheon/Random House.

Eastman, Charles 1973, "Automated Space Planning," in *Artificial Intelligence*, 4.

Evans, Robin 1986, "Translations from Drawing to Building," in AA Files 12.

Fleisher, Aaron 1990, "Grammatical Architecture?" Unpublished manuscript, Massachusetts Institute of Technology Department of Urban Studies and Planning.

Fletcher, Banister 1975, *A History of Architecture*. New York: Charles Scribner's Sons.

Flemming, Ulrich 1987, "The Role of Shape Grammars in the Analysis and Creation of Designs," in Kalay, Yahuda Ed., *Computability of Design*. New York: John Wiley & Sons.

- Goldman, Alvin 1986, *Epistemology and Cognition*. Cambridge: Harvard University Press.
- Graves, Michael 1977, "The Necessity of Drawing: Tangible Speculation," in *Architectural Design*, 47, No 6.
- Gross, Mark, Stephen Ervin, James Anderson and Aaron Fleisher 1987, "Designing with Constraints," in Kalay, Yahuda Ed., *Computability of Design*. New York: John Wiley & Sons.
- Habraken, John 1985, *The Appearance of the Form*. Cambridge: Awater Press.
- Heath, Tom 1989, "Lessons from Vitruvius," in *Design Studies* Volume 10, Number 4.
- Hejduk, John 1986, *Victims*. London: The Architecture Association.
- Herberg, Klaus 1983, *The Decorated Diagram: Harvard Architecture and the Failure of the Bauhaus Legacy*. Cambridge: The MIT Press.
- Hernion, Max 1978, "Automatic Space Planning: A Postmortem?" in Latombe, J. Ed., *Artificial Intelligence and Pattern Recognition in Computer Aided Design*. Amsterdam: North Holland Publishing Company.
- Hillier, Bill and Adrian Leaman 1973, "How is Design Possible?: a Sketch for a Theory," in *DMG/DRS Journal*, Vol 8 No 1.
- Hubbard, William 1986, *Complicity and Conviction: Steps toward an Architecture of Convention*. Cambridge: The MIT Press.
- Johnson, Timothy, et al. 1970, "IMAGE: An Interactive Graphics-Based Computer System for Multi-Constrained Spatial Synthesis", Massachusetts Institute of Technology Department of Architecture.
- Klotz, Heinrich Ed. 1989, *20th Century Architecture*. New York: Rizzoli.
- Koutamanis, Alexander 1990, "Development of a Computerized Handbook of Architectural Plans," PhD Dissertation, Delft University of Technology.
- Kroll, Lucien 1987, *An Architecture of Complexity*. Cambridge: The MIT Press.
- Le Corbusier 1987, *Le Corbusier: une Encyclopedie*. Lucan Jacques Ed., Paris: Centre Georges Pompidou.



Lakatos, Imre 1976, *Proofs and Refutations: The Logic of Mathematical Discovery*. New York: Cambridge University Press.

Markus, John 1968, *Sourcebook of Electronic Circuits*. New York: McGraw Hill.

Mitchell, William 1990, *The Logic of Architecture: Design, Computation and Cognition*. Cambridge: The MIT Press.

Pereira, Luis Monis 1978, "Artificial Intelligence Techniques in Automatic Layout Design," in Latombe, J. Ed., *Artificial Intelligence and Pattern Recognition in Computer Aided Design*. Amsterdam: North Holland Publishing Company.

Poe, Edgar Allan 1842, "Three Sundays in a Week," in Quinn, A.H. and E. H. O'Neil Eds., *The Complete Tales and Poems of Edgar Allan Poe*. 1989, New York: Dorset.

Poe, Edgar Allan 1846, "The Philosophy of Composition," in Harrison, J. Ed., *The Complete Works of Edgar Allan Poe*. 1965, New York: AMS Press.

Pohl, J., L. Myers, A. Chapman, J. Snyder, H. Chauvet, J. Cotton, C. Johnson and D. Johnson 1991, "ICADS Working Model Version 2 and Future Directions" California Polytechnic State University, San Luis Obispo, Design Institute Report CADR-05-91.

Ramsey, Charles and Harold Sleeper 1956, *Architectural Graphic Standards (5th Edition)*. New York: John Wiley & Sons.

Rossignac, Jarek, Paul Borrel and Lee Nackman 1989, "Interactive Design with Sequences of Parametrized Transformations," in Akman, V., P.J.W. ten Hagen, P.J. Veerkamp Eds., *Intelligent CAD Systems II: Implementation Issues*. Berlin: Springer-Verlag.

Rowe, Peter 1987, *Design Thinking*. Cambridge: The MIT Press.

Schon, Donald and Glen Wiggins 1990, "Kinds of Seeing and Their Function in Designing," Forthcoming, University of Manchester.

Simon, Herbert 1975, "Style in Design," in Eastman, C. Ed., *Spatial Synthesis in Computer-Aided Building Design*. New York: John Wiley & Sons.

Simon, Herbert 1981, *Sciences of the Artificial (2nd Edition)*. Cambridge: The MIT Press.

Stiny, George 1990, "What Designers do that Computers Should," in McCullough, M, W Mitchell and P Purcell, Eds. 1990 *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Era*. Cambridge: The MIT Press.

Stiny, George and William Mitchell 1978, "The Palladian Grammar," in *Environment and Planning B* Volume 5, Number 1.

Tzonis, Alexander 1972, *Towards a Non-Oppressive Environment*. Boston: i Press.

Tzonis, Alexander and Liane Lefaivre 1987, *Classical Architecture: the Poetics of Order*. Cambridge: The MIT Press.

Ulrich, Karl Thatcher 1987, "Computation and Pre-parametric Design," PhD Dissertation, Massachusetts Institute of Technology Department of Mechanical Engineering.

University of Chicago at Illinois 1986, *The Chicago Tapes*. New York: Rizzoli.

Veerkamp, Paul 1990, "Multiple Worlds in an Intelligent CAD System," Technical Report Number CS-R9057, Center for Mathematics and Computer Science, Amsterdam, the Netherlands.

Valery, Paul 1919, *Introduction a la Methode de Leonard De Vinci*. Paris: Nouvelle Revue Francaise.

Weinzapfel, Guy and Steven Handel 1975, "IMAGE: Computer Assistant for Architectural Design," in Eastman, C. Ed., *Spatial Synthesis in Computer-Aided Building Design*. New York: John Wiley & Sons.

Winston, Patrick 1984, *Artificial Intelligence (2nd Edition)*. Reading, Massachusetts: Addison Wesley.

Wittgenstein, Ludwig 1971, *Philosophische Bemerkungen*. Oxford: Basil Blackwell.