"Ill-Structured Diagrams" for Ill-Structured Problems

VINOD GOEL
University of California, Berkeley

Abstract

While the distinction between well-structured and ill-structured problems is widely recognized in cognitive science, it has not generally been noted that there are often significant differences in the external representations which accompany the two problem types. It is here suggested that there is a corresponding distinction to be made between "well-structured" and "ill-structured" representations. Such a distinction is used to further differentiate diagrams into finer-grained types, loosely corresponding to sketches and drafting type diagrams, and it is argued that ill-structured, open-ended problems, like the preliminary phases of design problem solving, *need* "ill-structured" diagrammatic representations. Data from protocol studies of expert designers are used to support the thesis.

1.0 Introduction

Cognitive science has made considerable progress in understanding how certain well-structured problems are solved and the role external representations play in such solutions (Newell & Simon, 1972). A typical example of a well-structured problem is the game of chess (see however, Simon (1973) for an alternate view). In chess, the start state is specified, as is the goal state and the set of legal transformations, (though generating/ selecting the "best" transformation at any given point is a non-trivial task). The representation of the task (whether it be in internal or external memory) is also "well-structured" in that it is clear what state is being instantiated, by virtue of what it is that state, what states of affairs are being referred to, and what the set of legal transformations from one state to another state are.

While it has been frequently noted that many of the problems that we confront in life are not well-structured, it has generally not been appreciated that the representations which often accompany the solutions to such problems are also not "well-structured" in the above sense. In fact, the predicates 'ill-structured' and 'well-structured' have not to my knowledge been applied to representations. The major differentiating criteria for representations has been, and continues to be, the diagrammatic (or pictorial or imagistic) and propositional (or linguistic) dimension (Kosslyn & Pomerantz, 1981; Larkin & Simon, 1987; Paivio, 1977; Pylyshyn, 1981; Rey, 1981; Simon, 1972).

While I do not deny the importance of the diagrammatic and propositional distinction, I will here focus on and argue for the "ill-structured" and "well-structured" distinction. In fact, I will use the "ill-structured" and "well-structured" distinction to further differentiate diagrammatic representations into finer-grained types.[1] Informally, and as a first pass, one might understand the "ill-structured" and "well-structured" distinction in terms of the difference between fast free-hand sketches and formal, box-like drafting diagrams, where sketches correspond to the former, and drafting diagrams to the latter.[2] A more formal statement follows in section 2.0.

The goal of this paper is to differentiate "well-structured" diagrammatic representations from "ill-structured" diagrammatic representations and to show that some ill-structured problems require "ill-structured" diagrammatic representations to prevent premature crystallization of ideas and to facilitate the generation and exploration of alternate solutions. This is a very brief summary of work reported in full elsewhere (Goel, 1991; Goel, 1992a; Goel & Pirolli, in press).

2.0 Differentiating "Well-Structured" & "Ill-Structured" Diagrams

Consider the diagrammatic representation in Figure 1. It depicts two states and a transformation in a game of chess. The representation and the symbol system it belongs to have the following seven properties:[3]

(p1) *Syntactic Disjointness:* Each mark (token) belongs to at most one character (type). Thus for example, no tokens of the type 'rook' belong to the type 'queen'.

(p2) *Syntactic Differentiation:* It is possible to tell which character a mark belongs to. So given the

[1]Thus the "ill-structured" and "well-structured" distinction would seem to be orthogonal to the diagrams/propositional categories. This, however, is a complex issue discussed elsewhere (Elgin, 1983; Goel, 1992b; Goodman, 1976).

[2]Strictly speaking this is not true. But it is a useful starting point.

[3]When talking about the syntactic elements of a symbol system I, following (Goodman, 1976), prefer the vocabulary of 'marks' and 'characters'. The characters are the types and should be thought of as equivalence-classes of marks, where marks are tokens. The reader should feel free to substitute the 'type/token' vocabulary as necessary. Also, a character here is to be more broadly construed than an element of the alphabet. A character is taken to be any symbolic expression, whether simple or complex. It can be anything from a single mark, (as in the case of a letter of the alphabet), an utterance, gesture, etc. to -- in the limiting case -- the whole work (e.g. paintings, sketches, novels, plays).

characters 'queen' and 'rook' and a token of the character 'rook' it is possible to tell which character it does and does not belong to.

(p3) *Unambiguity:* Every character has the same referent in each and every context in which it appears. Thus no 'bishop' refers to a knight regardless of context.

(p4) *Semantic Disjointness:* The classes of referents are disjoint; i.e. each object referred to belongs to at most one reference-class. So for example, no pawn belongs to the class of rooks.[4]

(p5) *Semantic Differentiation:* It is possible to tell which class a particular object belongs to. Thus given a king and two classes of objects, one could determine which class, if any, the king belongs to.

(p6) The rules of transformation of the system are well specified. Thus for example, there is no question as to what does and does not constitute a legal move for a bishop.

(p7) The legal transformations of the system are such that these properties are preserved at each and every state.

The first five of these properties (p1-p5) are adopted from Goodman (1976). The reader is referred to Goodman (1976), Elgin (1983) and Goel (1992b) for a more complete discussion.

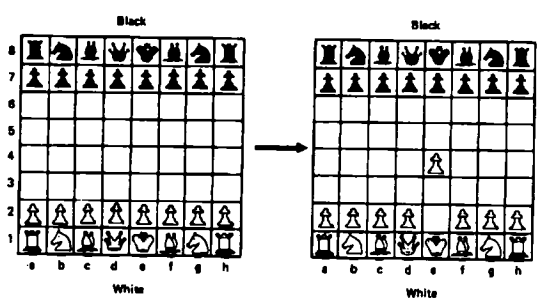

Figure 1: States and transformation from a game of chess (reproduced from Rich (1983, p.65)*)*

Six of these seven properties of symbol systems are actually presupposed by the notion of a computational problem space (Goel, 1991). The satisfaction of properties p1 and p2 is necessary for there to be a discernable fact of the matter as to what state is instantiated. Satisfaction of properties p3 and p5 is necessary for there to be a discernable fact of the matter as to what state of affairs is being referred to.[5] Property p6 is necessary to constrain the class of possible transformations, while property p7 is necessary to maintain the above properties during the transformation of one state to the next.

[4]This is of course true only in the vocabulary of chess, narrowly defined. In the larger context, a pawn also belongs to the class of chess pieces, the class of material objects, etc. But this is consistent with the point being made here.

[5]Property p4 is necessary to go from the referent, to the referring state. But it is not clear whether this is *necessary* for the notion of a problem space.

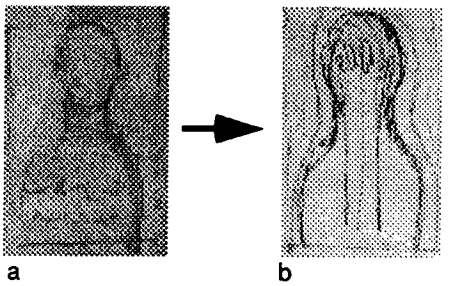

Figure 2: States and transformation from early part of a graphic design session

In contrast, consider the diagrammatic representations in Figure 2 extracted from the early part of a graphic design problem solving session. They belong to the symbol system of sketching and differ from the representations in Figure 1 with respect to each of the above seven properties. In fact, they fall on the opposite extreme with respect to each of the seven properties (p1-p7):

(p1') *Failure of Syntactic Disjointness:* Each mark (token) may belong to many characters (types) at the same time. That is, in the absence of any agreement as to the constitutive versus contingent properties of marks there may be no fact of the matter as to which equivalence-class they belong to. Thus for example, what equivalence-class does mark *a* in Figure 2 belong to? Do marks *a* and *b* belong to the same equivalence-class? There may be no agreed upon answers to these questions.

(p2') *Failure of Syntactic Differentiation (through density):* Because the symbol system of sketching allows for a dense ordering of characters or types (i.e. between any two characters there is a third), it is not always possible to tell which type a token belongs to. So, for example, even it we agree that the mark *a* in Figure 2 belongs to only one equivalence-class, it may not be possible to tell which of several classes it does or does not belong to.

(p3') *Ambiguity:* Characters do not have the same referent in each and every context in which they appear. For example, the mark *b* in Figure 2 was interpreted as a human head and later reinterpreted as a light bulb.

(p4') *Failure of Semantic Disjointness:* The classes of referents are not disjoint; i.e. each object referred to may belong to many reference-classes. So for example, the human-figure referred to by the character *a* may belong to the class of humans and the class of students.

(p5') *Failure of Semantic Differentiation:* The system of sketching allows for a dense ordering of reference-classes. When this is the case, it is not possible to tell which class a particular object belongs to. For example, in a perspective drawing of a human figure, every height of the mark would correspond to a different class of heights of human figures in the world, and these classes of heights are of course densely ordered. In such a case it would not be

possible to tell which class a particular human height belongs to.

(p6') The system of sketching has no well-specified rules for transforming one state into another. There is no transformation of the mark *b* which would be "incorrect" or "illegal".

(p7') As the properties p1-p6 are not present to begin with, they are not preserved in the transformation of the system from one state to the next.

Having defined "ill-structured" and "well-structured" representations as such, I will henceforth dispense with the scare quotes.

It makes sense that the representations which underlie well-structured problems (e.g. cryptarithmetic, Moore-Anderson task, Tower of Hanoi, 8-Puzzle problem, checkers, etc.) should be well-structured (by virtue of having properties p1-p7). After all, if these properties were absent then the states, operators, and evaluation functions could not be specified and the problem would not be a well-structured problem.

There is however no similar reason for ill-structured problem spaces to be accompanied by representations belonging to well-structured symbol systems. The fact that the problem spaces are ill-structured means that states, operators and evaluation functions are not defined, thus there is little need for the information which specifies them to be actually present. In fact, not only is there no compelling reason for representations accompanying ill-structured problems to be well-structured, there actually seems to be a case to be made to the effect that they *need* to be ill-structured to facilitate certain cognitive processes. This point is argued for in the next sections with some results from design problem solving.

3.0 The Role of Ill-Structured Diagrams in Design Problem Solving

Two empirical studies of design problem solving were conducted. The first examined some of the cognitive processes involved in design problem solving while the second focused on the impact on these cognitive processes when the symbol systems the designers were allowed to use were manipulated along the well-structured and ill-structured dimensions. A few of the results of each study are noted below. A complete discussion appears in Goel (1991).

3.1 Study I: The Structure of Design Problem Spaces

A study was conducted to differentiate design problem spaces from non-design problem spaces and to develop an information processing model of some of the cognitive processes involved in design problem solving (Goel, 1991; Goel & Pirolli, in press). Verbal protocols from 12 expert designers from the disciplines of architecture, mechanical engineering, and instructional design were compared to several cryptarithmetic and Moore-Anderson task protocols from (Newell & Simon, 1972). An interesting story which differentiated design (ill-structured) problem spaces from non-design (well-structured)[6] problem spaces emerged. Several aspects of the design problem space relevant to the present discussion are noted in Figure 3 and discussed below.

First, lets note some obvious, but often overlooked points. The input to the design process is a design brief. The output is a set of contract documents. Design involves the transformation of the design brief into a set of contract documents. Both the design brief and contract documents are representations, thus designers manipulate representations of the world rather then the world itself. The design brief is consistent with a number of alternative solutions. The contract documents are consistent with only one of these alternatives.

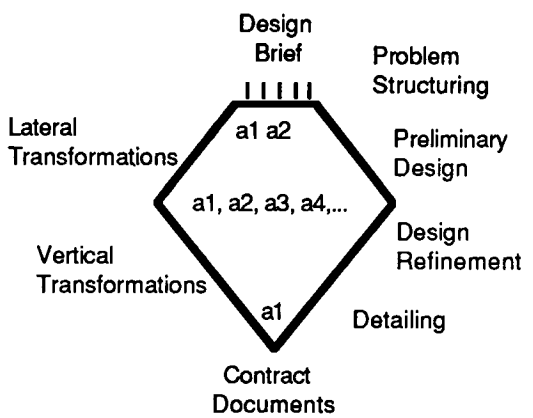

Figure 3: The design problem space. See text.

Some of the more substantive results were the following: The development of a design solution has several distinct phases. Four of these phases are problem structuring, preliminary design, refinement, and detailing.[7] These phases differ with respect to the type of information dealt with, the degree of commitment to generated ideas, the level of detail attended to, and the number and types of transformations engaged in.

Problem structuring is the process of retrieving information from long-term memory and external memory and using it to construct the problem space; i.e. to specify start states, goal states, operators, and evaluation functions. Problem structuring relies heavily on the client and design brief as a source of information, considers information at a higher level of abstraction, makes fewer commitments to decisions, and involves a higher percentage of add and propose operators (Goel, 1991).

Preliminary design is a classical case of creative, ill-structured problem solving. It is a phase where alternatives are generated and explored. Alternative solutions are not however fully developed when generated. They emerge through incremental transformations of a few kernel ideas. These kernel ideas are images, fragments of solutions, etc. to *other* problems which the designer has en-

[6]This is not to suggest that all non-design problem spaces are well-structured.

[7]While these categories may seem trivial, they do constitute a significant claim about the design problem space because they are not found in at least some nondesign problem spaces (Goel, 1991; Goel & Pirolli, in press).

countered at some point in his life experience. Since these "solutions" are solutions to other problems which are being mapped onto the current problem, they are, not surprisingly, always out of context or in someway inappropriate and need to be modified to constitute solutions to the present problem (Goel, 1991).

This generation and exploration of alternatives is facilitated by the abstract nature of information being considered, a low degree of commitment to generated ideas, the coarseness of detail, and a large number of lateral transformations. A lateral transformation is one where movement is from one idea to a slightly different idea rather than a more detailed version of the same idea. They are necessary for the widening of the problem space and the exploration and development of kernel ideas (Goel, 1991).

The *refinement* and *detailing phases* are more constrained and structured (though still very different from puzzle games). They are phases where commitments are made to a particular solution and propagated through the problem space. They are characterized by the concrete nature of information being considered, a high degree of commitment to generated ideas, attention to detail, and a large number of vertical transformations. A vertical transformation is one where movement is from one idea to a more detailed version of the same idea. It results in a deepening of the problem space (Goel, 1991).

It was also noted that the preliminary design phases were accompanied by ill-structured representations (belonging to the symbol system of sketching), while the refinement and detail phases were accompanied by more well-structured representations (belonging to the system of drafting). A second study was conducted to investigate the role played by ill-structured diagrams in the preliminary design phase.

3.2 Study II: The Role of Sketching in Preliminary Design Problem Solving

In a second protocol study the following four of the seven properties of ill-structured representations were examined and manipulated: (p1') failure of syntactic disjointness, (p2') failure of syntactic differentiation, (p3') ambiguity, and (p5') failure of semantic differentiation. It was predicted that when these properties are absent (i.e. when the symbol system is well-structured) the number of lateral transformations would be hampered. The underlying rationale was that properties p1' and p3' facilitate lateral movement by allowing multiple interpretation of characters (states in the problem space) while properties p2' and p5' facilitate lateral movement by allowing for overlapping (or closely ordered) characters (states) and ideas (Goel, 1991; Goel, 1992b).

Subjects & Design: Nine expert designers from the disciplines of industrial design and graphic design were engaged in two (one hour) problem solving sessions while the symbol systems they were allowed to use were manipulated along the dimensions of ill-structured and well-structured representations. In the one case subjects were allowed to use an ill-structured symbol system with properties p1', p2', p3', and p5'. In the other case they were requested to use a well-structured symbol system with properties p1, p2, p3, and p5.

Manipulation of Symbol Systems: The manipulation of symbol systems was through the manipulation of drawing tools and media. While it is true that there is a logical distinction to be made between drawing tools and media on the one hand and symbol systems on the other, in practice this distinction is often collapsed, especially in the case of computational interfaces. A computational interface provides not only a tool and a medium for drawing, it also specifies a symbol system by easily facilitating certain marks and operations and discouraging or even disallowing others.

In one session each designer was allowed to use the tools, media, and symbol systems of his/her choice. They invariably chose to use paper and pencils and did a lot of sketching. In the second session they were requested to use a computational interface. Specifically, they were asked to use a subset of the drawing package MacDraw[8] (version 1.9.5; with the freehand tool turned off and the grid turned on) running on a Mac II[9] with a large two-page monitor. MacDraw is not a sketching tool; it is a restrictive subset of a drawing or drafting tool. It only allows one to make precise lines, boxes, and circles. The subjects all used sophisticated computational drawing tools as part of their jobs and so were proficient with MacDraw.

The expectations were that free-hand sketching would be used to generate substantially ill-structured representations while the representations generated with MacDraw would be substantially well-structured. It was also expected (as noted above) that ill-structured representations would result in more lateral transformations.

Task Descriptions: There were three graphic design tasks and two industrial design tasks. The graphic design tasks were to design a (i) poster for the new cognitive science program at UC-Berkeley, (ii) a poster for the Shakespeare Festival at Stratford-on-Avon, and (iii) a poster promoting the city of San Francisco. The industrial design tasks required the design of (i) a desk time piece to commemorate Earth Day, and (ii) a toy to amuse and educate a 15-month old toddler.

Informal Overview of Data: Informally, the difference between the two cases seems to be the following: In free-hand, when a new idea is generated, a number of variations of it quickly follow. The variations expand the problem space and are necessary for the reasons noted earlier. One actually gets the sense that the exploration and transformation of ideas is happening on the paper in front of one's eyes as the subject moves from sketch to sketch. Indeed, designers have very strong intuitions to this effect.

When a new idea is generated in MacDraw, its external representation (in MacDraw) serves to fixate and stifle further exploration. Most subsequent effort after the initial generation is devoted to detailing and refining the same idea. One gets the feeling that all the work is being done internally and recorded after the fact, presumably be-

[8] 'MacDraw' is a registered trademark of Apple Computer, Inc..

[9] 'Mac II' is a registered trademark of Apple Computer, Inc..

cause the external symbol system cannot support such operations.

Coding Scheme: A coding scheme was developed to get at the differences more formally. A subset is presented here. The protocols were first broken up into episodes along the lines of alternative solutions (which correlated with drawings on a one to one basis). This resulted in mid-size chunks averaging 130 sec per episode for sketching or the ill-structured representations and 178 seconds per episode for MacDraw or the well-structured representations. Sketching averaged 19 episodes per session while MacDraw averaged 14 episodes per session.

Given this initial breakdown, the origin and evolution of each episode was traced out (see Figure 4). In particular we asked the following three questions with respect to each episode (see Figure 4): (i) where did the episode come from; (ii) how did it come about; and (iii) did the drawings accompanying the episode undergo any reinterpretations? There were two options for the origin of an episode, *long-term memory* or a *previous episode*. When the origin was a *previous episode*, it was asked whether the current episode was a *variation* of a previous episode (i.e. similar but not identical) or *identical* to a previous episode plus or minus a specifiable difference.

- where did episode come from:
  - LTM
  - Previous episode
    - Identical +/-
    - Variation
- how did episode come about:
  - New generation
  - Transformation
    - Lateral
    - Vertical
- did drawing accompanying episode undergo reinterpretation

Figure 4: Subset of coding scheme

There were also two possibilities with respect to how an episode came about. It could either have been a *new generation* or a *transformation*. Anything coded as coming from *long-term memory* was considered to be a *new generation*. If the origin of the episode was a *previous episode*, then it was coded as a *transformation*. Two types of transformations were coded for, *lateral* and *vertical*. As already noted, *lateral transformations* change or modify an idea into another related but distinctly different idea while *vertical transformations* reiterate and reinforce an existing idea through explication and detailing.

Hypotheses: It is necessary to measure two things: (1) How are the two symbol systems being used with respect to the ill-structured/well-structured properties? (2) How does this impact the number of lateral transformations and reinterpretations? The hypotheses with respect to (1) are the following:

(H1) Free-hand sketching is syntactically more dense than MacDraw.

(H2) Free-hand sketching is semantically more dense than MacDraw.

(H3) Free-hand sketching is less unambiguous and/or disjoint than MacDraw.[10]

The specific hypothesis with respect to (2) is the following:[3,7]

(H4) Well-structured representations will hamper the exploration and development of alternative solutions (i.e. lateral transformations) and force early crystallization of the design.

Results:[11] Sequences of episodes which received a *variation* rating were considered to be more densely ordered than those which received some other rating. Measured as such, the ordering of episodes (or alternative solutions) is significantly denser in free-hand sketching than in MacDraw. The row of Table 1 (Syntactic Density) shows the number of densely ordered drawings in free-hand sketching versus MacDraw per session. The second row of Table 1 (Semantic Density) indicates that the number of densely ordered ideas per session are also much greater in free-hand sketching than in MacDraw.

Table 1: Mean Numbers of Densely Ordered Episodes and Reinterpreted Episodes per Session.

	Free-hand	MacDraw
Syntactic Density	11.2 (S=5.1)	3.0 (S=3.4)
Semantic Density	10.4 (S=1.3)	4.1 (S=3.9)
Reinterpretations	2.4 (S=3.4)	0.67 (S=0.87)

There were also a significantly greater number of reinterpretations in free-hand sketching than in MacDraw (see Table 1, row 3). Thus as predicted, the free-hand sketches displayed greater ambiguity and/or lack of syntactic disjointness.

On the basis of these results, and converging verbal evidence, it is concluded that the two symbol systems are indeed being used in the way predicted. That is, the free-hand sketches belong to a substantially ill-structured symbol system while MacDraw drawings belong to a substantially well-structured system. Finally, we want to know whether this has the predicted impact on the number and types of transformations.

Table 2: Mean Numbers of Lateral Transformations per Session

	Free-hand	Macdraw
Syntactic Lateral Transformations	8.9 (S=4.4)	3.2 (S=3.2)
Semantic Lateral Transformations	8.0 (S=3.3)	3.9 (S=3.4)

[10] Notice here the collapse of the logically distinct notions of unambiguity and disjointness. It was not possible to distinguish between them with the given methodology.

[11] Each reported result is statistically significant (paired, one-tail t test, $P<0.05$).

It turns out that there is a statistically significant difference in the number and types of transformations (see Table 2). As predicted we get significantly more lateral transformations, at both the syntactic and semantic levels, with the ill-structured representations (free-hand sketching) than with the well-structured representations (MacDraw).

It is also worth noting that in both the density measurements and the lateral transformation measurements the difference between the ill-structured and well-structured representations is greater at the syntactic level than the semantic level. This may be due to the fact that less of the work is being off-loaded onto the external representation in the case where the external representations are well-structured.

On the basis of these results, the failure of alternative interpretations (Goel, 1991; Goel, 1992b), and the assumption that lateral transformations are desirable,[12] I conclude that, at least some ill-structured problems -- like design -- require (or at least benefit from) ill-structured diagrammatic representations at certain points.

4.0 Conclusion

A distinction has been made between ill-structured and well-structured representations to parallel the distinction between ill-structured and well-structured problems. It has been suggested that there may be a need for ill-structured problems to be accompanied by ill-structured representations. It has been shown that, at least in the case of preliminary design problem solving, ill-structured diagrams play an important cognitive role. When they are replaced by well-structured diagrams some predictable breakdowns occur in the number and types of state transformations that subjects engage.

Acknowledgements:

This paper is based on work completed for the author's Ph.D. dissertation (UC-Berkeley, 1991). The author is indebted to Peter Pirolli, Mimi Recker, and Susan Monaghan for helpful discussions and comments. This work has been supported by a Gale Fellowship, a Canada Mortgage and Housing Corporation Fellowship, a research internship at System Sciences Lab at Xerox PARC and CSLI at Stanford University, and an Office of Naval Research Cognitive Science Program grant (# N00014-88-K-0233 to Peter Pirolli).

References:

Elgin, C. Z. (1983). *With Reference to Reference*. Indianapolis, IN: Hackett Publishing.

Goel, V. (1991) *Sketches of Thought: A Study of the Role of Sketching in Design Problem Solving and its Implications for the Computational Theory of Mind*. Ph.D. Dissertation, University of California, Berkeley.

Goel, V. (1992a). The Cognitive Role of Sketching in Problem Solving. Manuscript submitted for publication. .

Goel, V. (1992b). Specifying and Classifying Representational Systems: A Critique and Proposal for Cognitive Science. In *Cognition and Representation*, . SUNY, Buffalo:

Goel, V., & Pirolli, P. (in press). The Structure of Design Problem Spaces. *Cognitive Science*.

Goodman, N. (1976). *Languages of Art: An Approach to a Theory of Symbols (second edition)*. Indianapolis, IN: Hackett Publishing.

Kosslyn, S. M., & Pomerantz, J. R. (1981). Imagery, Propositions, and the Form of Internal Representations. In N. Block (Eds.), *Readings in Philosophy of Psychology, Vol. 2* Cambridge, Mass.: Harvard University Press.

Larkin, J. H., & Simon, H. A. (1987). Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science, 11*, 65-99.

Newell, A., & Simon, H. A. (1972). *Human Problem Solving*. Englewood Cliffs, N.J.: Prentice-Hall.

Paivio, A. (1977). Images, Propositions, and Knowledge. In J. M. Nicholas (Eds.), *Images, Perception, and Knowledge* Dordrecht-Holland: D. Reidel Publishing Co.

Pylyshyn, Z. W. (1981). The Imagery Debate: Analogue Media Versus Tacit Knowledge. *Psychological Review, 88*(1), 16-45.

Rey, G. (1981). Introduction: What are Mental Images? In N. Block (Eds.), *Readings in Philosophy of Psychology, Vol. 2* Cambridge, Mass.: Harvard University Press.

Rich, E. (1983). *Artificial Intelligence*. N.Y.: McGraw-Hill Book Company.

Simon, H. A. (1972). What is Visual Imagery?: An Information Processing Interpretation. In L. W. Gregg (Eds.), *Cognition in Learning and Memory* --: John Wiley & Sons, Inc.

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

[12]This, I take it is an unproblematic assumption. It amounts to little more than the claim that better solutions will result if one is allowed to explore the space of solutions and to customize any preexisting solutions to the present context.