

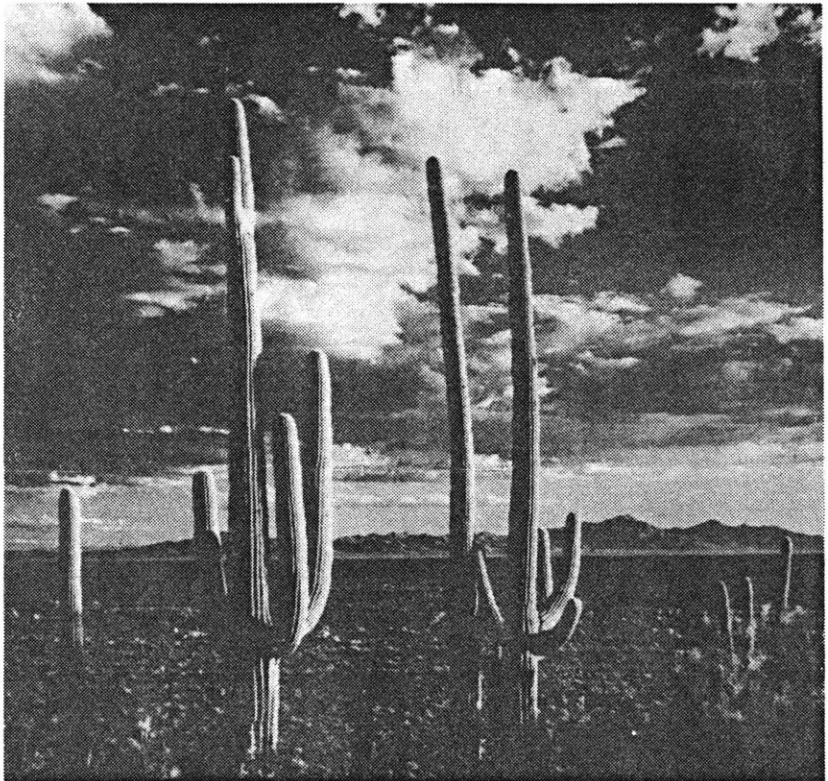
THE STRUCTURE OF CONSERVATION:

Experiments in Representing Design Knowledge
for Arid Lands Design

by

Andrew M. Bennett

Bachelor of Human Ecology
College of the Atlantic
Bar Harbor, Maine
1982



"You do not beg the sun for mercy.

*---'Muad'Dib's Travail'
from the Stilgar Commentary"
[i]*

SUBMITTED TO THE DEPARTMENT OF ARCHITECTURE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE
DEGREE
MASTER OF ARCHITECTURE AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 1987

© Andrew M. Bennett 1987

The Author hereby grants to M.I.T.
permission to reproduce and to distribute publicly copies
of this thesis document in whole or in part

Signature of the author _____
Andrew M. Bennett
Department of Architecture
May 8, 1987

Certified by _____
Frank Miller
Assistant Professor, Computer Aided Design
Thesis Supervisor

Accepted by _____
Julian Beinart
Chairman
Departmental Committee for Graduate Students

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
JUN 03 1987
LIBRARIES



THE STRUCTURE OF CONSERVATION: Experiments In Representing Design Knowledge for Arid Lands Design

by
Andrew Michael Bennett

Submitted to the Department of Architecture on May 8, 1987
in partial fulfillment of the requirements for the Degree of
Master of Architecture

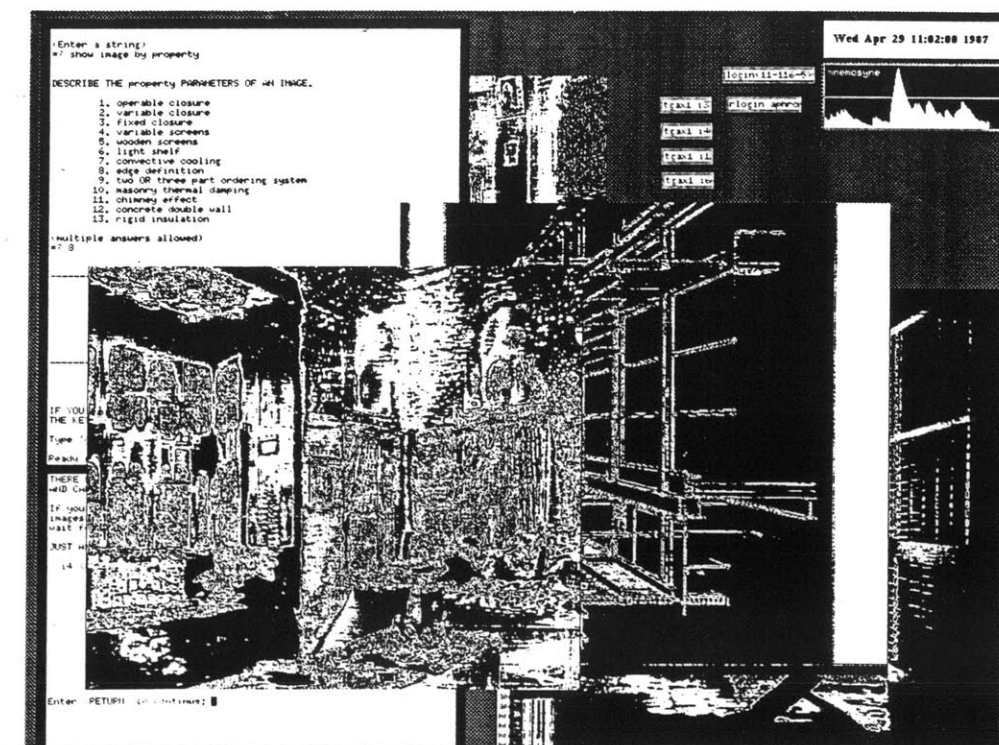
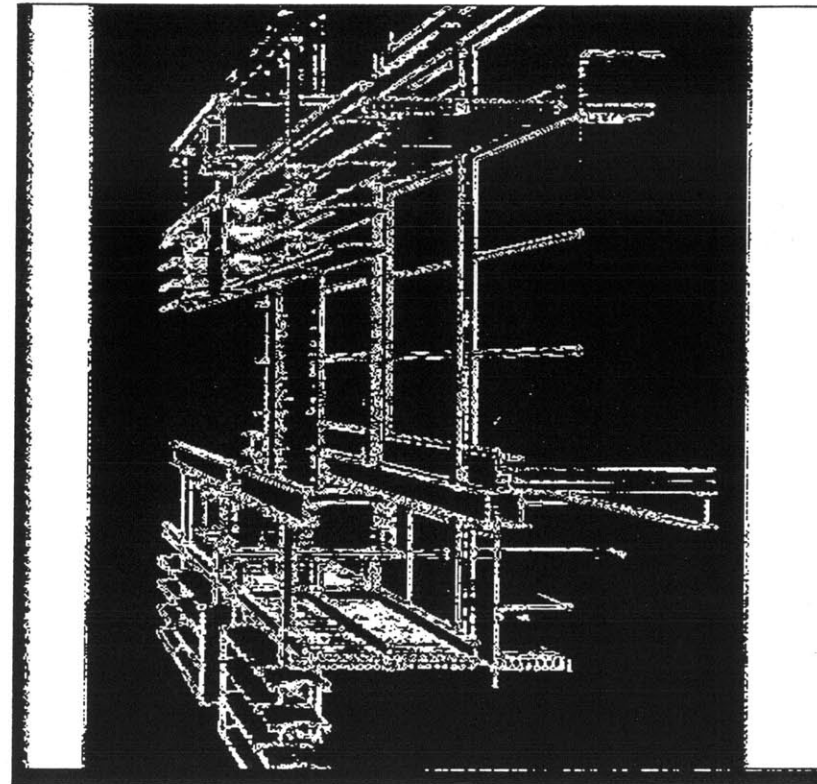
ABSTRACT

This thesis proposes, through a multi-layered exploration, the development of a system of computer tools for architects. The research consists of a series of "design sessions" in the context of a desert design problem. The goal is to create a knowledge-based system using a commercially available expert shell, which provides the designer with an automated interface to visual references.

Data can be seen as a collection of things, while knowledge can be similarly seen as a collection of relationships between things. An expert shell is literally a program that is "empty" of knowledge, and into which a designer puts knowledge: a knowledge-base is the result. The shell itself acts as a means of manipulating that knowledge-base by an inference process that is activated by rules, or hypotheses and tests. The experimental framework of the thesis is devised to evaluate both type of inference processes in relation to their capabilities for representing design knowledge.

The design problem serves to outline a methodology for understanding the process of design, but it also is the means by which a design grammar and syntax appropriate to the automated system are formally described. The intent is not to compile a vast domain of knowledge on all issues of arid lands design, but to focus on a specific architectural response to the climate: the relationship between the primary structural system and the secondary closure system. The design of a window system is the vehicle for documenting observations of the way visual references are used. From this process a descriptive system and body of "expert" rules are developed to define the function of the automated environment. The larger goal is to then relate the syntactical environment to a general image referencing system so that the expert system can act as a personal design consultant. The image referencing system is a distinct and important component of the automated environment, and as such a detailed specification of its nature and operation is intended to show the interdependence of the knowledge-base and a visual database.

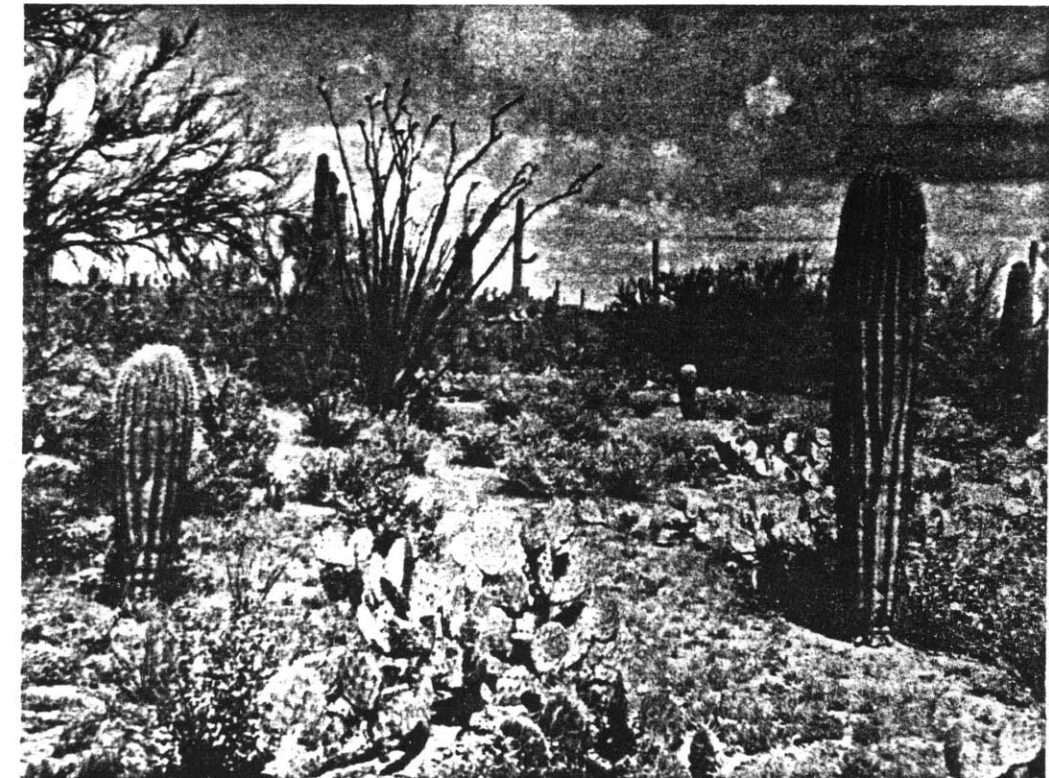
Thesis Supervisor: Frank Miller
Title: Assistant Professor, Computer Aided Design



PREFACE

For a great part of my life the desert has been the haven of many of my aspirations, both voiced and unvoiced. This thesis, although not solely a design problem, was an opportunity for me to synthesize a few disconnected interests, all of which are directed towards the practice I see myself constructing. I wrote a statement for admissions that voiced an opinion that is still active; *"I believe all architectural design must incorporate Nature as an active form-giver to the design."* The design component of the thesis stems from that sense of values. However, I am also exploring the landscape of a changing profession, a profession that is grappling with the transition to a more design specific and information hungry society. Computers are now becoming an accepted part of the profession and the need for people to understand the technology has increased considerably. Although the role of machines in the production environment is well established, the issues surrounding the use of computing tools in preliminary design processes are numerous, highly debatable, and still the frontier of applied research.

I never foresaw that I would participate in the world of computers as much as I have, but the knowledge and exposure that I have gained have set my doubts into perspective. I see the virtues of what the new technologies of knowledge engineering, optical memory storage, and image generation and processing offer for changing architectural education and practice. The direction is potentially towards new and creative horizons. The research I have been doing, and this thesis, are attempts at raising the conceptual issues and testing prototypes towards more effective use of tools, and thereby increasing the chances for innovation.



*"From the architectural world of pure harmonies one
should be able to experience Nature in all its phases."*

S. RASMUSSEN
Experiencing Architecture
[ii]

TABLE OF CONTENTS

ABSTRACT	2
PREFACE	3
TABLE OF CONTENTS	4
KNOWING IN ACTION	5
VISUAL CUES	22
DESIGN INFERENCE	32
REFERENCES	
Chapter 1	21
Chapter 2	31
Chapter 3	43
APPENDIX	44
POSTSCRIPT and ACKNOWLEDGEMENTS	53
BIBLIOGRAPHY	54

CHAPTER 1

KNOWING IN ACTION

The acts of the mind, wherein it exerts its power over simple ideas are chiefly these three: 1. Combining several simple ideas into one compound one, and thus all complex ideas are made. 2. The second is bringing two ideas, whether simple or complex, together, and setting them by one another so as to take a view of them at once, without uniting them into one, by which it gets all its ideas of relations. 3. The third separating them from all other ideas that accompany them in their real existence: this is called abstraction, and thus all its general ideas are made.

— JOHN LOCKE
An Essay Concerning Human Understanding
1690
[1]

Manifestation explained :
'primary system' in 'Behavior'
focus of problem :
generator :
competing possibilities :
i4
i8

Manifestation explained :
'secondary system' in 'Behavior'
focus of problem :
generator :
competing possibilities :
i4

Manifestation explained :
'light modulation' in 'Behavior'
focus of problem :
generator :
competing possibilities :
i2
i3
i5
i6
i7
i1
competing possibilities :
i4
generator :
competing possibilities :
i8
competing possibilities :
i2
i3
i5
i6
i7

DESCRIBE THE scope PARAMETERS OF AN IMAGE.

If you need explanations for the values of Scope, then type explain and the number of the value.

1. primary secondary
2. env_controls
3. formal organization
4. Exit

(multiple answers allowed)

=? explain 2

The term env_controls refers to that group of elements that might be defined as functions or components dealing with issues related to climate modulation, whether it be wind, moisture, temperature, or light factors.

The reason environmental controls is included in the Scope is that secondary systems are generally designed to satisfy the requirements of closure from the elements. If an attitude of design with Nature is accepted, then the more a designer is informed of means to deal with sun, wind, and moisture, the more freedom they are given to design variations.

If lower sill = 2' - 3' above floor and opening = singular
then overhang length equals height of opening.

If design idea = three part ordering scheme
then lower sill height = 2' - 3'.

mid-sill height = 4.5' - 5.5'.

overhang length = upper window opening height.

property = 2 and 3 part ordering.

property = edge definition.

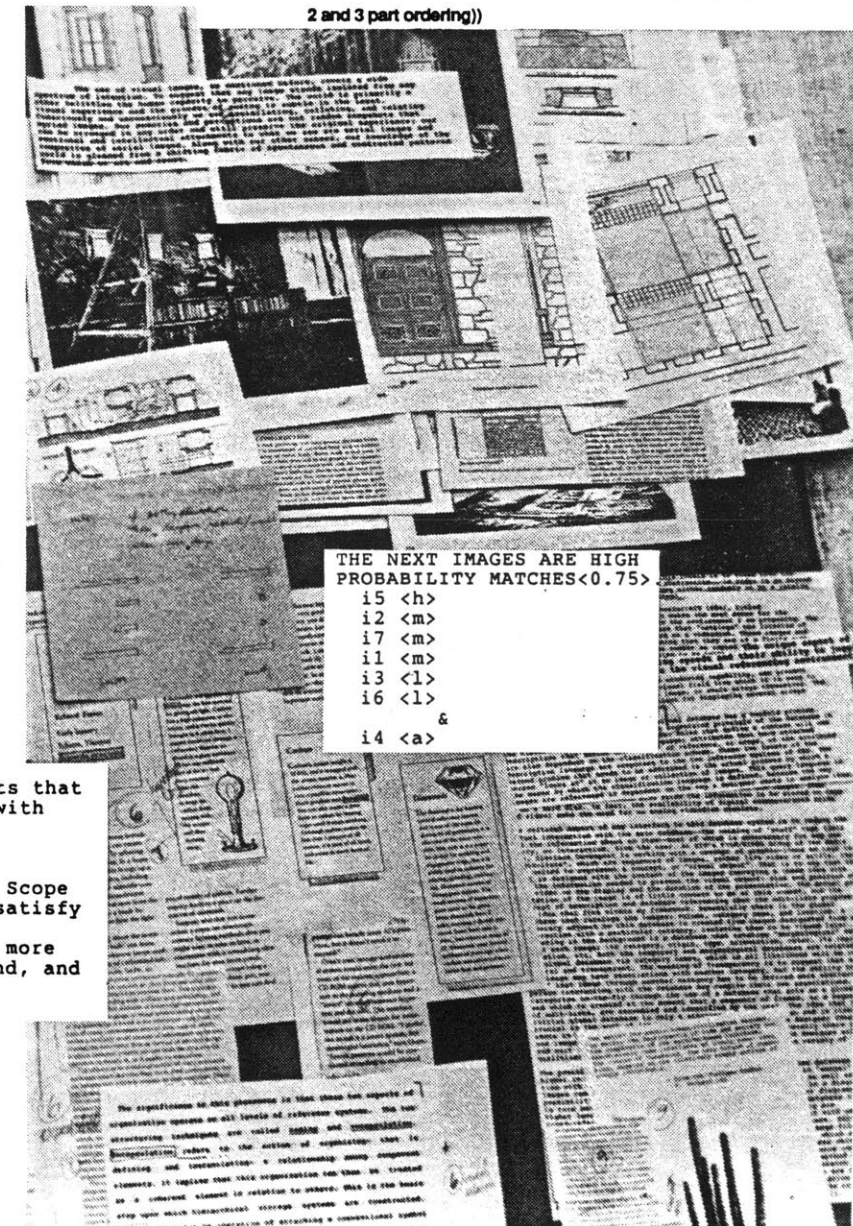
If opening = south & full height

then overhang length = window height | half of window height with use of vegetative screen.

scope = primary secondary & env_controls.

summary description:

(scope: primary secondary & env_controls (property: edge definition & 2 and 3 part ordering))



An explicit goal of the architectural design process is the resolution of a problem, or a set of problems, through visual dialog. Design is a purposeful behavior requiring specialized knowledge and processes, which by their nature are often a 'reflective conversation with a unique and uncertain situation'. [2] Uncertainty is the result of searching for possible solutions which can only be evaluated after generating them. Design is essentially an iterative, educated trial-and-error experience. The uniqueness of the process is that the designer's inferences are dependent on the internal constraints and the context of the problem as well as external information specific to the designer's experience, intentions, and expression of cultural associations and influences.

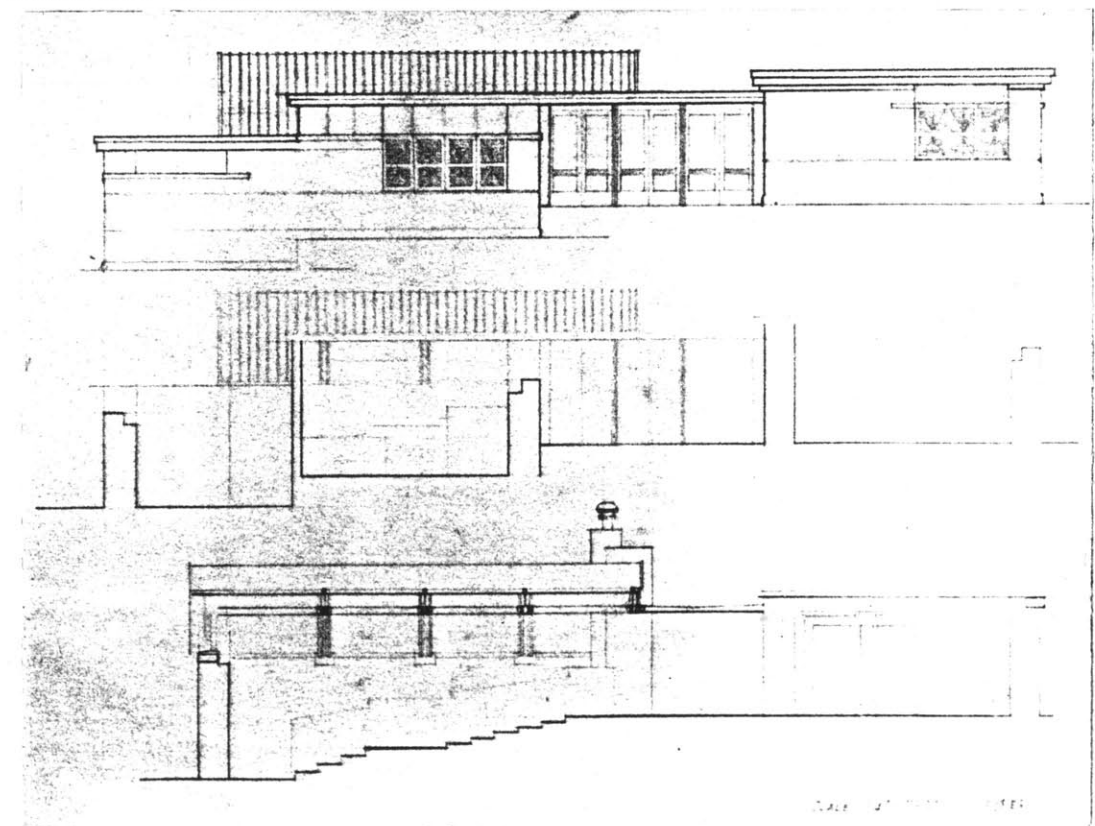
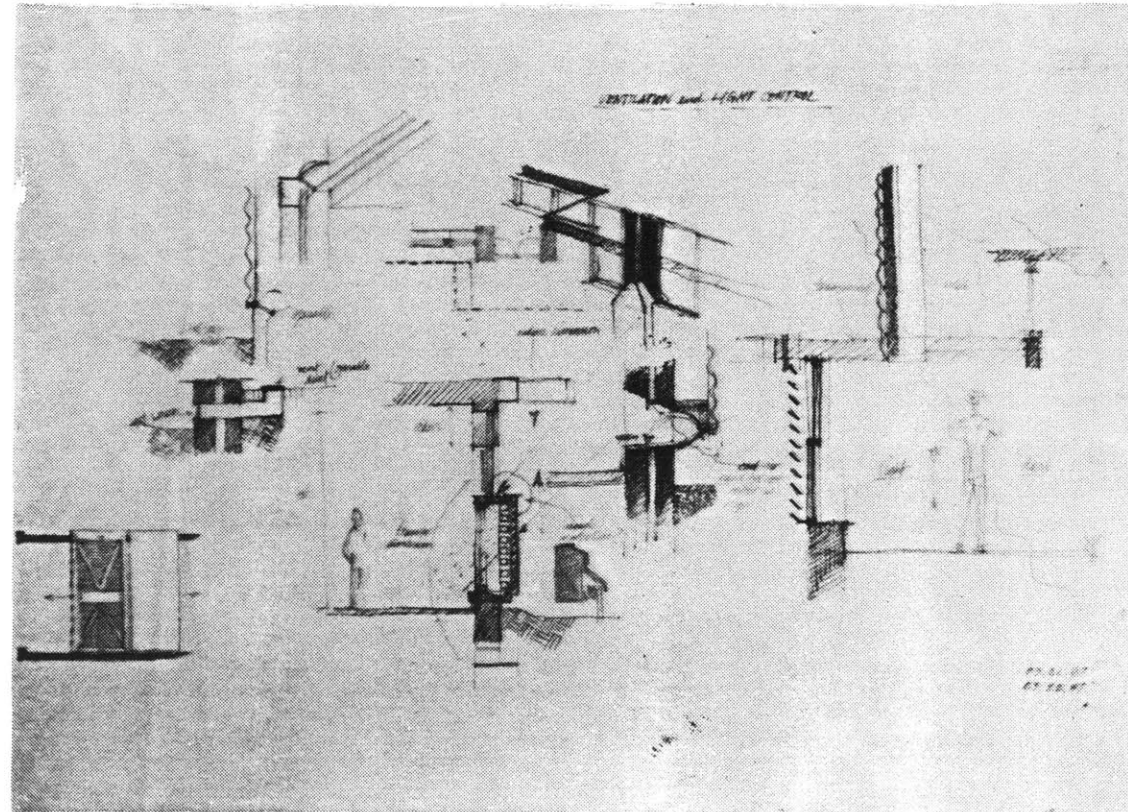
For anyone unfamiliar with design in general, and architecture specifically, the world of visual thinking seems as inexplicable as Margaret Boden suggests in her book, ARTIFICIAL INTELLIGENCE AND NATURAL MAN:

"The everyday assumption of a sharp dichotomy between seeing and thinking implicitly denies any possible contribution of the latter to the former. Nor is it clear just what one is required to think in any particular case, since usually one is introspectively unaware of the underlying inferential processes that are essential to vision, and unable consciously to call them into play." [3]

However, Donald Schon, in THE REFLECTIVE PRACTITIONER, graciously gives credit to our intuitions where reflection may fail:

"When we go about the spontaneous, intuitive performance of the actions of everyday life, we show ourselves to be knowledgeable in a special way. Often we cannot say what it is we know. When we try to describe it we find ourselves at a loss, or we produce descriptions that are obviously inappropriate. Our knowing is ordinarily tacit, implicit in our patterns of action and in our feel for the stuff with which we are dealing. It seems right to say that our knowing is in our action." [4]

Nevertheless, we shall see that in order for a successful conclusion to occur within a reasonable time, the visual inferences an architect relies upon must be as conscious as possible. The question becomes, **"How is something designed? What are the influences and the nature of the process?"**.



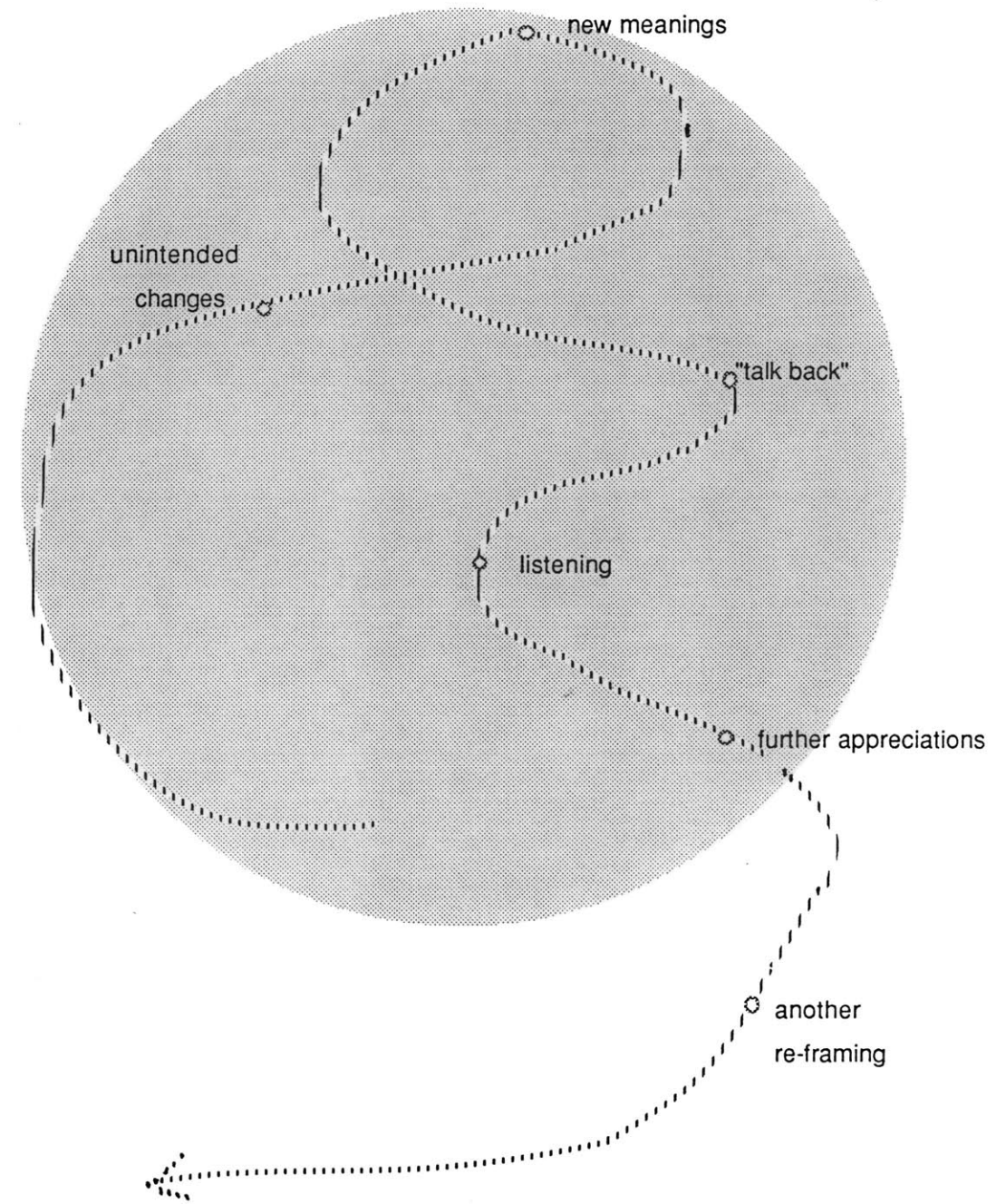


fig. 1

The diagram overlaying this page represents how the design process may be interpreted within a domain of purposeful activities. The diagram is also intended to model the nature of this paper. Architectural design is not a linear methodology, and although published text is inherently a linear representation of a language, I intend to portray the dialogs of the thesis, the design problem and the knowledge-base system in a manner more accurate to their visual nature. After all, the thesis is about referencing images and the relations that may be derived through them. An analogy I seek to reinforce is that the collage of references on my display board is not only an effective model of "process", but is also a model of the ideal display required for the machine interface. Finally, the diagram acts as an informal reference point to the organization and grammar of what I wish to present.

Several categories of information define the global frame --- the 'situation' --- of interest of the thesis:

- 1) The Desert
constraints; lessons from Nature; architectural rules-of-thumb
- 2) The Design context
constraints; location; intentions
- 3) The Image references
What are they? Why? How are they used?
- 4) The Design problem
What of? Why? knowledge acquisition process
- 5) The Computer
context; ideal and real specification; Why?
- 6) The System components
visual database; expert shell; optical storage; graphic imaging
- 7) The Visual sub-system
details and interface
- 8) The Knowledge sub-system
details(code) and interface
- 9) The Evaluation
What's appropriate?

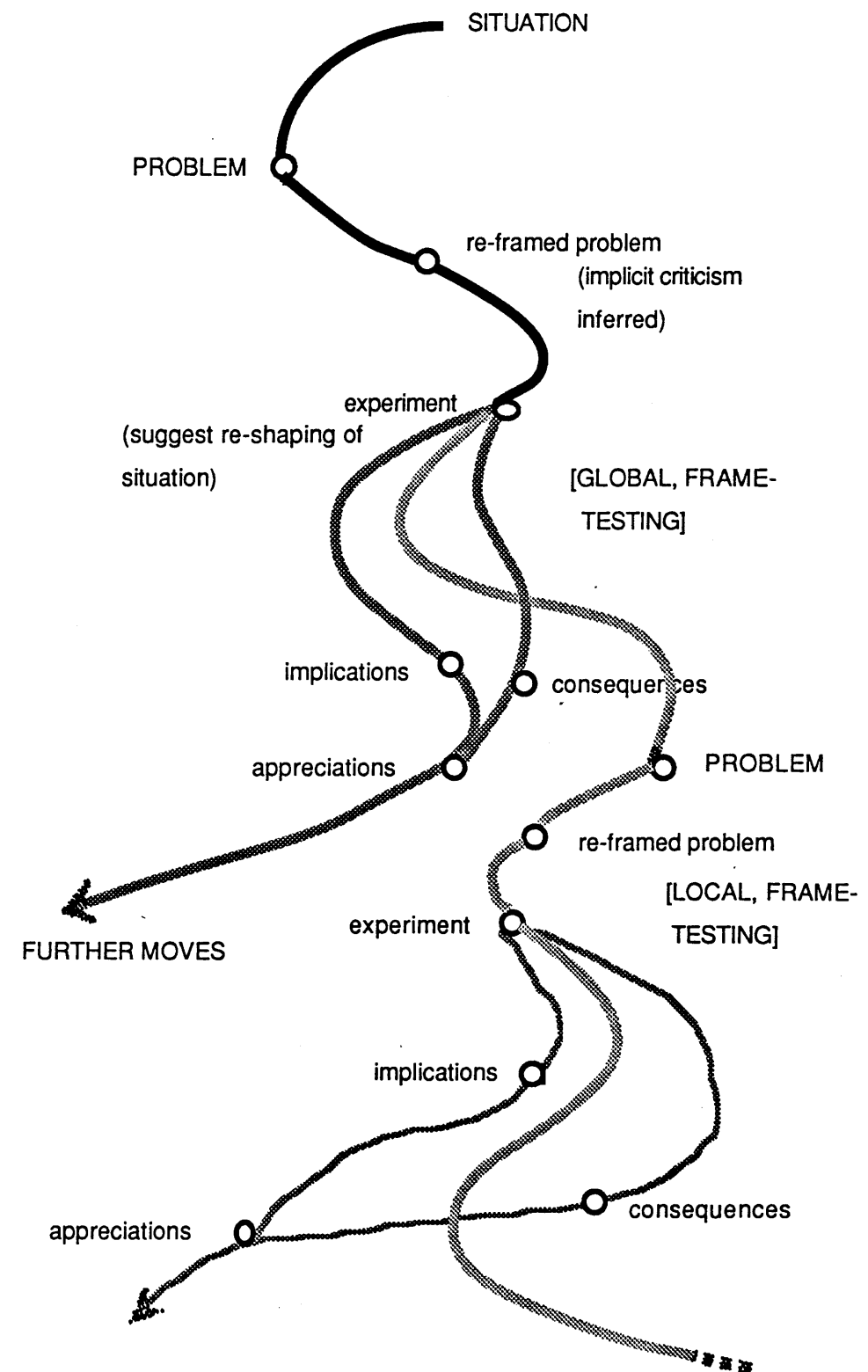
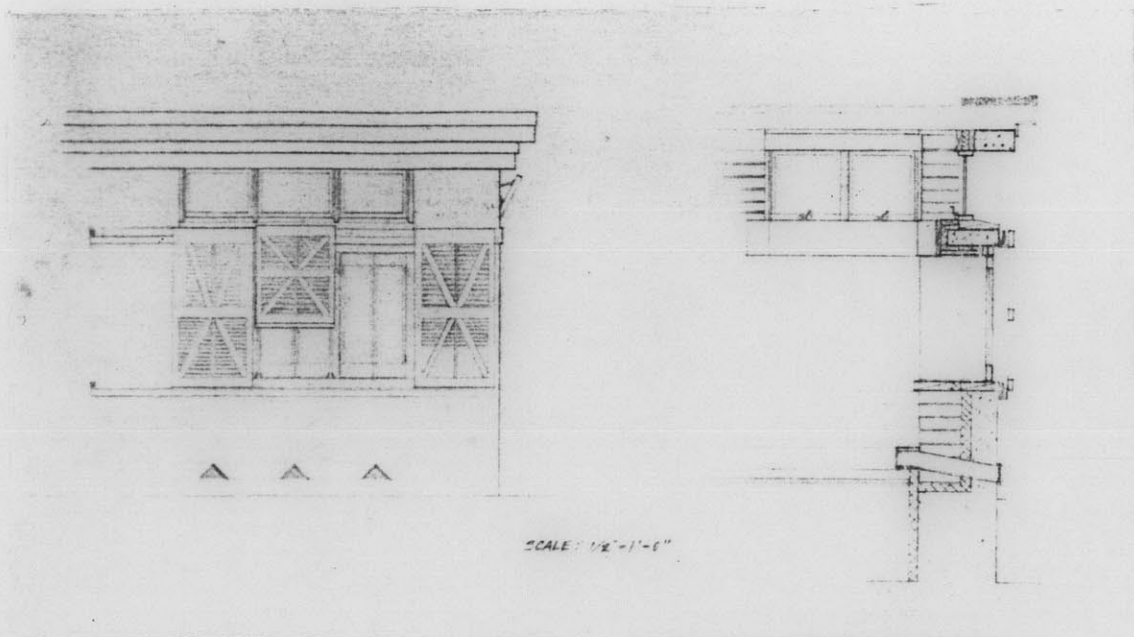
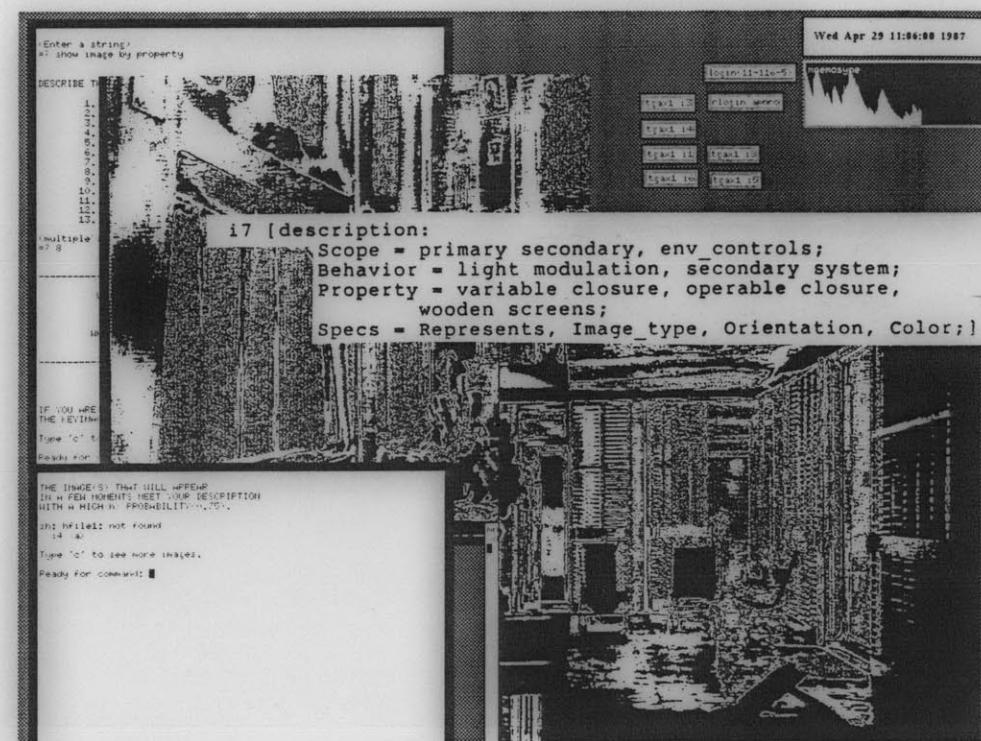
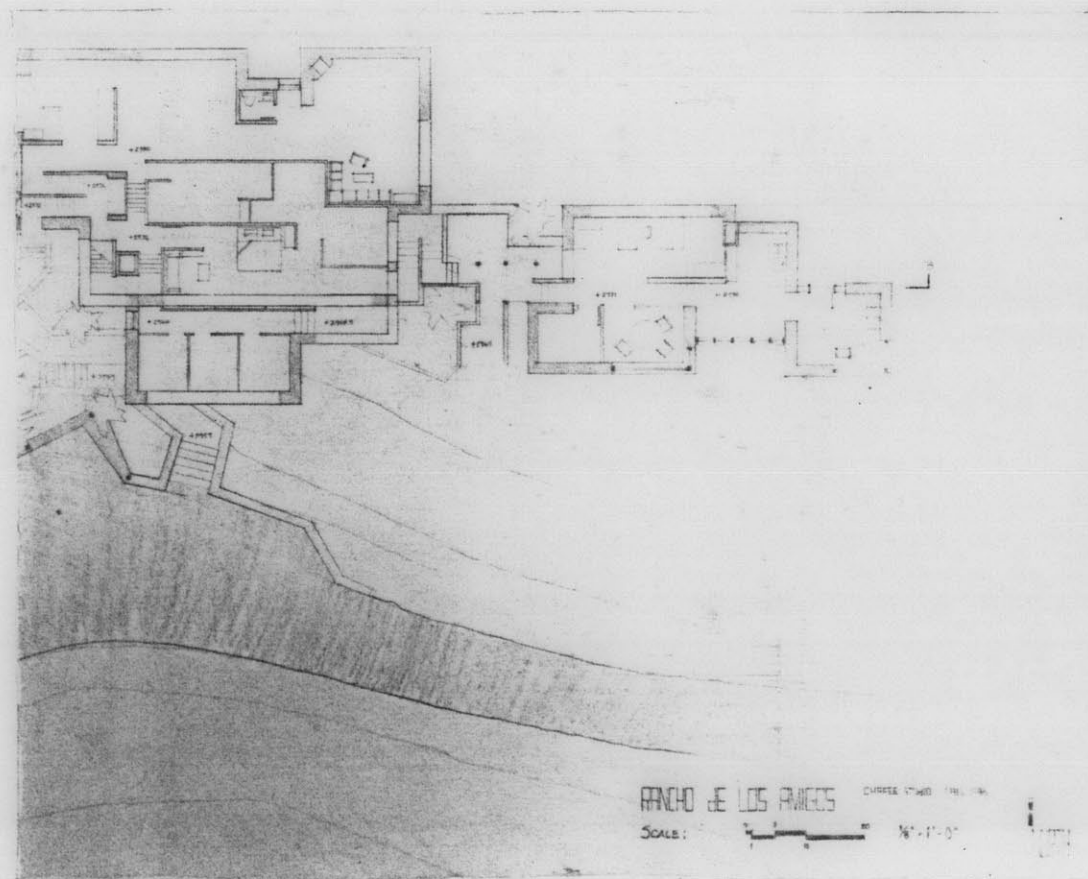


fig. 2



LOCAL FRAME
The Design



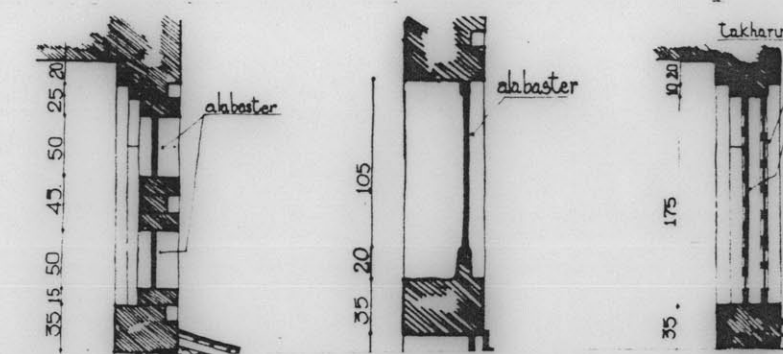
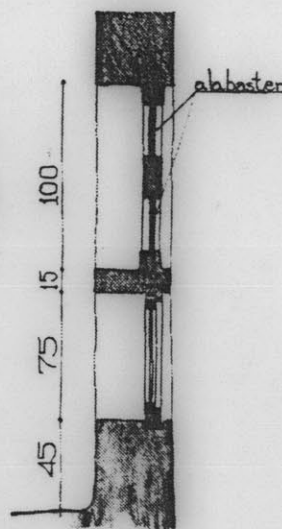
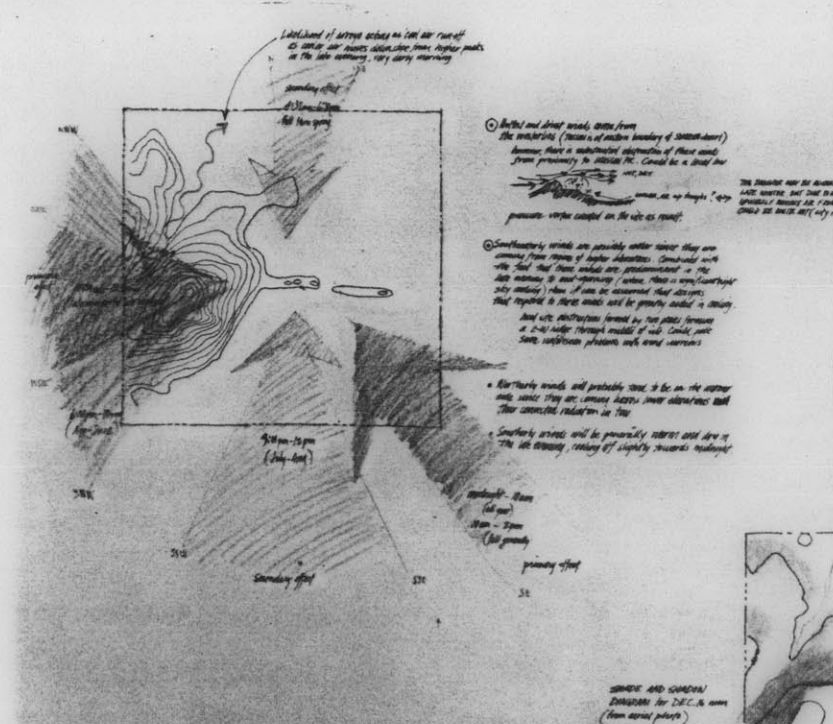
LOCAL FRAME
The Automated Environment

```
\04.23.87
\A.Bennett
\Thesis Experiments: Query Interface(q_new3)
\Kes production-rule system(KES.ps)
```

```
\*****
text:
```

```
{banner:"      ---- ARID LANDS DESIGN ADVISOR ----      ",
  "THIS MODULE IS SPECIFICALLY FOR SECONDARY SYSTEM DESIGN",
  "[MASSACHUSETTS INSTITUTE OF TECHNOLOGY; Department of ",
  "Architecture and Urban Planning Computer Resources Lab.]",
  "Author:A.BENNETT",
  "THIS IS A KNOWLEDGE-BASED VISUAL REFERENCING SYSTEM",
  "WHICH IS USED AS A DESIGN TOOL FOR ARCHITECTURAL",
  "APPLICATIONS SPECIFIC TO DESIGN IN ARID LANDS.",
  "THE SYSTEM IS COMPOSED OF TWO PARTS:",
  "  1)AN INTERFACE SYSTEM FOR CHANNELING DESIGN QUERIES",
  "  TO A VISUAL KNOWLEDGE-BASE and",
  "  2)THE VISUAL KNOWLEDGE-BASE WHICH IS WHERE THE",
  "    INTERACTIVE DISPLAY OF SELECTED IMAGES AND",
  "    KNOWLEDGE SOURCES IS DEMONSTRATED.",
  "PART ONE OF THE REFERENCING SYSTEM ALSO MANAGES MORE",
  "IN-DEPTH DATABASE APPLICATIONS THAT ARE TIED DIRECTLY",
  "TO THE 'LIBRARY' OF AVAILABLE IMAGES. IMAGE PROCESSING",
  "AND MANAGEMENT OF VISUAL RESOURCES ARE TASKS AVAILABLE",
  "THROUGH THIS OTHER CHANNEL OF THE QUERY INTERFACE."}
```

fig. 3



DESCRIBE THE property PARAMETERS OF AN IMAGE.

1. operable closure
2. variable closure
3. fixed closure
4. variable screens
5. wooden screens
6. light shelf
7. convective cooling
8. edge definition
9. two OR three part ordering system
10. masonry thermal damping
11. chimney effect
12. concrete double wall
13. rigid insulation
14. NONE

Type in your query using the following examples as a model of sentence building.

Show image by Scope , or Show image by combo

(multiple answers allowed)
=? 10

options: verbs objects modifiers

SHOW

IMAGE
ARRAY
REFERENCE

(by) SCOPE
BEHAVIOR
PROPERTY
SPECIFICATION

Manifestation explained :
'masonry thermal damping' in 'Property'

THE IMAGES ARE LOCATED

CHANGE or
GIVE

INFORMATION (on) IMAGE

CATEGORY(ingres)
GLOSSARY(ingres)

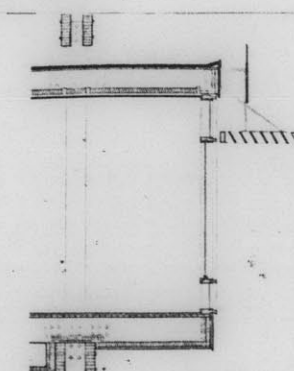
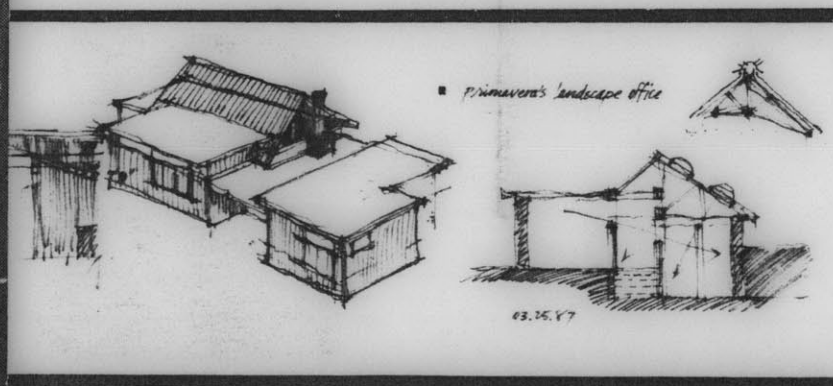
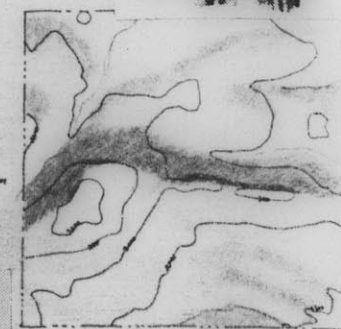
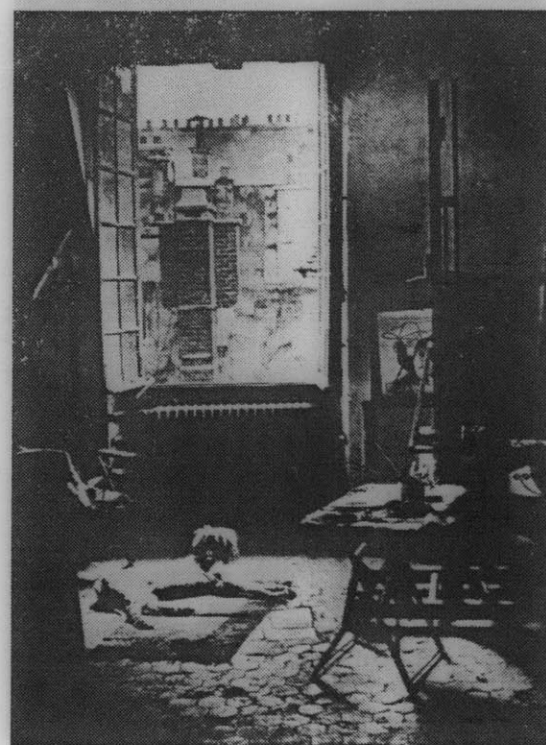
'e not satisfactory, wait

JUST ASK TO SEE OTHER IMAGES.

These are the images that match your description. Those followed by the <a> value will be displayed first, then in descending order thereafter.

i5 <h>
i2 <m>
i7 <m>
i1 <m>
i3 <l>
i6 <l>
&
i4 <a>

236 WINDOWS WHICH
OPEN WIDE*

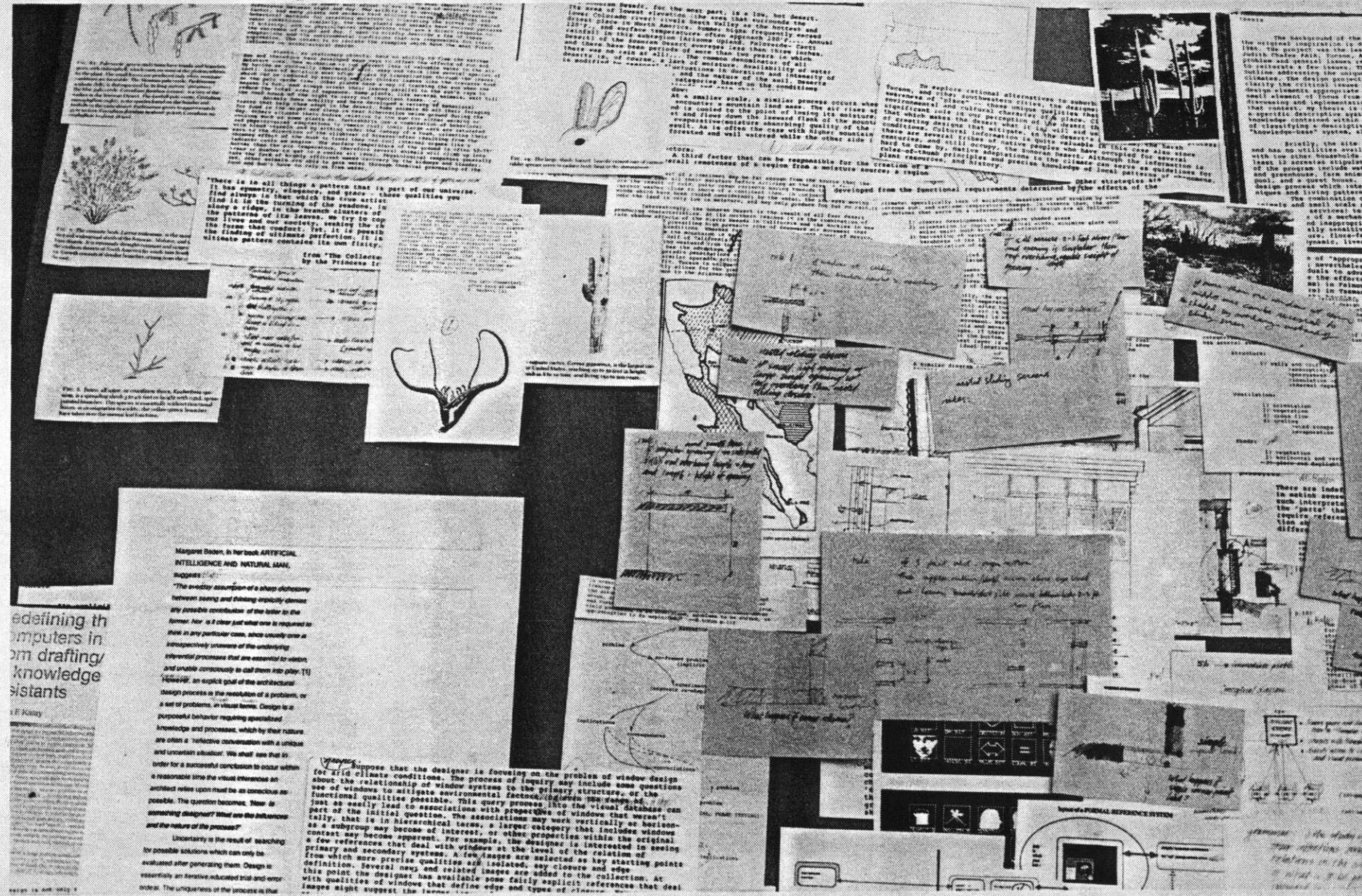


cd thesis/exp
rm desc resp reqst
xterm -fn 6x10 -bw 5 =80x30+1+1 -rv -e kr q_new2 &

fig. 4

THE HOMOSOTE MONITOR

"Global Frame"



```
\*****
externals:
```

```
    image_a:
        [program: "afile1"].
    image_h:
        [program: "hfile1"].
    image_m:
        [program: "mfile1"].
    image_l:
        [program: "lfile1"].

    setmode:
        [program: "chmode"].

    clean:
        [program: "clean_um"].

    parse:
        [program: "htout.parse"]
```

■ ELABORATION: Determining and assigning subframes to show more specific details.

■ ALTERATION: Finding a frame to replace one that

```
doesn't work for the conditions. while image_mnu # return do
    if image_mnu = no good
    then message "", "THE IMAGE(S) THAT WILL APPEAR",
        "IN A FEW MOMENTS MEET YOUR DESCRIPTION"
        "WITH A HIGH<h> PROBABILITY(0.75).", ""
    run image_h.
    display value of keyimage.
    message "Type 'c' to see more images.".
    break.
    message "", "THE IMAGE(S) THAT WILL APPEAR",
        "IN A FEW MOMENTS MEET YOUR DESCRIPTION"
        "WITH A MEDIUM<m> PROBABILITY(0.50).", ""
    run image_m.
    display value of keyimage.
    run image_l.
endif.
```

■ NOVELTY: Determining an alternate move if no acceptable frame is located. Locate a new frame or alter an old one?

fig. 5

When working within a bounded domain[a frame]...

```
%
\*****
actions:

    trace.

    read "com.bo", request.
    while request # combo do
        if request = scope
            then sc = true.
            Behavior = NONE.
            Property = NONE.
            Specs = NONE.
            name = NONE.
        else

while response # "Exit" do
    while request # combo and
        descriptor # EXIT do

        write "com.bo", request.

        if fire_file = one
            then write "resp", Scope.
        endif.
```

■ EXPECTATION: Determining an initial frame to meet a set of given conditions.

[program: "start_ingres"].

■ LEARNING: Determining what frames should be saved or appended as a result of the interaction.

fig. 6

This may seem like a table of contents or an outline, but it is only a convenient diagram of the parallel and layered sources of information, or directions of search that I perform in defining a sense of design knowledge. The setting of image referencing, which is a domain of the design problem, can be called a local frame. The global frame includes the range of all possible solutions, generally derived from testing of local frames. The process of problem solving is to try and isolate alternative solutions to advance from one local test state to another. The final solution can be described as the culmination of several transitions between local frames where certain goals have been satisfied and re-framed to define a new frame, or state that is the most current. [5] Refer to the diagram of the design process (Fig. 2) to find the equivalent in the phase that begins with a "re-framing of the problem" and passes through "experimentation" to "further moves". On paper this theory seems quite dry and uni-dimensional, where in actuality a more accurate description would acknowledge that global and local frame-testing can, and often do, occur simultaneously. The heuristics, or the knowledge that guides the search, relies on information generated internally from the problem as well as from external sources. Knowledge acquisition during this process can be happen analytically, rationally, randomly, or intuitively.

It's important to clarify that reasoned inference is only part of the design process. The architect has only a fragment of all the constraints to the design available from the start. One of the difficulties that the architect must face is devising some means by which new frames can be developed from the current state. If no additional constraints can be derived from those which are known, then the architect must seek information from outside of the problem to help determine new connections or constraints.

-- Those which can be expressed are not universally valid. Neither are they universally recognized. They may be valid for only a small domain of design, and believed to be false for other domains.

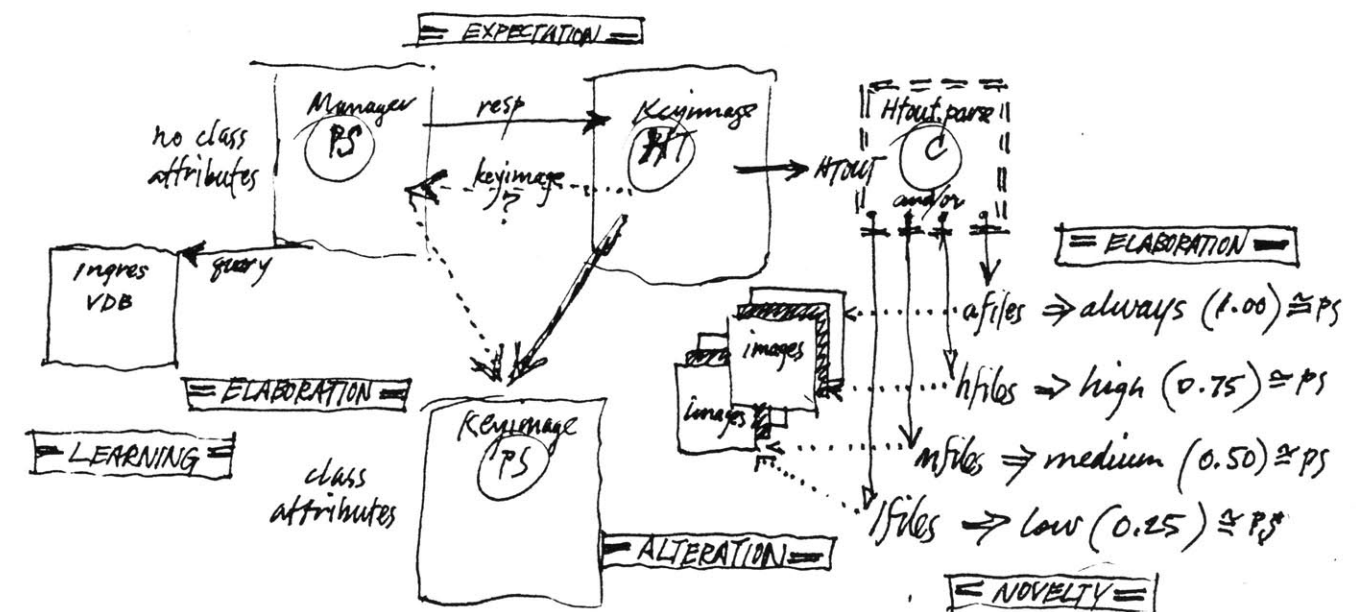
-- They are also not uniformly well articulated. Thus those which are known are expressed in terms which cannot be generalized from one feature to another.

```

if request = combo
then write "com.bo", request.
  message "", "",
  "-----",
  "",
  "          LINKING TO KEYIMAGE KNOWLEDGE-BASE...  ",
  "",
  "          *** PLEASE WAIT ***                      ",
  "",
  "          When the other window appears, move the  ",
  "          cursor there to answer any queries.      ",
  "-----",
  "", "",
run keyimage_kb.
erase descriptor.
descriptor = EXIT.
endif.

```

heuristic: Computer Sci. Relating to or using a problem-solving technique in which the most appropriate solution of several found by alternative methods is selected at successive stages of a program for use in the next step of the program. [*< Gk. heuriskein, to find. >*] [6]



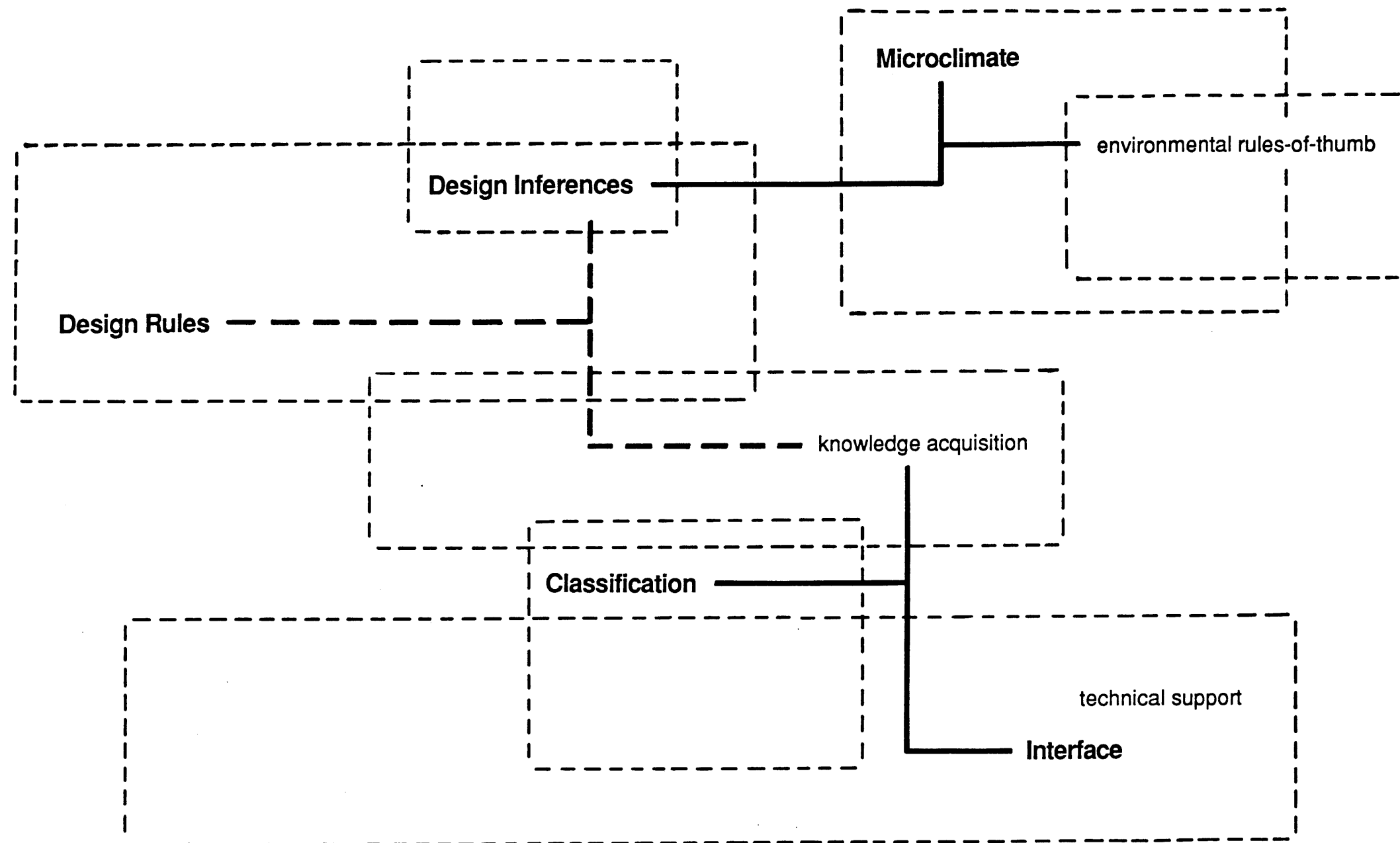


fig. 7 General Connections

**VARIABLE CLOSURE
CONVECTIVE COOLING**

if night sky cooling and variable closure
then three part organization and
ground level air inlet or
low sill with operable closure.
endif.

night sky cooling

timing of CO₂ uptake and photosynthesis

PRODUCTION RULES

*INFERENCE and
BACKWARD-CHAINING
obtain goal...*

BEHAVIOR = ventilation
PROPERTY = variable closure &
operable closure

show image by property

GENERAL CATEGORY: window system

They are not uniformly significant. The various properties are known in differing levels of detail. Each property is known differently depending on the context within which it is considered.

— The mechanism of inference is not uniformly directed. Thus the constraints which determine a property in one instance may be determined by it in another. " [7]

The combination of the internal searching and the external connection is perhaps the most challenging aspect of the process, and is certainly the most challenging to represent.

Suppose that a designer is focusing on the problem of window design for arid climate conditions. The process of inquiry may include some questions about the relationship of the primary structural system and a proposed secondary system of closure, or the use of windows to mitigate special environmental features, or even the range of functional types possible for windows. The options are virtually inexhaustible and include the specific and the general; a valid query may be at the level of research into unique qualities of the desert climate that may provide new associations, or an exploration that attempts to formalize notions about edge definition. By focusing one's attention to the use of visual references an initial query can easily lead to associations with properties that weren't framed in the original question. Simple associations may work vertically or horizontally in a hierarchy; a larger category that includes windows as a subgroup may be valid in the problem, or other equivalent properties in the original frame may be valid in relation to each other.

For example, I am interested in finding a few references that deal with windows in relation to the secondary system. An initial search is directed at collecting images of windows in general; this is a process by which a few images can be weeded out to serve as key starting points from which more precise properties will be isolated. I determine that operable closure and edge definition are properties of the secondary system, which can be described as a behavioral component of the primary and secondary system context. Several more images are added to the collection and a few are eliminated. Each level of description defines more explicitly the domain of each preceding interaction. The focus has narrowed to a set of images that are implicitly related by virtue of the

edge definition

behaviors

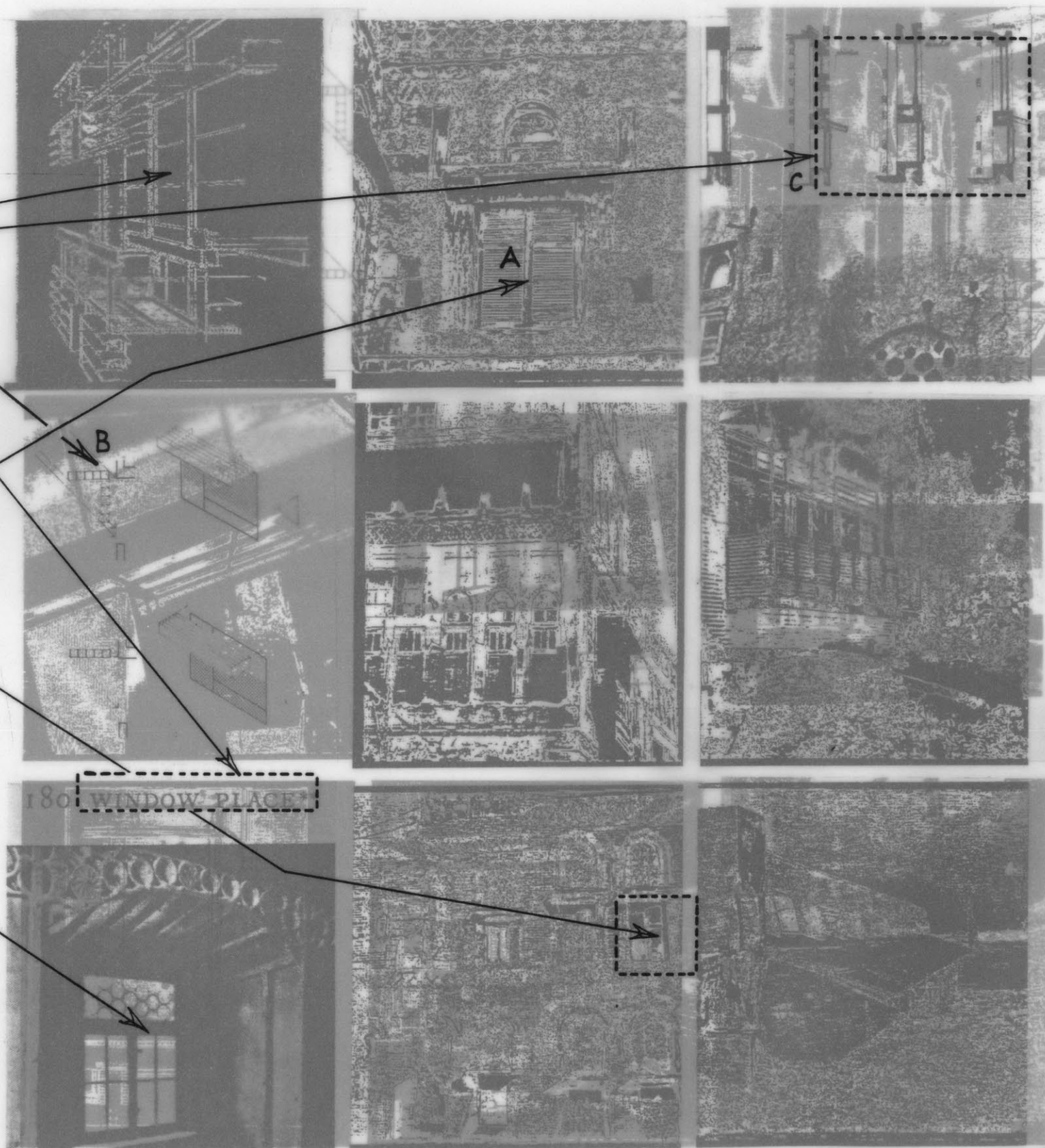
operable closure

A — wooden screens

B — light shelf

C — three part ordering

fig. 9
fig. 10



-- They are not uniformly significant. The various properties are known in differing levels of detail. Each property is known differently depending on the context within which it is considered.

-- The mechanism of inference is not uniformly directed. Thus the constraints which determine a property in one instance may be determined by it in another. " [7]

The combination of the internal searching and the external connecting is perhaps the most challenging aspect of the process, and is certainly the most challenging to represent.

Suppose that a designer is focusing on the problem of window design for arid climate conditions. The process of inquiry may include some questions about the relationship of the primary structural system and a proposed secondary system of closure, or the use of windows to mitigate special environmental features, or even the range of functional types possible for windows. The options are virtually inexhaustible and include the specific and the general; a valid query may be at the level of research into unique qualities of the desert climate that may provide new associations, or an exploration that attempts to formalize notions about edge definition. By focusing one's attention to the use of visual references an initial query can easily lead to associations with properties that weren't framed in the original question. Simple associations may work vertically or horizontally in a hierarchy; a larger category that includes windows as a subgroup may be valid in the problem, or other equivalent properties in the original frame may be valid in relation to each other.

For example, I am interested in seeing a few references that deal with windows in relation to the primary and secondary system. An initial search is directed at collecting images of windows in general; this is a process by which a few images can be weeded out to serve as key starting points from which more precise properties will be isolated. I determine that operable closure and edge definition are properties of the secondary system, which can be described as a behavioral component of the primary and secondary system context. Several more images are added to the collection and a few are eliminated. Each level of description defines more explicitly the domain of each preceding interaction. The focus has narrowed to a set of images that are implicitly related by virtue of the

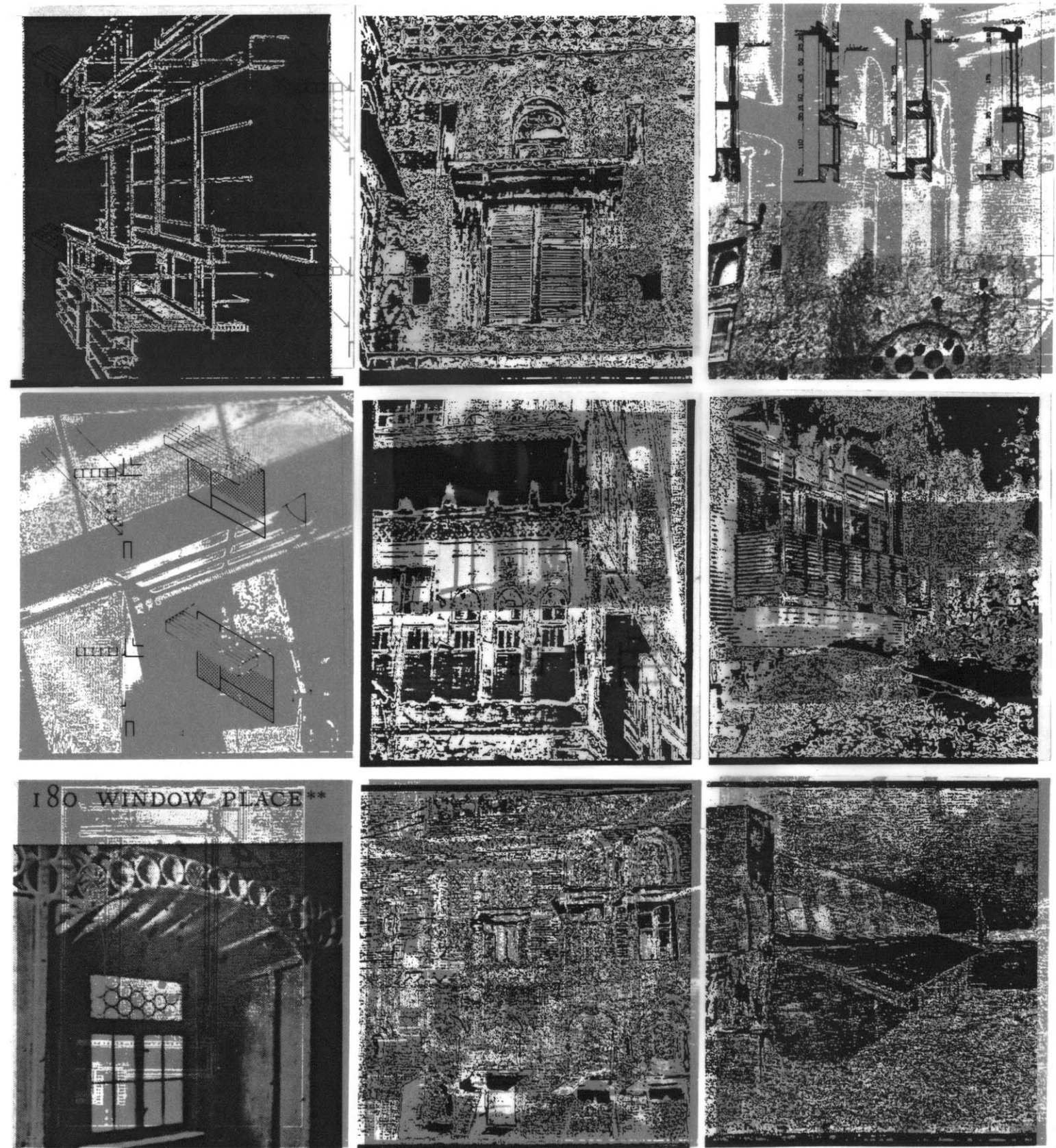


fig. 10

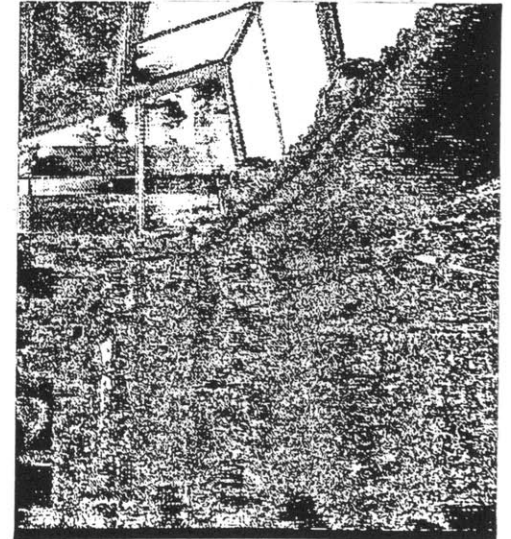
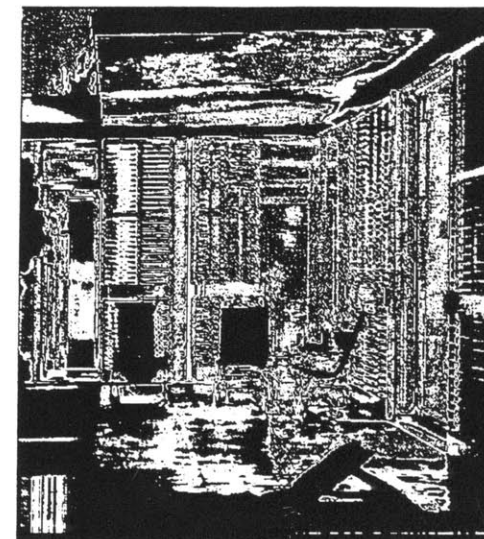
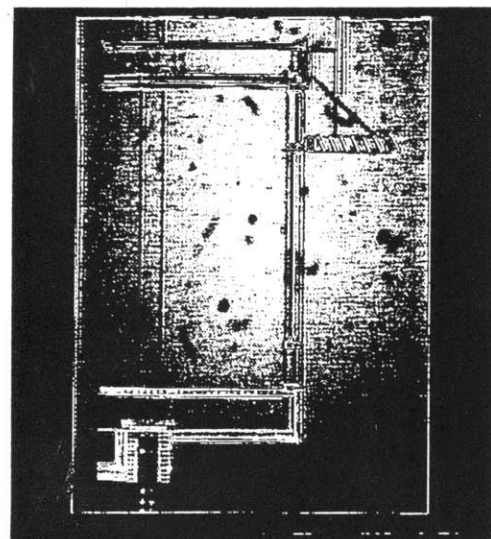
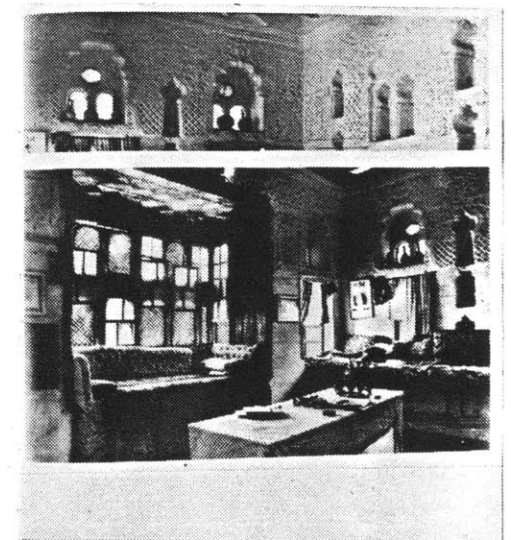
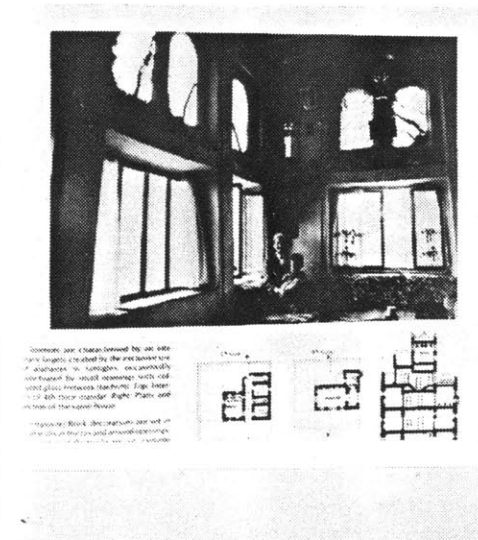
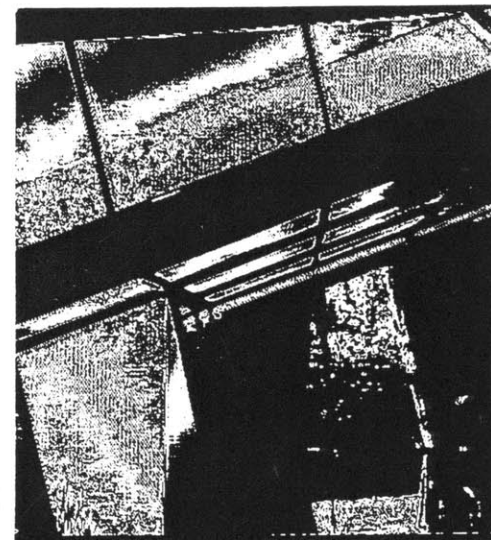
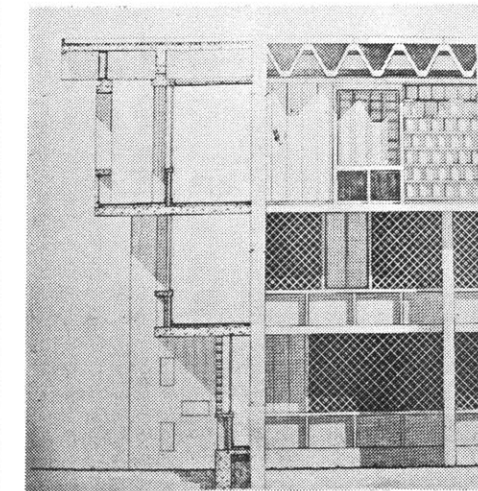
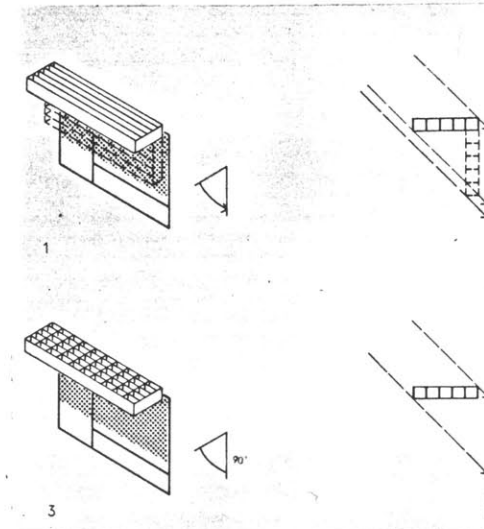
-- They are not uniformly significant. The various properties are known in differing levels of detail. Each property is known differently depending on the context within which it is considered.

-- The mechanism of inference is not uniformly directed. Thus the constraints which determine a property in one instance may be determined by it in another. " [7]

The combination of the internal searching and the external connecting is perhaps the most challenging aspect of the process, and is certainly the most challenging to represent.

Suppose that a designer is focusing on the problem of window design for arid climate conditions. The process of inquiry may include some questions about the relationship of the primary structural system and a proposed secondary system of closure, or the use of windows to mitigate special environmental features, or even the range of functional types possible for windows. The options are virtually inexhaustible and include the specific and the general; a valid query may be at the level of research into unique qualities of the desert climate that may provide new associations, or an exploration that attempts to formalize notions about edge definition. By focusing one's attention to the use of visual references an initial query can easily lead to associations with properties that weren't framed in the original question. Simple associations may work vertically or horizontally in a hierarchy; a larger category that includes windows as a subgroup may be valid in the problem, or other equivalent properties in the original frame may be valid in relation to each other.

For example, I am interested in seeing a few references that deal with windows in relation to the primary and secondary system. An initial search is directed at collecting images of windows in general; this is a process by which a few images can be weeded out to serve as key starting points from which more precise properties will be isolated. I determine that operable closure and edge definition are properties of the secondary system, which can be described as a behavioral component of the primary and secondary system context. Several more images are added to the collection and a few are eliminated. Each level of description defines more explicitly the domain of each preceding interaction. The focus has narrowed to a set of images that are implicitly related by virtue of the



process of connecting isolated descriptions into a serial sentence: "primary and secondary - secondary system - operable closure and edge definition". In a database of images the string of descriptors are the facts of the images. What's in the image is right before our eyes, so to speak, but a critical understanding of their contents of facts and their relation to the facts of other images is not directly apparent. An expert must keep account of the above facts primarily in terms of their relations to be able to answer the questions, "Now that I've found some images, what knowledge do they provide to the current problem? Why is it a reference? What does it inform me of? How is the knowledge applied? Are the images related and how? What about supplementary information sources? How does an automated system respond to the requirements and actions of the above questions? "

In order to go beyond the obvious such as referencing images randomly or through the card-file in the library, an automated system must express representations of knowledge. There needs to be a clear attitude about the use of computers in this regard; it's entirely reasonable to have a tool operate in its simplest capacity as one familiarizes oneself with its operation. Eventually you will want to exploit its usefulness to whatever limit. The point is that as the interaction with the automated system proceeds by levels, or steps, there is a desire for an active increase in the amount of information and the varieties of ways to get at it. Constructing methods of referencing and strategies for their use is a formidable task, one in which I must defer to my own methods. From time to time I also look to other sources to help re-frame the problem. Again I refer to Margaret Boden ,

"There are indefinitely many representational schemes one might employ in making sense, of one sort or another, out of a phenomenon.... But any such interpretative scheme requires an analysis of the phenomenon into the 'parts' that it takes to be significant. Different questions may require reference to different cues.... Equally, one and the same question about the target domain may be answerable with the help of different types of cues from the representational domain.... Cues, in short, have to be defined intentionally, by reference to the method of representation assumed in the interpretative process.... Intelligence, artificial or otherwise, depends largely on finding more and more ways to treat features of the world as cues to matter of interest." [8]

REFERENCES

- [i] Herbert, Frank. *Dune*. New York: Berkeley Publ. Corp., 1965; p.69.
- [ii] Rasmussen, Steen E. *Experiencing Architecture*. Cambridge, MA: MIT Press, 1962; p.190.
- [1] Abelson, Harold and Sussman, Gerald J. *Structure and Interpretation of Computer Programs*. Cambridge, MA: MIT Press, 1985; p.1.
- [2] Miller, Frank C. "The Electronic Studio," *Symposium : Computers in Design , Emerging Research Directions*. Cambridge, MA: MIT Department of Architecture and Urban Planning Computer Resources Lab, Graphic Systems, Inc., 1986.
- [3] Boden, Margaret. *Artificial Intelligence and Natural Man*. New York: Basic Books, Inc., 1977; p.179.
- [4] Schon, Donald A. *The Reflective Practitioner: how professionals think in action*. New York: Basic Books, Inc., 1982 ; p.49.
- [5] Kalay, Yehuda E. "Redefining the role of computers in architecture: from drafting/modelling tools to knowledge-based design assistants," *Computer Aided Design*. 7 (Sept. 1985); pp.319-328.
- [6] *Merriam-Webster Dictionary*. Springfield, MA: G & C Merriam, Co., 1974.
- [7] Anderson, James. "Lecture Notes: Relational Formal Language(4.182)." Cambridge, MA: MIT, 1987; p. 2.
- [8] op cit., Boden. pp.183, 185.

CHAPTER 2

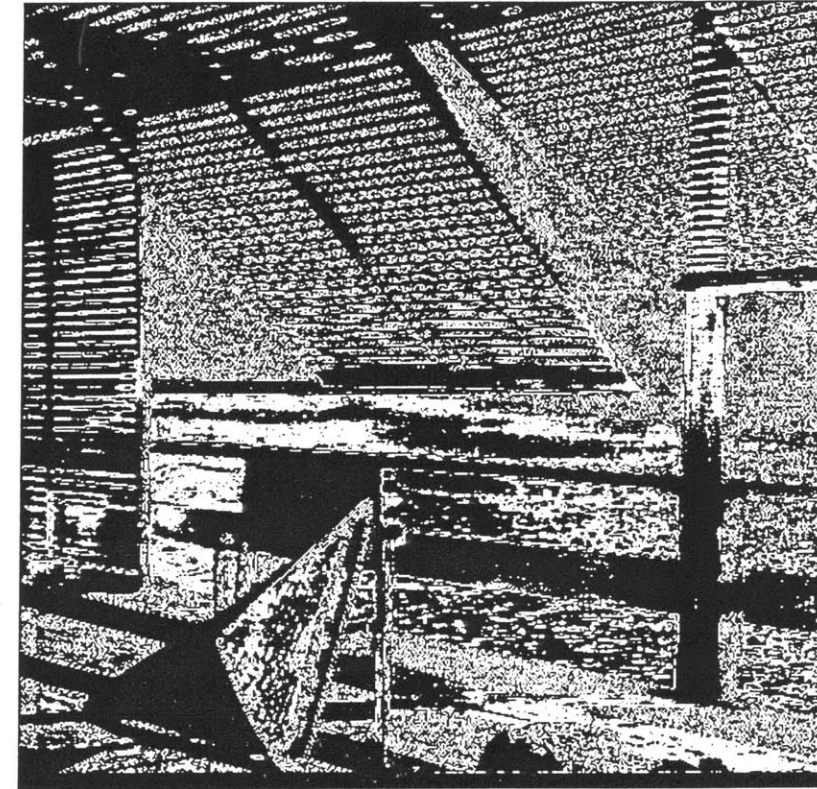
VISUAL CUES

Observation is the beginning of understanding. Yet there are the further issues of deciding what to observe and how to describe something so that it is useful in design.

---RELATIONAL FORMAL LANGUAGE
Course Description 1987
T.Chastain and J.Anderson

Description must fit purpose, and a good description is notable for what it chooses to ignore.

---KEVIN LYNCH
A Theory of Good City Form
[1]



■ ATTRIBUTE HIERARCHY:

```

file (inferred from):
  keycom1 (inferred from):
    SCOPE, in class description (inferred from):
      single (continued)
      Scope
    BEHAVIOR, in class description (inferred from):
      single (continued)
      Scope
      Behavior
    PROPERTY, in class description (inferred from):
      single (continued)
      Scope
      Behavior
      Property
    SPECIFICATION, in class description (inferred from):
      single (continued)
      Scope
      Behavior
      Property
      Specification
      single (continued)
      Scope
      Behavior
      Property
      Specification
      request (continued)
response (inferred from):
  verb (inferred from):
    sentence
  object (inferred from):
    sentence
single (inferred from):
  request (continued)
request (inferred from):
  modifier (inferred from):
    sentence

```

■ INFERRED ATTRIBUTES:

In the main kb:

```

verb
object
modifier
response
request
keycom1
single
fire_file

```

■ INPUT ATTRIBUTES:

In the main kb:

```

sentence
respond
Scope
Behavior
Property
Specification

```

■ keyimage = i6

```

[ description :
  Scope = primary secondary <m>
    env_controls <m>
  Behavior = light modulation <m>
    secondary system <m>
  Property = variable closure <m>
    operable closure <m>
    fixed closure <m>
    wooden screens <m>
  Specs = Source <m>
    Source_ref <m>
    Represents <m>
    Image_type <m>

```

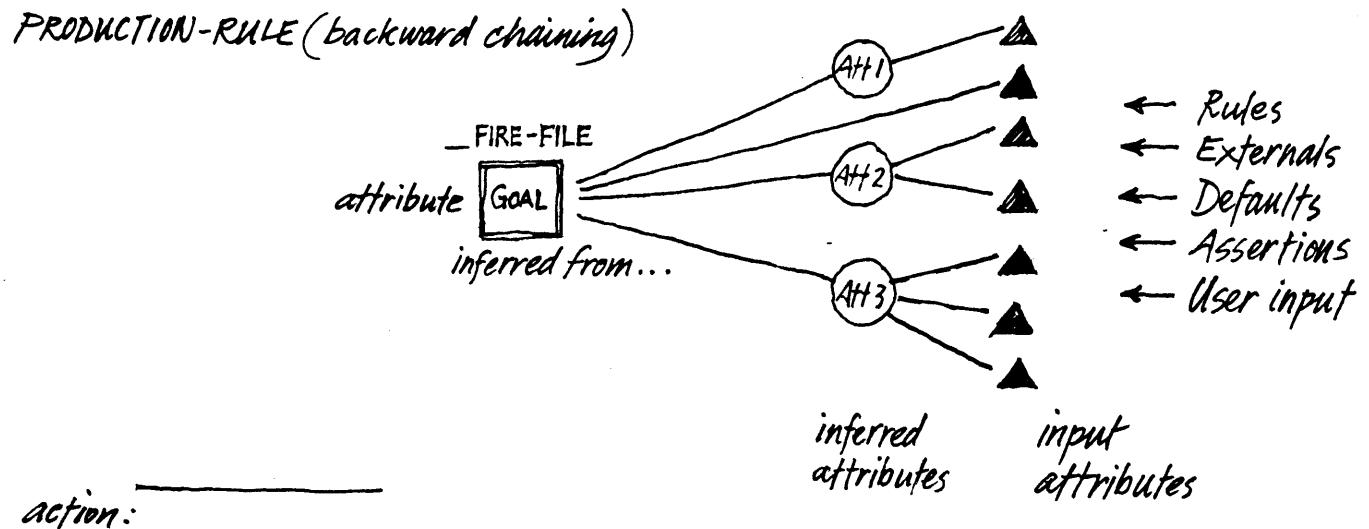
fig. 11

The use of visual images covers a wide spectrum of intention. To suggest that any image stands isolated from any other belittles the human capacity to perceive. The world is primarily a visual experience and the process of seeing is subtle in the least. We are consciously and unconsciously assimilating, collating, and relating myriad images at every waking moment. Our memories and perceptions are not random snapshots that can be arranged in any order and still preserve the original experience; generally, our emotional and intellectual attachments to what we see are serial images and composites of serial images that are quite flexible to re-combination.

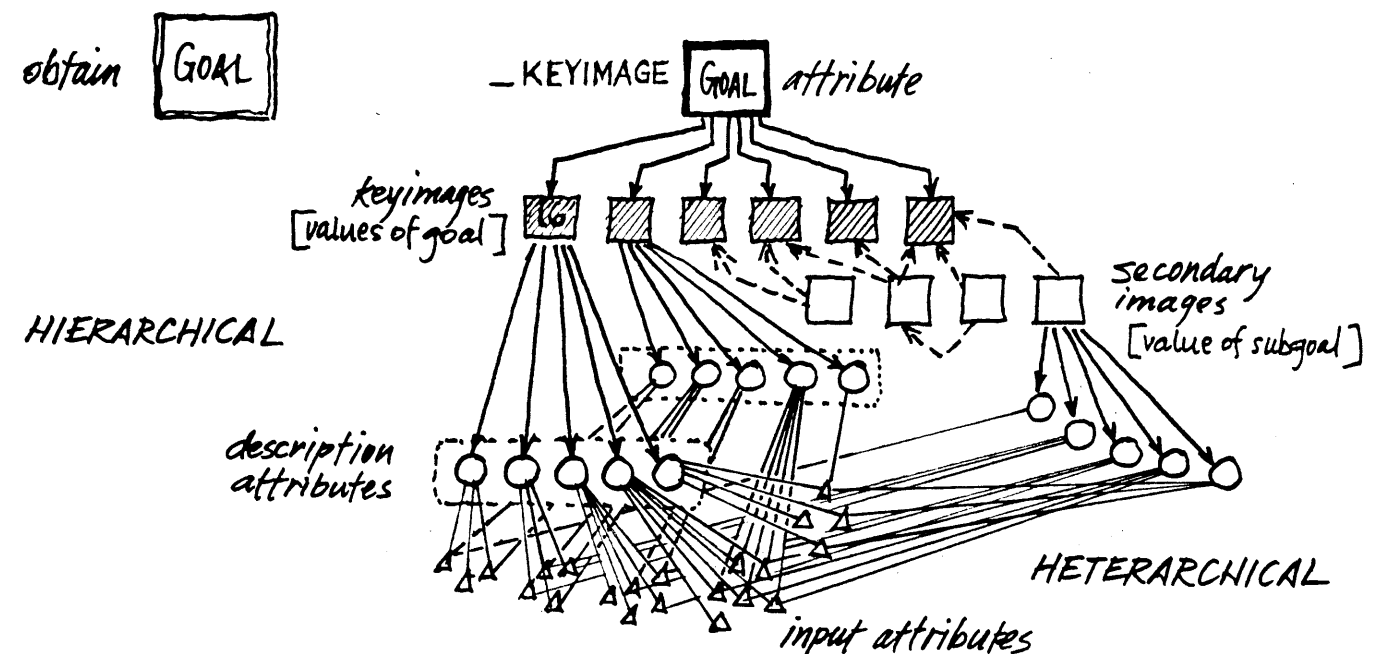
Our ability to construct patterns out of visual information is an important aspect of how we represent knowledge about the world around us. Patterns are summary suggestions, or cues, from which we construct hypotheses. A two-way methodology that readily accommodates the function of hypotheses in a design environment includes "top-down" and "bottom-up" thinking. In more global, wholistic(top-down) thinking a hypothesis can direct our search for alternative confirmatory cues, which are generally composites of many details. When there is no specific reason to operate from a specific hypothesis then our attention may be directed towards a bottom-up review of details. Design inference proceeds from the particular in addition to the global, so the precedence exists for specifying an automated system that is capable of inferring from either the top or the bottom.

The simplest reference to a description is a single image, yet an architect often creates in an environment of multiple images. The single image juxtaposed with an array of other images provides a richer display where ambiguous references can be accepted for their qualities as well. Arrays are built from categories of related images, related images from unrelated categories, unrelated images from several categories, ad infinitum. The need to view the growing collection of images, or even distinct arrays, as if pinning paper to display boards, is a fundamental requirement of the automated system. In an architectural world of no machines, the designer is left to more conventional media: tacking images to a surface that will allow moving the images about in relation to each other, removing some and replacing with others, and adding diagrams(new images) created from the collection or the design. In a machine world the, computer must offer the same flexibility, yet add a dimension of use unique to

PRODUCTION-RULE (backward chaining)



HYPOTHESIS-TEST (minimal set covering; backward chaining)



itself. The definition of which is an underlying aim of this paper. What's apparent is that the variety of possible presentations will influence the nature of the interpretative model. The interpretative model that I refer to is the knowledge-based expert system.

"An important subset of the general area of expert systems concentrates on explicitly representing an expert's knowledge about a class of problems, and then providing a separate reasoning mechanism(usually called an inference engine). These kinds of systems are called 'knowledge-based' expert systems.... Like conventional programs, expert systems usually perform relatively well-defined tasks. Unlike conventional programs, expert systems also explain their actions, justify their conclusions, and provide end users with details of the knowledge they contain." [2]

The proposed model functions around an inference process where a series of queries are used to build a descriptive string. This 'sentence' acts as a pattern that sifts for the images fitting the description. Although the method is exactly like a database query, "Show me images of secondary systems", the inference engine searches for images and returns the values with a 'certainty factor'. The certainty factor is a representation of how closely the description matches with characteristics of the image. Unlike a visual database which stores the records of all available images, the knowledge-base can only function well in a search process where the knowledge store is relatively small. The strategy is to focus the design of the expert system on a group of images that are best representatives of a type of knowledge. The images are "keyimages" or cues to knowledge within a frame of reference. The fact that the knowledge-base can weight its inferences(ie., 'always' matches: 100% certainty, 'high': 75% certainty, or 'medium': 50%, or 'low': 25%) allows an interaction based on less exact criteria. This is possibly a way to see more images if the best matches are not satisfactory, or to view the best matches within the context of nearly related images. From this point the architect can either choose to interact more directly with specific images or diverge into compiling more varied arrays and image descriptions. If the former choice is taken, then the actual knowledge-base function is exercised. If the latter

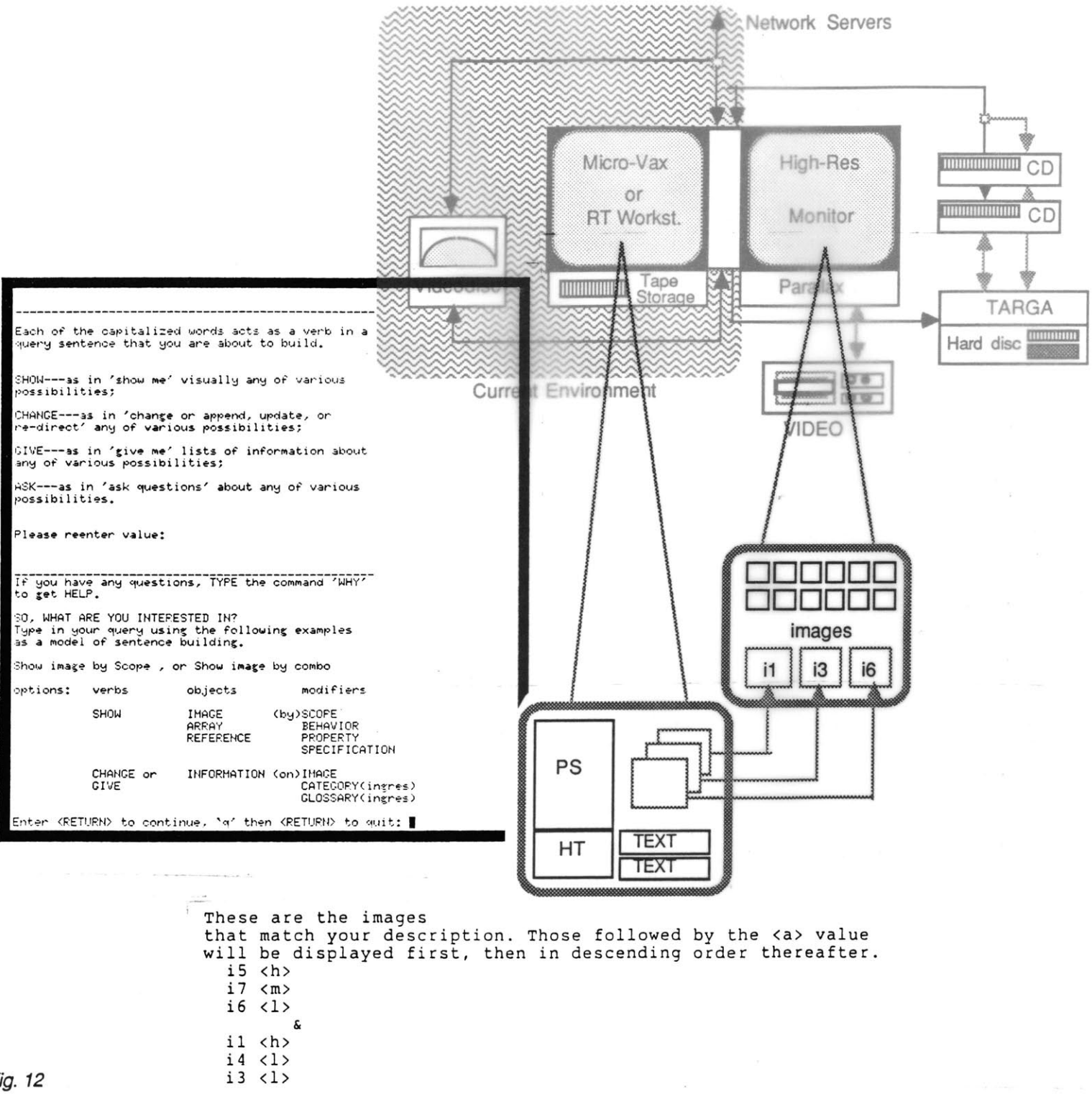
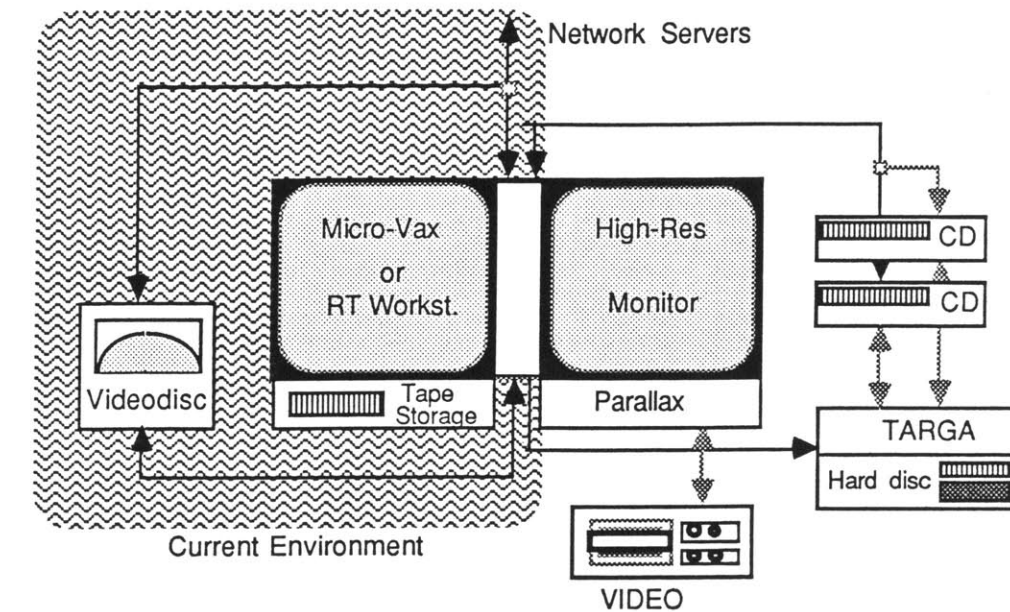


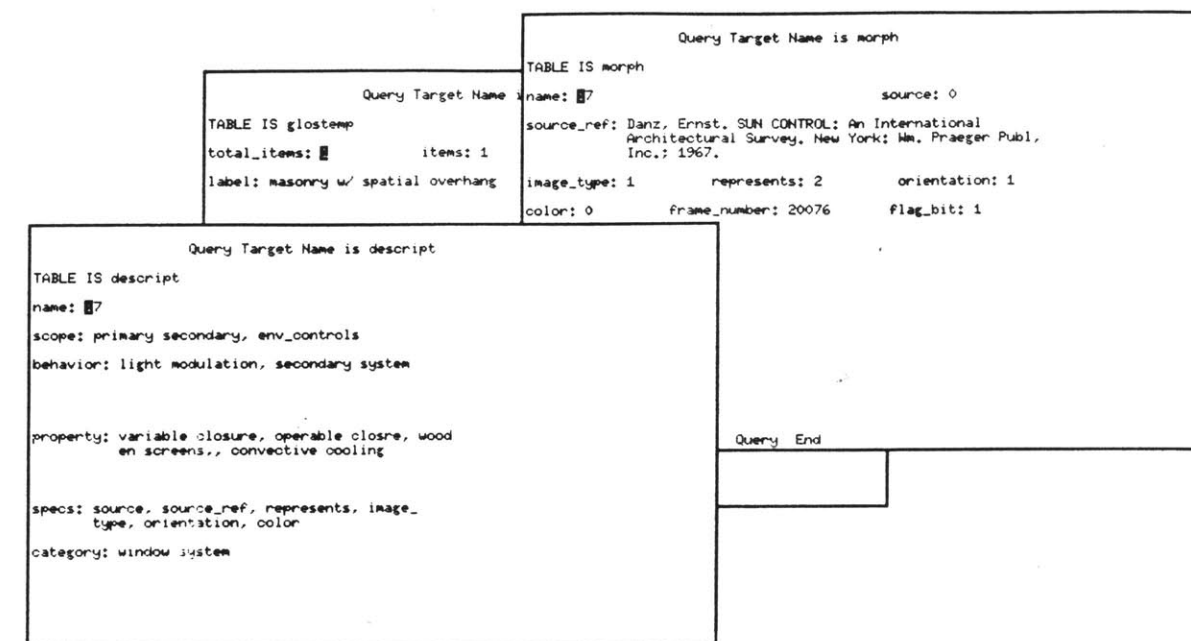
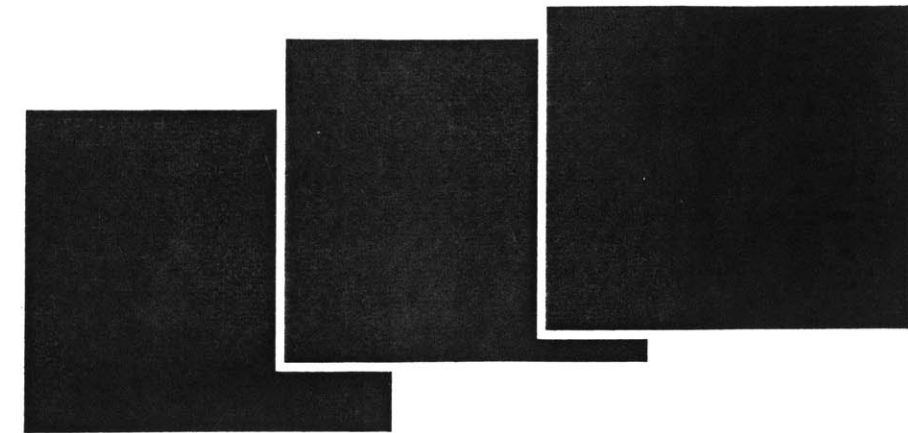
fig. 12

itself. The definition of which is an underlying aim of this paper. What's apparent is that the variety of possible presentations will influence the nature of the interpretative model. The interpretative model that I refer to is the knowledge-based expert system.

"An important subset of the general area of expert systems concentrates on explicitly representing an expert's knowledge about a class of problems, and then providing a separate reasoning mechanism(usually called an inference engine). These kinds of systems are called 'knowledge-based' expert systems.... Like conventional programs, expert systems usually perform relatively well-defined tasks. Unlike conventional programs, expert systems also explain their actions, justify their conclusions, and provide end users with details of the knowledge they contain." [2]

The proposed model functions around an inference process where a series of queries are used to build a descriptive string. This 'sentence' acts as a pattern that sifts for the images fitting the description. Although the method is exactly like a database query, "Show me images of secondary systems", the inference engine searches for images and returns the values with a 'certainty factor'. The certainty factor is a representation of how closely the description matches with characteristics of the image. Unlike a visual database which stores the records of all available images, the knowledge-base can only function well in a search process where the knowledge store is relatively small. The strategy is to focus the design of the expert system on a group of images that are best representatives of a type of knowledge. The images are "keyimages" or cues to knowledge within a frame of reference. The fact that the knowledge-base can weight its inferences(ie., 'always' matches: 100% certainty, 'high': 75% certainty, or 'medium': 50%, or 'low': 25%) allows an interaction based on less exact criteria. This is possibly a way to see more images if the best matches are not satisfactory, or to view the best matches within the context of nearly related images. From this point the architect can either choose to interact more directly with specific images or diverge into compiling more varied arrays and image descriptions. If the former choice is taken, then the actual knowledge-base function is exercised. If the latter



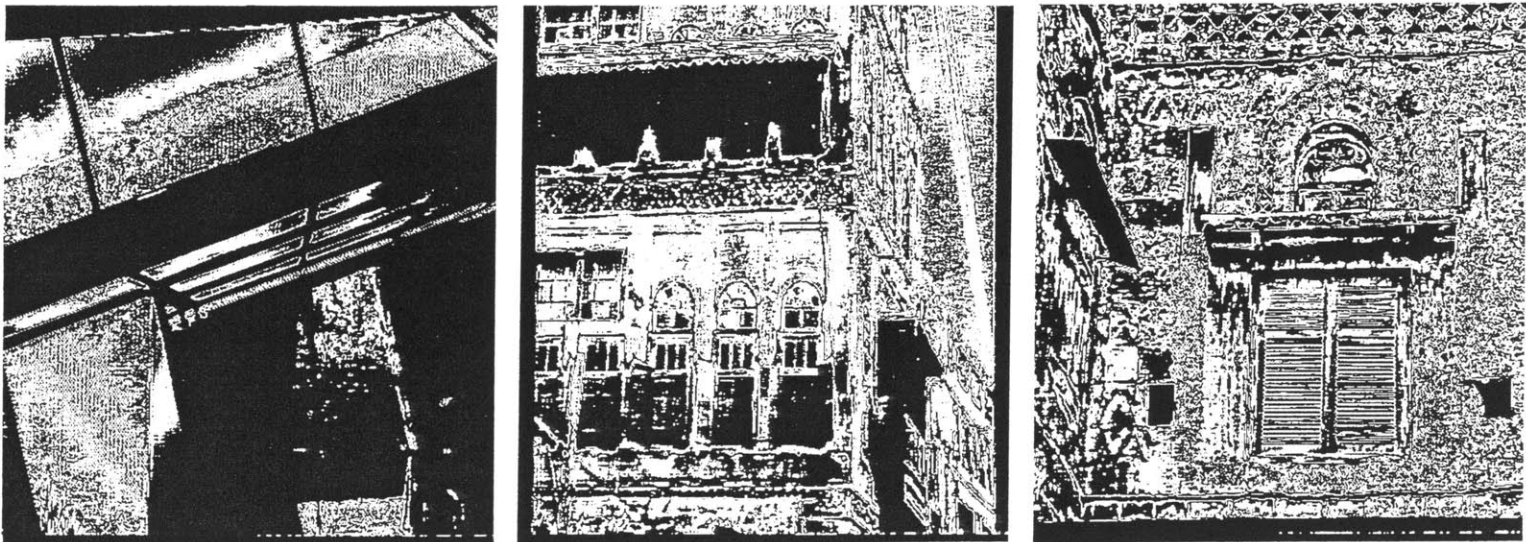


INGRES/MENU		Database: amb
To run a highlighted command, place the cursor over it, press the <MENU KEY> and type "go".		
To run a menu command, press the <MENU KEY> and type that command.		
Commands	Description	
QUERY	RUN simple or saved QUERY to retrieve, modify or append data	
REPORT	RUN default or saved REPORT	
GRAPH	RUN saved GRAPH	
QBF	Use QUERY-BY-FORMS to develop and test query definitions	
RBF	Use REPORT-BY-FORMS to design or modify reports	
GFB	Use GRAPH-BY-FORMS to design, modify or test graphs	
ABF	Use APPLICATIONS-BY-FORMS to design and test applications	
TABLES	CREATE, MANIPULATE or LOOKUP tables in the database	
VIFRED	EDIT forms by using the VISUAL-FORMS-EDITOR	
QUEL	ENTER QUEL statements using the TERMINAL MONITOR	
SREPORT	SAVE REPORT-WRITER commands in the reports catalog	

fig. 13

route is chosen, then an expanded, more powerful set of visual database tools are available for use.

I've been purposefully avoiding a direct discussion of the knowledge capabilities in the referencing system, because 1) I wish to describe its supporting environment and interface and 2) I wish to discuss its role within an actual design inference, which is the intent of chapter three. The specification for a visual database is designed after a working model created by Paul Paternoster in his thesis, CATALOG/DIALOG [3]. His model makes the most sense for the visual environment that parallels and supports the expert system. Paternoster's thesis demonstrates the use of a global database that "catalogs" the location of images by category as they are established on a fixed storage medium like a videodisc. The term fixed refers to the fact that once the images are recorded to the disc no changes or additions are possible. The categories and locations of individual frames are therefore locked into place. As is often the case when a person is referring to an image there are no defined boundaries as to how one might describe the image or the category it resides within; the original name might not seem so appropriate in a new context, especially if the image is a source for some fragment of itself or if it is called up by new associations. Paternoster was interested in devising a means to accommodate a more flexible, user-defined tool for accessing visual resources. His solution was to enable the user to reference images and create a second-level database(separate from the global database) that could act as a glossary of new, defined "on-the-fly" categories, or new word-image associations. The glossary would refer to the same set of frame numbers as the global database, but the relational information would be different; a new piece of information is appended to the original without suffering from the limitation of a "read-only" storage medium. The most powerful capability of the system is the means by which a user can edit and create new images with the aid of an image processing application. The fixed number of images on the disc or similar storage media becomes only a relative limitation since the user can modify and extract, diagram and notate images as they see fit. The *glossary* actually becomes a tool of immense use to the designer; the cross-referencing and definition of word associations that describe images is only limited by the needs and imagination of the user.



Query Target Name is descript

TABLE IS descript
name: 7
scope: primary secondary, env_controls
behavior: light modulation, secondary system

property: variable closure, operable closure, wood
en screens,, convective cooling

Query Target Name is descript

TABLE IS descript
name: 3
scope: primary secondary, env_controls, form_org
behavior: light modulation, secondary system, ven
tilation, relationship between prima

property: operable closure, variable screens, two
or three part ordering system

specs: source, source_ref, represents, image_t
ype, orientation, color
category: window system

Help Next(control_F) Query End

Query Target Name is morph

TABLE IS morph
name:
source:
source_ref:

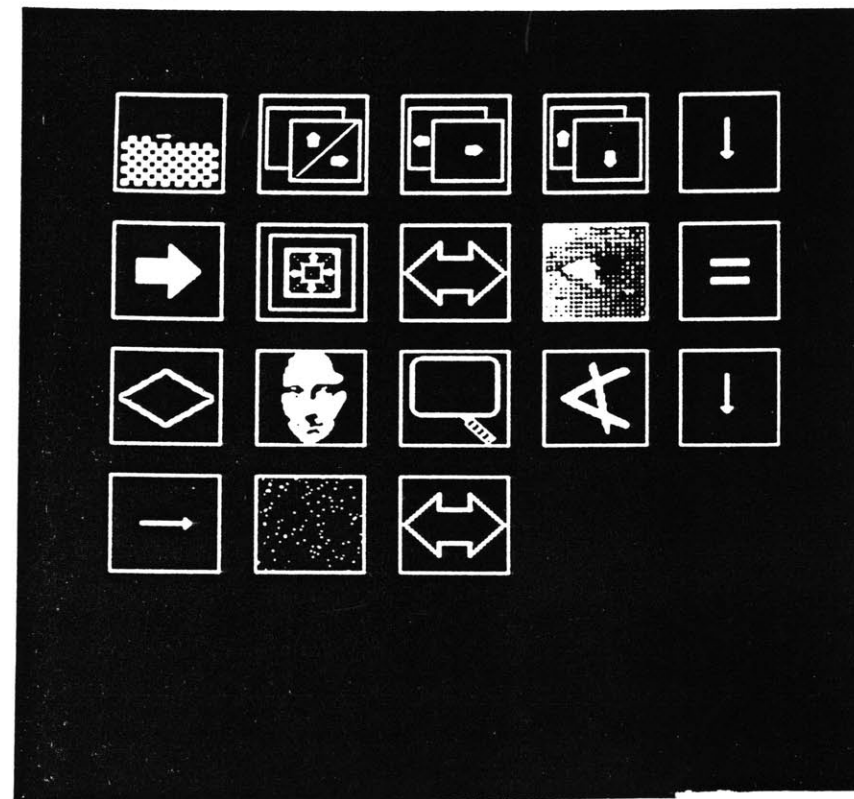
image_type: represents: orientation:
color: frame_number: flag_bit:

Help BlankForm Go LastQuery End

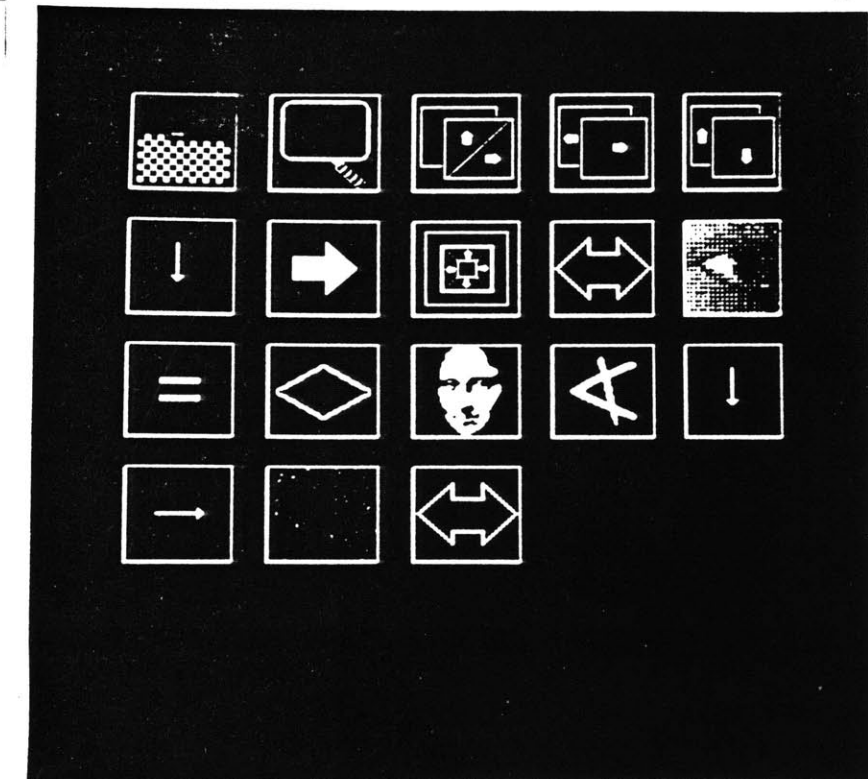
The critical aspect of any interface to the database system is the reduction of actual interaction with the most primitive level of data : the catalog of "stuff" that lists all the information about an item, most of which is unessential for any one query from the user.

" In addition to comparing strictly quantitative data, it is possible to order, to organize symbolic data. To decide whether two symbols are the same, or to establish the ordering relationship between them, develops to an issue of representation and comparison. Once it is possible to order this data, the next question becomes: what information or knowledge is associated with those symbols which may be automatically manipulated?" [4]

At MIT's FILM/VIDEO Group of the Media-Lab, Glorianna Davenport is working with a concept that uses icons as visual tags to lists of images, specifically raw or edited segments of live video or stills. These "edit-lists" represent a working concept of their contents, essentially becoming iconic visual objects. The objects are readily manipulated by database functions and are the focus of a richer relational system that conveys visual, rather than verbal information regarding themes and presentation requirements(the information needed to specify a display environment, size, duration, and location). In summary, there is essentially a global repository that catalogs the image resources by storage place, category name, and actual location on the storage medium. A temporary adjunct to this database is created from designating images from a particular category where individual images are isolated for later viewing or editing. Operating in the background is a database that catalogs specification data: the image's category, origin or source, type(photo, film, drawing, etc.), color, orientation(horizontal, vertical, null), and its representation(live-action, portraits, renderings, etc.). Both the temporary glossary and the specification databases are related to the global database via generally transparent category names. The meaning of transparency is suggestive of images having associations with other categories, but the ability to "see" that connection is dependent on relational descriptions that are variable over time, whereas the original category designations may not. The idea that the initial compiling of the data onto a read-only storage medium is a starting point and not a static end in itself is the crucial point. However, the technology of



The sequence to be rearranged.



The new sequence redisplayed.

storage systems is rapidly changing the limitations that Paternoster had to face three years ago. Write capabilities with optical discs are now possible, which opens the door to even more powerful relational applications. The designer ideally wants to access images that are dense in associations; generally there is not one image that can achieve that distinction, therefore the designer will sift and sort together a set of interrelated images that provide the variety of qualities, references, and relations that are needed for the problem frame.

The generation of new associations within the same database is a step higher in usefulness, but the architect is also interested in manual interaction with the images. By this I mean that architects look as well as draw. Figurative drawings, or diagrams, help to draw out the associations or relationships within images.

"The diagram as intermediary loosens rather than loses, developing its attributes from both aspects of opposing dualisms and thus presenting rather than representing, explaining rather than embodying. The proposal of the diagram is tentative and temporary. It makes more apparent the processes of type referencing that propose strength, certainty, and origin and that establish the 'lateness' of the activity of reading buildings, at the same time providing for the endless reinspection of the specific attributes of a building. Such ongoing examination exposes weakness, uncertainty, and originality and establishes the revisionary character of interpretation. The diagram, then, can be a simultaneous discussion both of the thing itself and of what it manifests." [5]

The possibility of using images from the database as sources for diagrams, which in turn become new images in the database, strengthens the visual system by adding another level of relational complexity. New images created from existing in turn become new images in the database, strengthens the visual system by adding another level of relational complexity. New images created from existing are cross-referenced with those already in store and then can be shuffled about with new associations: all cataloged in the glossary database. The editing of images falls into the same realm; editing by notation or sketching applications, both of which are independent of the image source, is an adjunct to the database and defines another component of the entire referencing network. Notation is a formal marking application, which includes both text and graphics, used to isolate,

point-out, or demonstrate aspects of an image that may be used in later references. Sketching is a less formal application where the designer can actually diagram over the image, or it is an application that allows on-the-fly diagramming in a simple drawing editor. The edited images are then free to be designated as new images and categorized however. The use of the notation and sketch facilities are not necessarily limited to the database of existing images, but may become an entirely robust image-processing system of its own. The glossary, notation, and sketching facilities are a means to erode the limitations created from the static state of the database when it was initially created. Similarly, the visual database as a whole is a means to erode the limitations of the knowledge-base when it was created.

What I have proposed is a system that allows interpenetration of two types of diagramming, one visual and one verbal. The verbal diagram is the descriptive classification scheme: the series of hierarchical word associations that the designer builds during the query process. The structure of the description statement is based on the work of James Anderson and Tom Chastain:

"It is both the ability to name the observed phenomena once you recognize it in order to locate it in the future relative to other names, and it is the ability to use the instance as a criteria in its own right, to classify other instances, or other configurations according to the features which are important." [6]

I have adapted the syntax wholesale with minimal criticism because of its natural fit with the inferencing structure inherent to the expert shell. The hierarchical relationship of the words in the classification scheme acts as a descriptive tag that goes beyond a mere naming of the image. A name may help locate the image relative to other names, but there is no indication of the contents. Each syntactical element of the description helps define the level of detail for each preceding element. The syntax is also responsive to the notion that similar descriptive terms may have criteria unique to its case. Specifying a grammar of this nature allows inflection, the fact that there are levels of interpretation in design thinking. An architect uses a "language of description" that often undergoes slight changes of form as the situation warrants it. The level of design thinking an architect uses towards the design of a room versus a house, or versus a cluster of

"The notion of a design grammar is based on an analogy between design and language: in the same way that rules(a syntax) exist for the way in which the elements of language are put together to form meaningful sentences, it is suggested that rules can be identified for the way the elements of a design can be put together to form designs. A particular set of rules lead to a particular style or school of design, which may be the result of many different designers sharing the same language or the unique work of a single master. In the past, these rule shave rarely been written down. Rather, they are the way in which designers respond to a situation; the way they recognize a pattern in the environment, in a design brief, or in the state of a developing design and match an appropriate response to that pattern in the design." [7]

houses, is unique to each case although a common language with a common syntax is used throughout.

Each descriptive "label has a number of parts, called *fields*. Each field serves to bind the reference to its property relative to one design issue." [8] Each descriptive sentence has the syntax form:

(scope, (behavior, (property)), specification).

The classification scheme resides in both the knowledge-bases and the visual database. The hypothesis-test inference engine that weights the image's match to a description follows a similar syntactic structure. Each image, which is a possible value of an attribute called 'keyimage', has a list of attributes that are descriptions of the characteristics each image possesses. The characteristics are attributes themselves and are of the form shown above. The production-rule inference engine is designed to use rules to infer knowledge about the images, which are solutions from the hypothesis-test inference process.

Stepping back for a moment to review my assumptions, I realize that , at one level, the system I keep referring to seems to have grown from a global hypothesis I made early on. The details of which were patiently discovered and molded through the creation of the expert system. However, what is not apparent is the fact that I have utilized a backward-chaining method as well. Top-down decisions such as using a keyimage model result directly from the constraints of the expert shell, which is unable to effectively store large amounts of data. subtleties of interaction with the expert system and the relationships between the production-rules and the hypothesis-test are more easily determined from the details of the design process. The sketch problem, to be presented in chapter three, was ultimately, and rightly so, the determinant of how the system should operate. Design inference in action was the only way of getting at the issues and details that would shape new hypotheses, re-frame others, and highlight the ways images are referenced and knowledge is acquired.

In this chapter I have attempted to describe a visual context of architectural design in general and a system to support its near equivalent representation in an automated environment. Central to my hypothesis is the use of visual cues: references inferred from a descriptive grammar and related to patterns of knowledge by rules. The task remains to clarify and formalize the rules by which a particular response to a pattern occurs.

"A design grammar is basically an attempt to formalize some of those responses by a system of transformation rules that replace one pattern with another. It consists of a series of rules of the generic form:

If A, then B

where A is in the most general sense a pattern which exists in the design context(including brief, climate, building regulations and so on), and B is a new pattern which replaces or responds to A in the design if the rule is fired. Executing a design grammar depends on finding patterns in the partially developed design and in conditions outside the design which match the left hand side of a rule." [9]

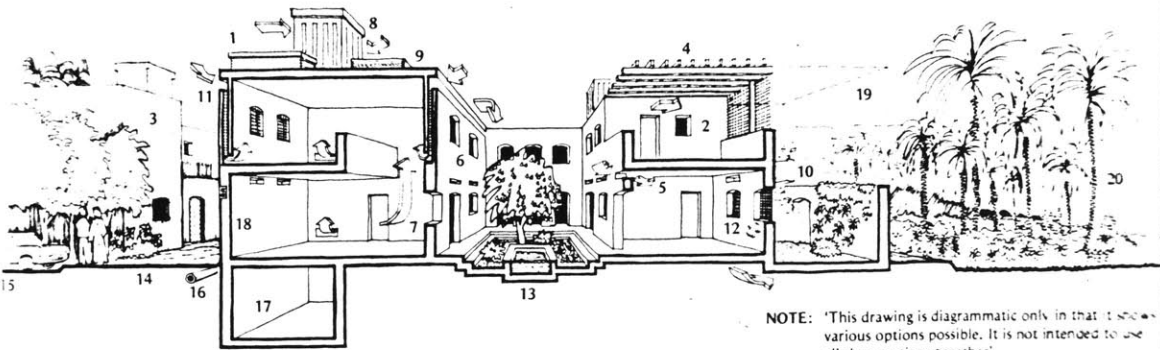
REFERENCES

- [1] Lynch, Kevin. *A Theory of Good City Form*. Cambridge, MA: MIT Press, 1981; p. 358.
- [2] Software Architecture & Engineering, Inc. *KES HT: Knowledge Base Author's Manual*. Arlington, VA: Software A & E, Inc., 1986; p. 2-1.
- [3] Patemoster, Paul. *Catalog/Dialog*. Masters Thesis(MSVS). Cambridge, MA: MIT, 1984.
- [4] Anderson, James. "Lecture Notes: Relational Formal Language(4.182)." Cambridge, MA: MIT, 1987; p. 5.
- [5] Graf, Douglas. "Diagrams", *Perspecta*. 20 (1983). Cambridge, MA: MIT Press; p.43.
- [6] op cit., Anderson. pp.8, 9.
- [7] Hansen, Neil L. R. and Radford, Antony. "Living on the Edge: a grammar for some country houses by Glenn Murcutt," *Architecture Australia*. 5 (July 1986); p.66.
- [8] op cit., Anderson. p.11.
- [9] op cit., Hansen and Radford. p. 67.

DESIGN INFERENCE

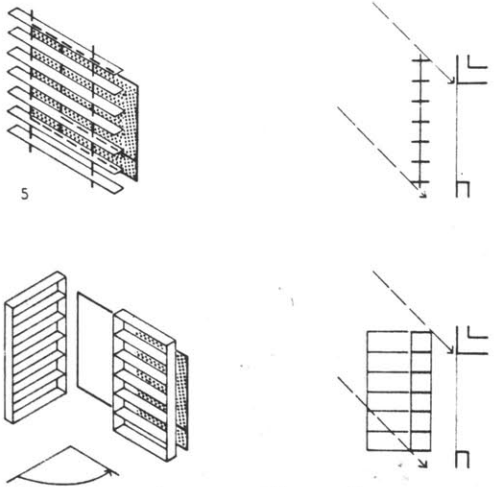
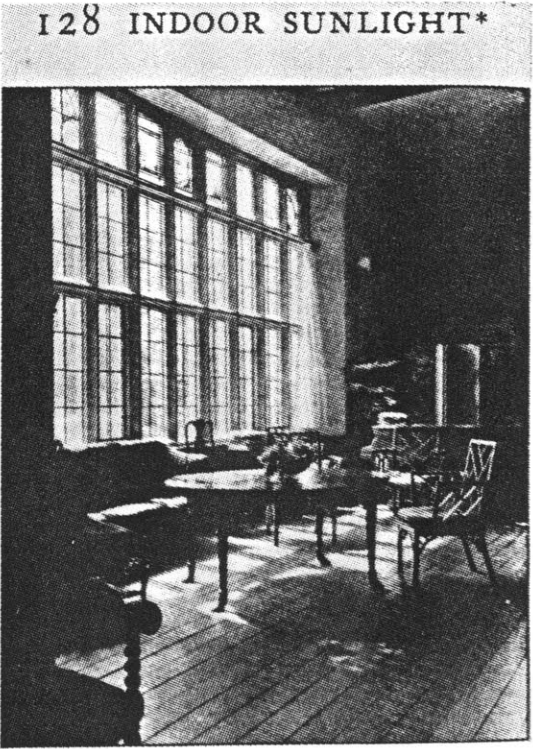
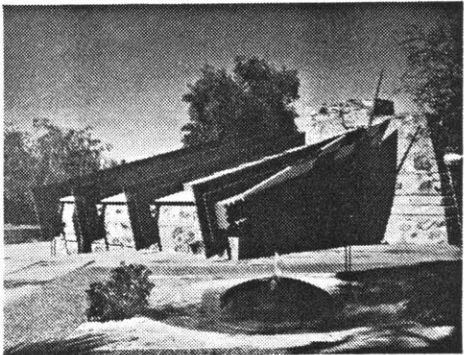
For the process of analysis, it would be useful to have available an 'interpreter', a device that could simultaneously negotiate between a series of opposing propositions, between typologies that identify the organization and those that identify the components between the specific qualities of the building and the general qualities of the architecture, between processes of cognition and processes of perception, and between the dynamism of operation and the stasis of configuration. This is the role of the diagram.

— DOUGLAS GRAF
"Diagrams"
[1]



Schematic Section through Courtyard House, showing:

Prickly pear cacti, *Opuntia sp.*, are among the most common of the cacti and are found in many areas outside the desert. The flattened fleshy stems store moisture. In times of prolonged drought, the water-depleted pads become wrinkled and shrunken.



The original inspiration of the thesis was a studio project from the fall of 1986. The project was the last of three modules for the term and was taught by Judith Chafee FAIA, Tucson, Arizona. I'd like to reiterate a few issues from the brief before describing the scope of the design problem and general issues of design in arid climates. The design task that Chafee proposed served as an outline that addressed the unique problems of designing structures that would be "suited" to their climate. The project was carefully constructed by Chafee to help clarify the critical issues that might be described as a grammar and syntax of design elements appropriate to desert conditions. I have chosen to isolate a small element of focus from my studio work to act as a new focal point for discussing and implementing an automated design knowledge system. The design elements are visual images and the knowledge contained within those images.

Briefly, the project site has no utilities and is an isolated tract of 160 acres in the mountains near Tucson, Arizona. An architect had joined forces with two other households to build a pilot housing cluster that must occupy at least 12 acres(as stipulated in the local zoning ordinance). Shared elements of the program include water harvesting, water storage and transport, gardens and greenhouse, farm animals, methane generation and compost, swimming pool and guest house. *"The purpose of this project is to participate in a design process which integrates unfamiliar elements of climate, building techniques and living patterns."* [2]

The combination of arid climate, shared elements of water conservation, animals and small scale agriculture suggested an integrated system(design and services) to link the individual households as a true cluster. A mechanically conditioned interior environment was both energy intensive and inappropriate to the requirements of the location. Therefore, a careful combination of design parameters generated from a sensitive approach to climate, to sun movement, to diurnal changes in temperature, to flora-fauna interrelationships, and to human needs was necessary to achieve a dynamic, liveable prototype.

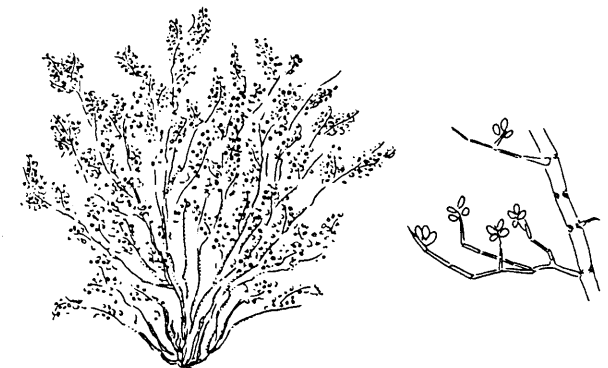
Climate plays such a significant role in our perception of the desert that an understanding of that which makes the desert what it is will aid us considerably in the process of creating designs that mitigate, in the desert's own terms, the harsh conditions. Rational alternatives to suburban sprawl in the southwestern

deserts requires an architect to establish a sense of "place" --- the unique environmental and cultural characteristics that differentiate one place from another. Cultural information is quite valuable in this case since there are several tribes of Native Americans who have occupied for centuries, and still occupy, the lands of the Sonoran Desert. Although the information is hard to come by, native adaptations to the climate can serve as guidelines, rules-of-thumb, or precise knowledge of local patterns of phenomena, plant resources, and history.

An attitude of "appropriate and wholistic" design is relatively grass-roots in origin, but there is a significant effort by organizations and individuals to advance ecological principles through research and practice. Most notable are the efforts of The New Alchemy Institute(focus on whole energy and food systems) in Falmouth, Massachusetts, and Bill Mollison(the principles and practice of Permaculture) in Tasmania, Australia. The basis of my attitudes towards the design of the housing cluster are sympathetic to both. A requisite of the studio was to seek alternatives to the kind of development that encourages sprawl and a disregard for natural systems; therefore I have approached the architectural design of the housing cluster with these attitudes in mind.

"There is in all things a pattern that is part of our universe. It has symmetry, elegance, and grace --- those qualities you find always in that which the true artist captures. You can find it in the turning of the seasons, in the way sand trails along a ridge, in the branch clusters of the creosote bush or the patterns of its leaves. We try to copy those patterns in our lives and our society, seeking the rhythms, the dances, the forms that comfort. Yet, it is possible to see peril in the finding of the ultimate perfection. It is clear that the ultimate pattern contains its own fixity.

*from 'The Collected Sayings of Muad'Dib'
by the Princess Irulan" [3]*



The creosote bush, *Larrea tridentata*, is the most abundant and widespread shrub of the Sonoran, Mohave, and Chihuahuan deserts. It is extremely drought-resistant and ranges from the southern deserts of North America far south into South America.

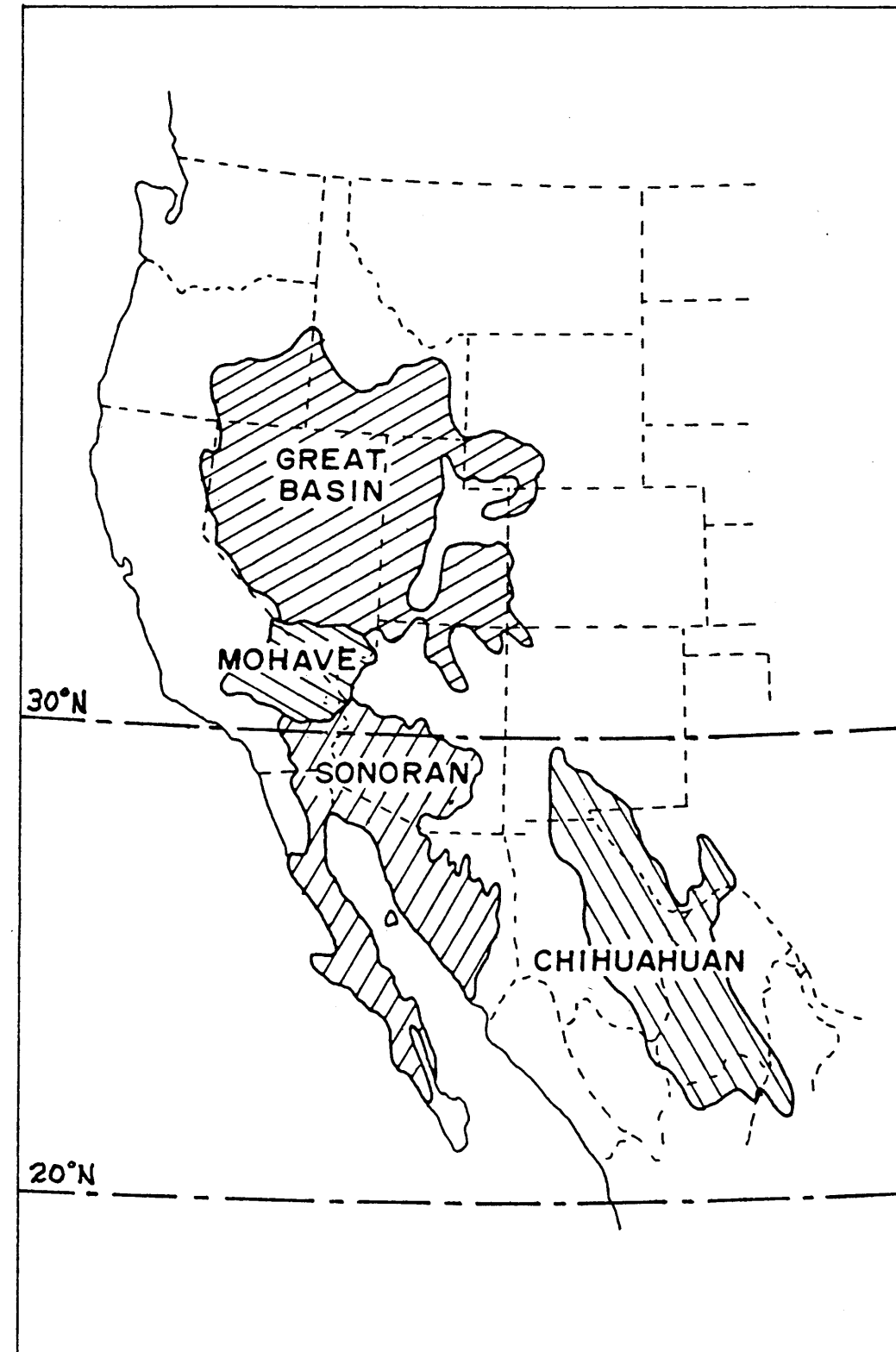
Place: The Sonoran Desert

Macroclimate

"The Sonoran Desert, for the most part, is a low, hot desert. Its area of lowest elevation (the area that surrounds the lower Colorado River) rivals Death Valley as the hottest and driest place in North America. Summer highs frequently exceed 120°F, and surface temperatures often approach 200°F....Annual rainfall in this subregion [Arizona Upland, Paloverde-Cacti Desert] of the Sonoran Desert averages less than three inches, and there have been periods of more than two years in which there was no rain at all!... The common denominator of all deserts is, of course, a lack of moisture. Availability of water is modified by seasonal fluctuations in duration and intensity of rains, rate of evaporation, and the nature of the soil. Deserts are classified in two main categories based on the conditions that create them: Horse Latitudes and rain shadow." [4]

Horse Latitudes:

Between 15° and 35° North and South latitude, respectively referred to as the Tropics of Cancer and Capricorn, unique weather patterns are responsible for the fact that the world's deserts are situated between these two boundaries. Warm, moisture laden air rising from the oceans near the equator cool as they rise, and the moisture condenses as rain. The cooled, low moisture air moves towards the poles and downward from the upper atmosphere. As the cool air moves downward in the subtropical regions of the Horse Latitudes it warms, thereby increasing its ability to hold moisture. Generally then, the cool dry air, as it warms, acts as a sponge to any available moisture and causes the aridity common to the land masses straddling the Tropics of Cancer and Capricorn. The areas to the north and south of these latitudes receive the absorbed moisture as it rises, moves poleward, and drops its moisture again.



Rain Shadow:

At a smaller scale, a similar process occurs when moisture laden air masses encounter a mountain land mass. The air rises against the windward flank and is cooled to the point of losing its moisture. The cool, dry air tips the crest and flows down the leeward flank of the mountain, warming as it descends. The evaporative(moisture absorbing) quality of the air is increased as a result. As is often the case with higher mountain ranges, one side is heavily forested and well watered while the other side is desert or steppe.

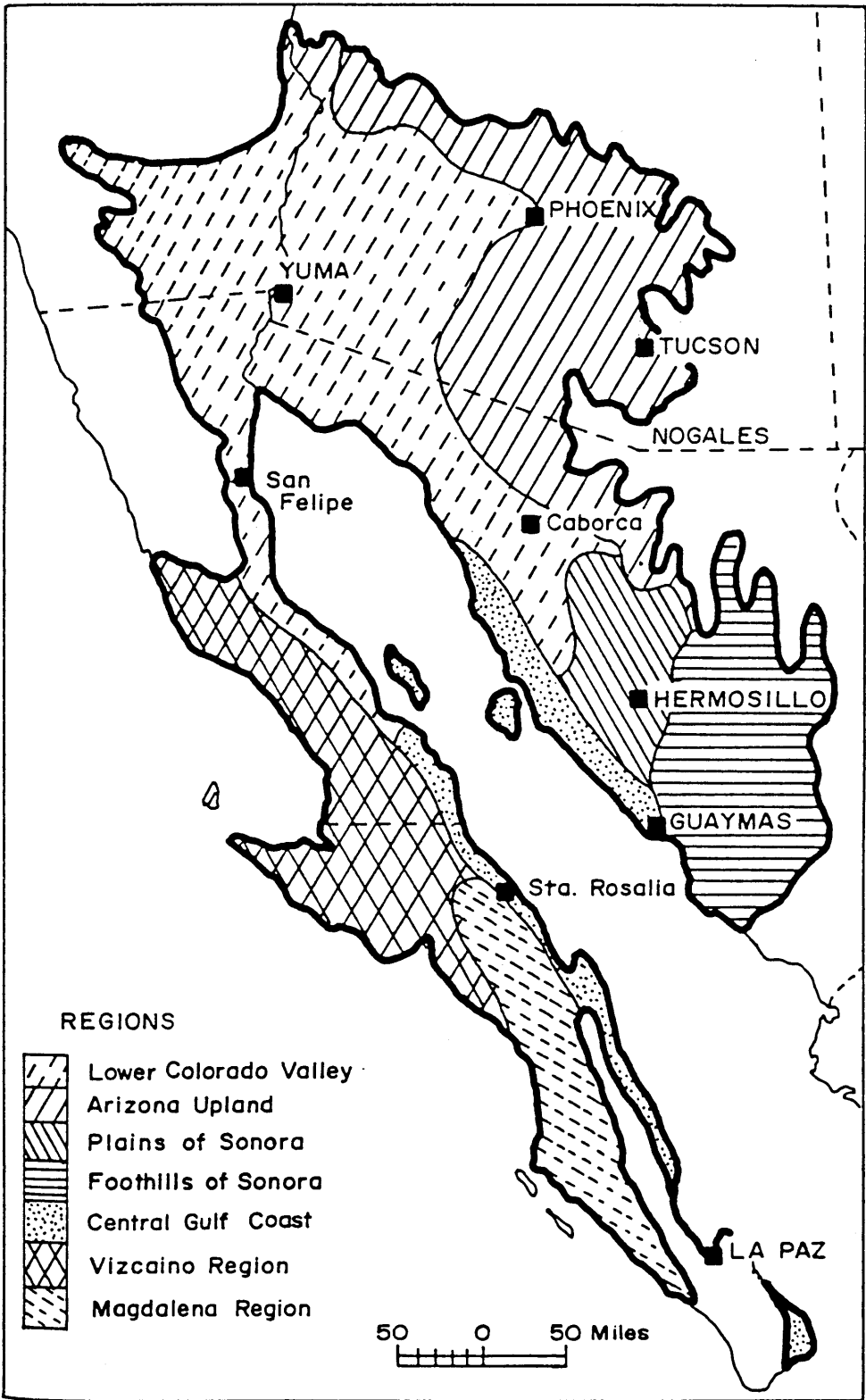
Other Factors:

A third factor that can be responsible for the creation of a desert is the overall remoteness of a region from a moisture source; a region deep in the heart of a continent may be far enough from the source that the air masses lose all their moisture before arriving at the area. A few deserts are the result of cold, polar ocean currents moving towards the equator. If the current washes close to shore of a continent, then, in conjunction with on-shore winds cooled considerably by the water, the air mass moving onto the land is very low in moisture and may only produce fog but no rain. [5]

Interestingly, the Sonoran Desert is the result of all four desert producing characteristics. At its northern reaches it is the combination of rain shadow effects of the Sierra Nevada in California and the Rockies in New Mexico, plus the prevailing high pressure zone of downward moving air that creates desert conditions. Whereas further south on the west coast of Mexico, high aridity is created from rain shadow from the mountains in Baja California to the west and the crests of the Sierra Madre to the east, plus cool ocean currents looping upward into the Sea of Cortez from the Pacific. Tucson has the advantage of lying at the edge of these effects and benefits from sporadic penetrations of rain showers from the Gulf of Mexico during the summer and cyclonic storms originating in the Pacific during the winter. Tucson is unique in that it enjoys a bi-annual rain cycle, which consequently encourages the great variety and lushness of plant communities common to the Arizona-Upland division.

Additional Subjects:

Cloudless skies and clean, dry air at night are conducive to rapid cooling of the Earth's surface: the day's warmth being radiated directly to the sky.

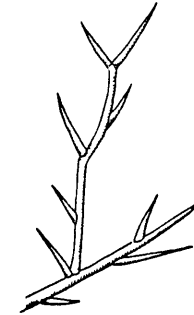
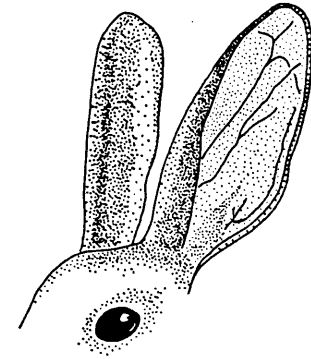
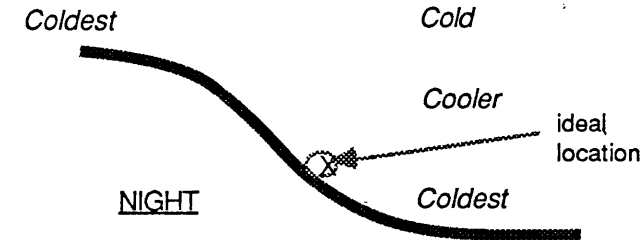


Consequently, the ground surface is the level that experiences the most extreme temperature swings. Additionally, the effects of slope and altitude will alter the microclimatic variation of moisture and temperature. Cool air is heavier than warm and will seek its lowest level; the behavior of a liquid is a good model for describing the behavior of cool and warm air movements. Warm air radiated to the sky at higher elevations cools and moves downslope, generally following water drainage troughs. If there is a leveling of the trough then the cool air ponds at the bottom. The cross-section of the trough would show that the coolest layer is at the bottom and that the rim is the warmest. The ideal location is just above the cool air pond. A similar cross section can represent the layering of air masses where the terrain is relatively flat. Again, the ground layer is the coolest (or the hottest) with successive gradations in temperature as you move upwards. Below the surface there is a gradual warming (or cooling) till the zone of constant temperature is reached several feet below the surface.

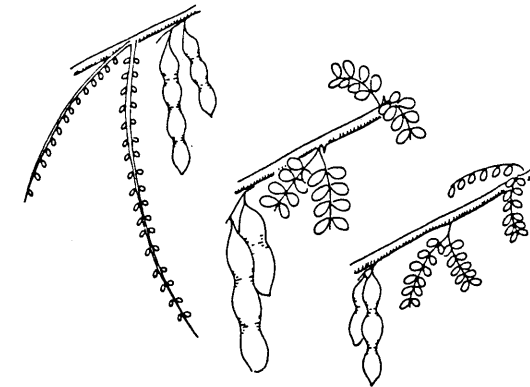
Microclimate:

The features that are important to emphasize for this paper relate to the local effects of diurnal change. Perhaps the two most important variables that any life in the desert must adapt to are the lack of moisture and the extreme difference between night and day. The aridity is an obvious life-threatening factor, and many plants and animals have evolved patterns of activity, and/or physiological adaptations that are a direct response to the difference between night and day conditions. I want to briefly describe a few of these adaptations and the environmental conditions that are responsible for their expression. The intent is to isolate the examples which have the most direct influence on architectural design.

Plants, as stationary objects, have no ability to flee the extreme sun and therefore have developed many ways to offset the effects of sun and aridity. One mechanism is to prevent loss of valuable water to the air by having a virtually impermeable skin, waxy and thick like several species of cacti exhibit. Loss of moisture occurs most often by transpiration through the leaves, which can be reduced by evolving smaller thicker leaves like the creosote bush or the palo verde tree. Other plants have a way of tracking the sun when the season is hot, and always orient their leaves parallel to the sun's rays to reduce their total exposure. Some plants, like several animal species, wait till the cool hours of the late evening to carry out important tasks.



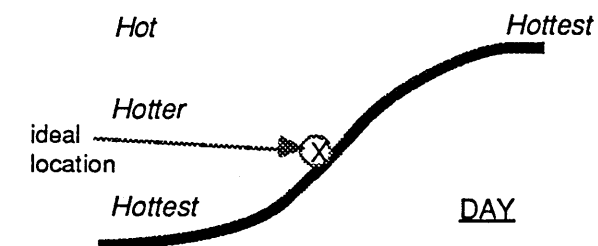
Junco, all spine, or crucifixion thorn, *Koeberlinia spinosa*, is a spreading shrub 3 to 4½ feet in height with rigid, spine-pointed branches. This species is essentially without functional leaves; as an adaptation to aridity, the yellow-green branches have taken over the normal leaf functions.



Palo verde (green stick) trees have trunks and branches, green with chlorophyll, which carry on most of the photosynthesis for the plant. The small leaves are produced only when moisture conditions are favorable. The Mexican palo verde, *Parkinsonia aculeata* (left), is distinguished by its long narrow "streamers." The blue palo verde, *Cercidium floridum* (center), is a tree of the washes in the Sonoran Desert. The foothill palo verde, *Cercidium microphyllum* (right), is the species found on the bajadas and foothills in the same desert. *Parkinsonia* grows to approximately 45 to 50 feet; *Cercidium floridum* to about 30 feet; *Cercidium microphyllum* to about 26 feet.



The saguaro cactus, *Cereus giganteus*.



Animals have mobility and have developed a variety of behavioral traits that help them adapt to the desert. Some animals do all their foraging for food in the night and sleep in near hibernation below ground during the day. Others situate their bodies relative to the sun for either warming or cooling. Jackrabbits use a combination of behavioral and physiological adaptations to remain cool; their large ears are saturated with blood vessels and by sitting under the canopy of a bush, orienting their ears to the direction of the slightest of breezes(often caused by a small chimney effect of the bush) they can cool internal organs like a radiator cools an engine. The surface temperature of the first several inches of air at and above the ground is often 50-75°F higher than the ambient temperature, therefore many animals will climb up into trees or bushes to escape the heat close to the ground. [6]

Design Correlations:

environment	design inferences
impermeable skin	insulation
thick skin	masonry mass
small leaves	reduced surface and openings
orientation and form	change shape to reduce exposure
	variable secondary
go underground	underground construction and earth-berming
timing of CO ₂ uptake and photosynthesis	variable openings and timing with diurnal cycle
openings under leaves or under hairs	screens and overhangs
variable activity cycles	seasonal use, daily use patterns
move to higher or lower level	daily use, seasonal use patterns
radiator surface	rockbed heating and cooling; raised, reflective roofing
respiratory cooling	evaporative cooling

Inferring an Automated Setting

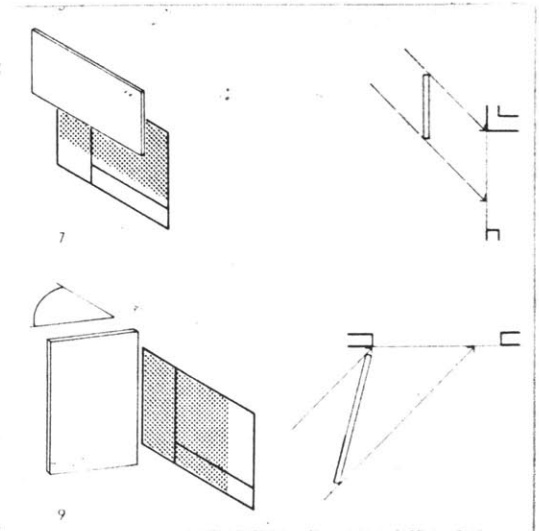
"First, we want to establish the idea that a computer language is not just a way of getting a computer to perform operations but rather that it is a novel formal medium for expressing ideas about methodology. These programs must be written for people to read, and only incidentally for machines to execute." [7]

I should be clear about my intentions with the sketch problem; in a more rigorous design process there would be several iterations of the particular problem at hand, and generally some influence of the design on work that was done up to that point: "further appreciations". However, due to the research work necessary to allow the design process to take root in the automated system, I had to compromise my time involved with the design only. The sketch problem had to satisfy two problems: the design of a secondary system and the problem of defining components of the referencing process. I believe that by being , or trying to be, conscious of how you design, and how each decision influences other steps in the process, is challenging and slow. The pace quickens with experience since rules(or decisions) that often evolve implicitly become recognizable through the practice of reflective thinking, problem after problem.

The sketch problem, given that I had made a preliminary pass at the gross sections and plans during the studio, was an attempt to develop an appropriate secondary system of closure. The general parameters of closure were a somewhat contradictory response to sunlight(small opening size to reduce exposure, but also openness to the landscape), closure that opens and closes for ventilation, and a system that worked coherently with the primary system. By coherence, I intended to use the secondary system to play off of and reduce the apparent heaviness of the concrete and masonry primary.

There were several possible resolution scenarios, some of which were actually implemented in the initial studio design phase, while others are part of general experience that I "carry" with me. The scenarios include:

- 1) the memory of past references and experiences
- 2) the research into issues of desert design: constraints and rules-of-thumb



- 3) the random search for analogies without specific knowledge of a reference, but guided by inarticulate information from the design
- 4) the research of and reference to traditional and contemporary examples

Any combination of the above scenarios informs the designer, then re-forms the design which re-informs the designer of what aspects of the problem may need further referencing.

The Expert Shell:

The limitation of using the expert shell as a database for images places more specific requirements on the images actually used by the expert shell. The "keyimages" are few, consequently, the definition and selection criteria for those images are reflective of an understanding of the nature of references in general, and the specific quality of knowledge attached to the image. The knowledge ideally represents both general and detailed rules of transformation. References are a *"means through which knowledge is collected, passed on, created --- knowledge of design transformations which have worked, of those which have failed --- knowledge of particular issues, of particular ways to divide the problem, to limit the scope of investigation --- knowledge of other important instances: recognized, but articulate, and less understood."* [8] However, compiled knowledge described in the expert shell will never be universally or indefinitely acceptable without constant revision. This fact is the most critical fault of using a compiled program. The relational database link with, and the script output from, the expert shell are methods that offset the faults of compiled information. Neither method affects the compiled knowledge source, but each performs as an additional layer by which judgements may be made; the architect can create knowledgeable links in the database or evaluate steps in the inference process by analyzing the script record for each interactive session.

Reference Elements:

A) Prior Knowledge

- from experience... [qualitative preferences]
- thick walls
 - courtyards

- small openings
- overhangs(verandahs, awnings, pergolas)
- corrugated metal(roofs, siding, etc.)
- exemplary projects

sources for: images and rules-of-thumb

B) Formal Research

- traditional examples
- lessons from Nature
- design guidelines

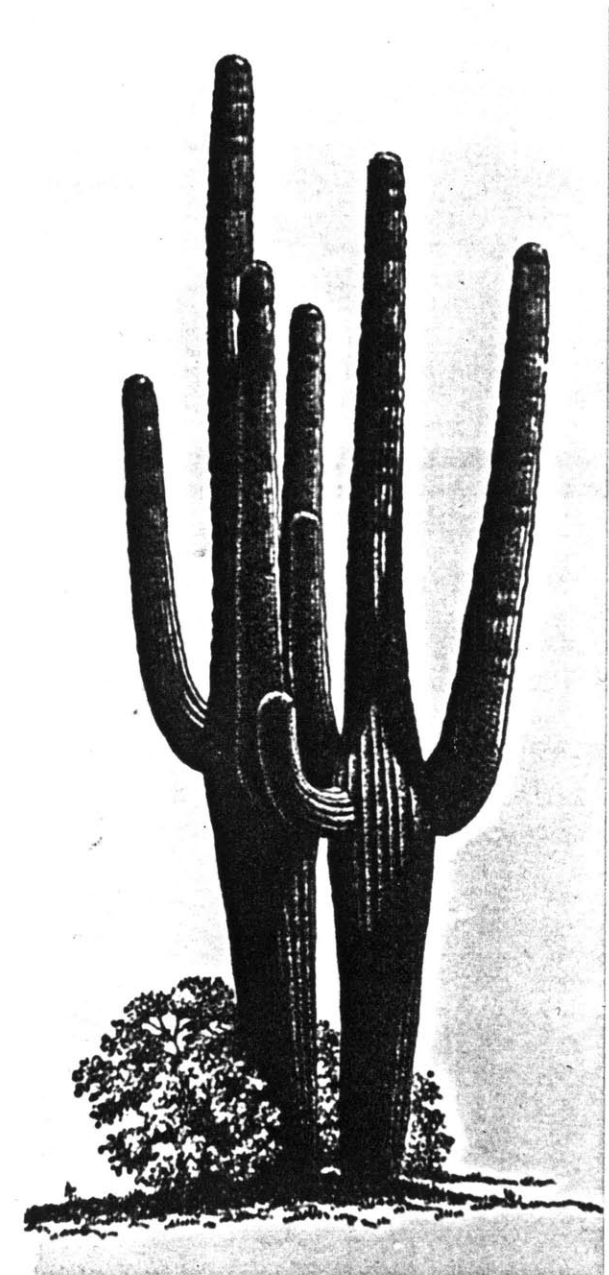
sources for: images and rules-of-thumb

environmental rules-of-thumb:

insulation
masonry mass
reduced surface and openings
orientation and form
variable secondary
underground construction and earth-berming
variable openings and timing with diurnal cycle
screens and overhangs
seasonal use, daily use patterns
rockbed heating and cooling; raised, reflective roofing
evaporative cooling

architectural rules-of-thumb:

from guidelines...
under roof drafts
positioning windows to look onto shaded courtyards reduces heat gain and glare
external areas designed to promote convection breezes for cooling outside
spaces and inside rooms
window lattices filter and lower light contrasts
exposed walls are massive and with few openings to reduce gain
cluster developments to maximize shaded areas



adjust orientation to capture beneficial winds, minimize glare, and
 reduce solar gain(both direct and reflected)
 harvest water to mitigate general aridity and to effect cooling by
 evaporation
 concentrate open courtyards for ventilation and shade
 provide shaded, protected path networks
 reduce large, open unshaded areas

The architectural rules-of-thumb evolved primarily from formal research of guidelines, and from the environmental clues. My first referencing query began with looking at images having to do with the general category of windows and edge conditions. This visual journey included only the terrain of literature I had located or specifically remembered. I made an assumption early on that using references is knowing some issues, only then can formal descriptions begin to define themselves. Logically this assumption seems backwards because the description that is used to find images in the proposed automated system is the starting point of interaction rather than a result. The description syntax in the classification scheme is not what might be called natural language, so a bit of discovery was necessary to specify a vocabulary for the descriptions. Although the sketch problem required discovering a vocabulary for the automated system, that experience could be incorporated into the knowledge-base to help teach about and define the classification scheme. The scheme does not exist independently, but is one of several choices in a simple query interface, which would be a more recognizable lead-in to actions within the expert shell. The purpose was to incorporate the possible "actions", the menu of choices, into an interaction model that followed a familiar 'verb, subject, modifier' sequence.

VISUAL DIALOG

I began the sketch problem by defining some diagrammatic guidelines that would start a pattern I could match with subsequent references. The first sketches were primarily for framing the design idea into a focus having to do with the spatial relationship of the secondary system to the thick masonry primary, and the vocabulary of materials to be used. While I was sketching and using the

references I generated several rules that helped infer values to 'description' attributes.

If design Inference = masonry mass

then design Idea = exposed massive walls

design Idea = small openings

design Idea = deep reveals.

explain = Pattern 223 (*A Pattern Language: C. Alexander*)

scope = primary secondary.

behavior = primary system.

property = masonry thermal damping.

Thick walls allowed deep reveals, which satisfied one of my qualitative preferences. However, the need for a high quality interior light conflicted with the requirement that openings be small to reduce gain and glare.

If design Idea = small openings

then light = minimum.

If light = minimum

then recommend = open wall to light.

adjust = secondary system to modulate light , solar gain.

If adjust = secondary system to modulate light, solar gain

then behavior = secondary system & light modulation.

scope = primary secondary.

summary description:

(scope: primary secondary (behavior: primary system & secondary system & light modulation (property: masonry thermal damping)))

To design the secondary system required the referencing of concrete examples. I had a basic understanding of principles behind the use of overhangs and light shelves, but my knowledge of variations was weak. The references I found on the architecture of Yemen were excellent, since the "system" that is traditional to buildings in Yemen is consistent, clear, and flexible to interpretation. My observations showed that windows are a combination of fixed and operable elements arranged in relation to a three part ordering scheme. The purpose of the three ordering elements could be organized as follows:

1) Ventilation --- there are small openings up high, and generally

separate from , and next to the primary penetration of the masonry to allow venting of warm air.

2) The topmost opening is always a fixed closure and functions to allow light into the space only.

3) The lower opening, the largest of the three, has many operable elements. The opening and closing of various shutters serves several options: allows unobstructed views out the window, adjusts the intensity of light, aids in adjusting ventilation, and allows complete closure. Some systems are simple, only consisting of a pair of wooden shutters at the outer edge of the opening, or some are complex, consisting of up to three pairs of shutters and screens. Sometimes there is a light shelf above this large opening to help block intense, midday sun. The general use of the larger, more variable opening is related to the need for people to view their surroundings and have more subtle control of environmental effects while the space is used. [9]

If openings = two

then use = for light & for view , ventilation

If use = for light

then zone = top.

else If use = for view, ventilation

then zone = bottom.

explain = "In Yemen the floor is used for social gathering; everyone sits

on cushions arranged along the perimeter of the walls, and generally next to the light, view and breeze."

If zone = bottom

then closure = operable.

behavior = ventilation.

If zone = top

then closure = fixed.

sill placement = shoulder to head height.

behavior = light modulation.

If closure = fixed and

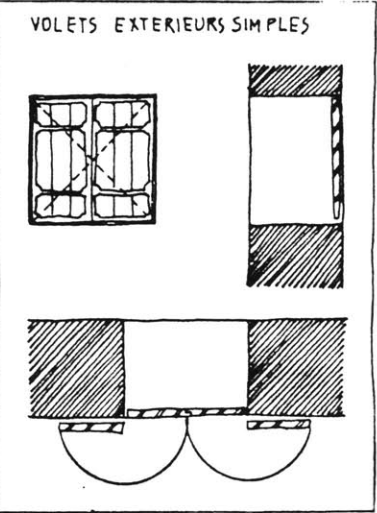
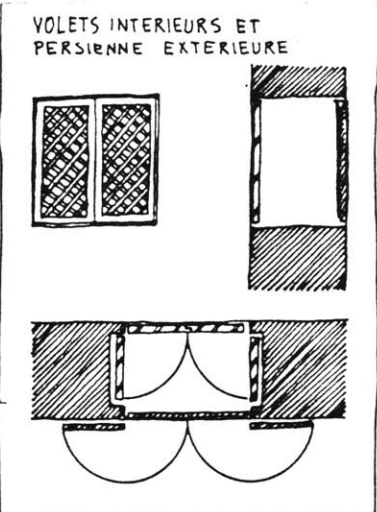
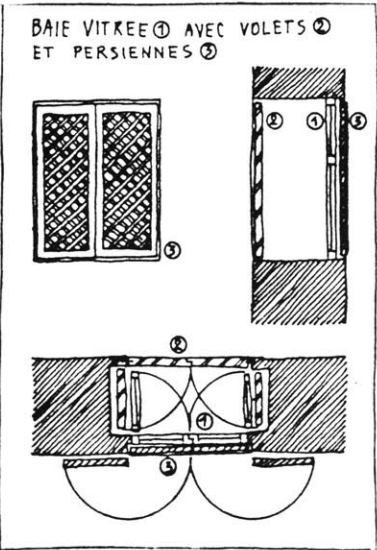
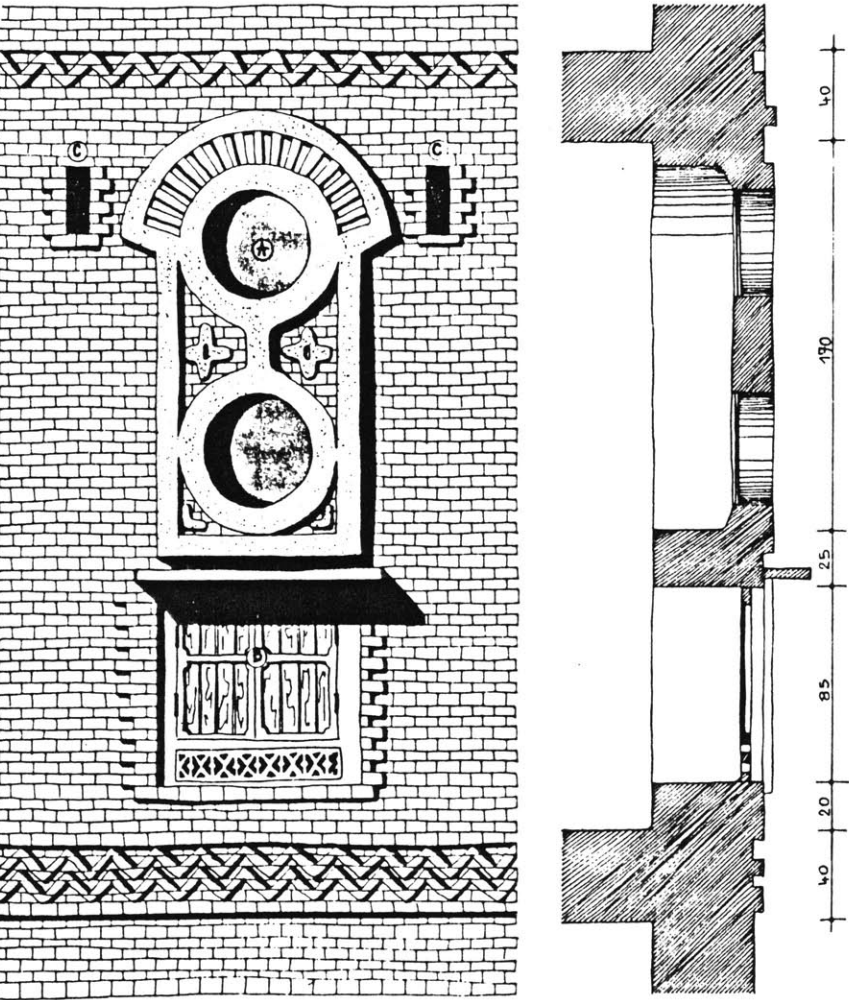
elements = multiple

then place = staggered.

explain = "When there are multiple closure elements they are

SANA'A. EXEMPLE D'UNE FENÊTRE ANCIENNE

TYPE DE L'OUVERTURE	CARACTERISTIQUE PRINCIPALE	FONCTION
Ⓐ FENÊTRE ORNEMENTALE	ALBATRE	LUMIERE
Ⓑ FENÊTRE INFÉRIEURE	VOLETS DE BOIS	VUE
Ⓒ FENÊTRE DE VENTILATION	MEURTRIÈRE OU VOLLET BOIS	VENTILATION



staggered vertically in relation to each other. The top most element is located close to the outer edge with the bottom most element generally located at mid-wall. This pattern increases the depth of reveal thereby decreasing glare; see Pattern 223: "Deep Reveals", (*A Pattern Language*). The general effect is that light is stronger in intensity high in the space."

If use = for light & for view, ventilation

then scope = env_controls.

property = convective cooling.

vent placement = Inlet at bottom, outlet at top.

If vent placement = Inlet at bottom

then closure = operable & variable.

property = operable closure & variable closure.

If zone = bottom and

closure = operable | (or) variable

then place = outside edge of opening.

explain = "This is true if there is only a shutter and a screen or a shutter/screen combo. If there are three closure elements in the opening, one is at the innermost edge and generally functions as privacy closure."

summary description:

(scope: env_controls (behavior: ventilation & light modulation

(property: convective cooling & operable closure & variable closure)))

The second round of referencing built off the first sequence, but was more focused to deal with the design of space. I felt that details of the wall section would be clarified as I attempted to understand the relationship of the secondary system to issues of edge definition and spatial definition. The plan organization maintains a continuity of spaces with some alternation of enclosure along its axis. That formal organization was dependent on different qualities of light to reinforce "place", direction, and working comfort. The goal was to eventually develop a general derivative that could be used as a system throughout the entire project. The references I turned to included the works of Frank Lloyd Wright, particularly his work between 1937 and 1941, when he developed his Usonian prototype for domestic construction. I was specifically interested in one

project, Taliesin West, which was being built during this period. Taliesin West offered an example of desert design and, since Wright was designing his own home and foundation studios, I assumed his attention to space and nature would be more rich and varied than might otherwise be expected. Wright was not the definitive master of desert houses, but he was sensitive to the environment and for the moment his works are well documented and accessible.

Wright used similar design decisions throughout his work, amplifying or subduing elements as warranted by the climate, site, and formal intentions. A recurring pattern that I noticed in his work was the use of overhangs/cantilevers and high, horizontal window bands. Wright was able to create a wide variety of spacial qualities by varying the relationship between these two elements. Taliesin West is a rich example of the use of this pattern. Overhangs and cantilevers are exaggerated to act as sunblocks where walls open up to the outside, while horizontal window bands are generally located up high with small overhangs where the wall is intended to be more solid.

If wall orientation = south

then design idea = reduced opening | deep overhang.

explain = "There is often an alternation of the two in the same space: one wall with light up high (small, reduced openings), and a neighboring wall with virtually no mass--- all glass."

If wall = solid

then place = high.

overhang = shallow.

explain = "By placing the daylight source up high, direct and reflected light washes from above, lighting the ceiling and reducing its apparent closure. The space, although relatively contained, is relieved of its enclosure and darkness."

else If wall = open

then overhang = deep.

behavior = relationship between primary and secondary.

explain = "Opening the wall allows the interior and exterior spaces to become more continuous. The deep overhang carries the interior definition to the outside while simultaneously defining a volume of space unique to the outside: reciprocity and interpenetration of spatial volumes."

223 DEEP REVEALS



If overhang = shallow

then alternate = operable extensions.

explain = "Adjustable louvers or shades give more freedom to the dimension of the opening by providing another means of modulating the amount of direct sun or glare."

If overhang = deep

then alternate = reduced structure.

property = wooden screens.

explain = "Instead of making the overhang totally solid, variations include making the overhang a trellis, thereby screening light rather than blocking it. The filtering nature of the trellis can be further diversified if used in conjunction with vegetation. Seasonal variation of deciduous vines will modulate light when it is most beneficial, and also add moisture to the air for evaporative cooling. See Pattern 238: "Filtered Light", (*A Pattern Language*). "

summary description:

(behavior: relationship between primary and secondary (property: wooden screens))

The inference process helped to develop the descriptors that were used both in the automated classification and in keyimage descriptions. The interesting thing about the process is that the rules are reversed within the compiled knowledge-base. The above rules were responsible for determining the description vocabulary, which is used in the classification scheme. However, in the knowledge-base for an image the rule logic was reversed:

ie., if property = wooden screens and alternate = reduced overhang
then overhang = deep.

The "explanation" of the rule can be called within the expert shell by using a 'why' command or by explicitly asking for the justification of an inference. The level of referencing represented in the above groups of rules were crude, but they did help initiate the first iteration of the design. Once the design was in progress I discovered new issues for focusing further referencing.

If lower sill = 2' - 3' above floor and opening = singular

then overhang length equals height of opening.

If design idea = three part ordering scheme

then lower sill height = 2' - 3'.

mid-sill height = 4.5' - 5.5'.

overhang length = upper window opening height.

property = 2 and 3 part ordering.

property = edge definition.

If opening = south & full height

then overhang length = window height | half of window height with use of vegetative screen.

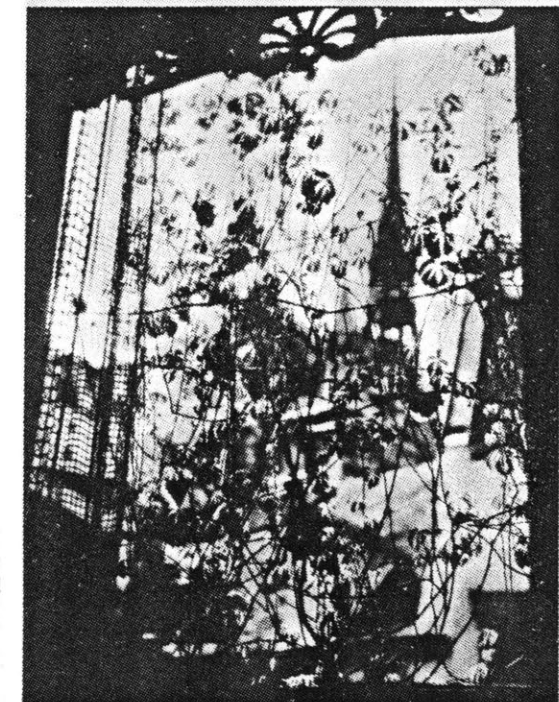
scope = primary secondary & env_controls.

summary description:

(scope: primary secondary & env_controls (property: edge definition & 2 and 3 part ordering))

All the rules so far are representative of the domain of knowledge that assisted me in the design, but they only represent the fruits of a "first pass". The design passed from general to specific quite rapidly. The next iteration would have to include a global re-evaluation or top down re-framing of the problem. The design would then be substantial enough in details and principles to be decomposed into a general diagram for new sets of rules, which ideally address the architectural inferences as a more generative system capable of being used on many buildings. This level of knowledge conveys the most valuable information, but is dependent on having been designed with expert knowledge upfront, prior to compiling rules in the expert shell. Any learning that occurs thereafter is absorbed by the relational techniques in the database, or by a re-authoring of the system. One strategy is to design a knowledge-base that has been developed through interaction with experts who have extensive experience in the problem area; the expert system starts out from the beginning as a fully operable, concise system. Another strategy is to develop a prototype that tries to take advantage of inherent strengths and couple them with other tools that compensate for the expert system's weaknesses. Finally, the other strategy is to view the design of the system as a dynamic, evolving entity that is part of the toolkit that a designer may use from day to day, and change to respond to new conditions. This last strategy requires that the user be an author also.

238 FILTERED LIGHT*



The construction and maintenance of a referencing expert requires that the designer be familiar enough with the software to be an author who is capable of updating, or even re-writing the knowledge-base. The original keyimages are certainly open to re-interpretation or replacement over time. Rules about the images and the relationships between them are also quite malleable over time, especially as an individual's design experience evolves. The limitations that an automated system like I have proposed stem directly from its inability to learn. I have composed an organization that responds as best as possible, but ultimately, the record of learning is very dependent on a wider context than just the expert shell and knowledge base. I have been careful to place the actions of the automated system in a context whereby it functions, or can function with a host of other tools. Creating interdependencies was a means to compensate for the inadequacies of any one tool. The other goal was to root the work into a larger array of related research, all under the umbrella of "visual information systems".

I realize that the actual knowledge representation accomplished by the expert shell is rudimentary; yet the fault is not attributable to the capabilities of the software. I spent a significant amount of time developing the framework of connections, each path generated being a necessary branch to achieve a higher goal. The goal was, and still is, the implementation of a network of inferences capable of giving knowledgeable advice. The first iteration of the knowledge-base, like the first iteration of the sketch problem, has been successful in generating the principles for a vigorous system. I think there is always a desire to create software that works entirely as it's supposed to, under every possible condition; however, a research environment promotes prototyping of schematics. If successful, prototypes can become general utilities useable by a larger audience, but I have chosen a slightly different strategy by realizing the inherent frustrations of authoring a tool that can be used by anyone. Issues of interface and error prevention soon overwhelm the actual content when developing a general utility. My intention throughout was to develop a personal tool amenable to my habits and my growth as a designer. I suppose a prerequisite of the knowledge-base is that the potential user must also be a potential author. Implied in this strategy is the fact that any learning can be applied more fluidly and accurately towards the next evolution of the system. The continuity of the design process is thereby preserved.

The automated environments currently under development in architecture are just beginning to skim the surface of visual thinking and design. To suggest that computers have all the answers is an accepted fallacy, but there is practical evidence that a substantial amount of "answering" can be offered knowledgeably by computers. By keeping in perspective the relation of human thinking to integrity of process, the design of automated systems may help us perceive new ways of understanding our own thinking. In the end...

"The program, the hardware, indeed the output of the computer is always limited: design is a human response and as such relies on the widest possible frame of reference. Every sight we see, every sound we hear, every building we visit has or could have an influence on our next design....[computers] will always be tools, very powerful tools, but at some stage in the design process their influence will stop." [10]

REFERENCES

- [1] Graf, Douglas. "Diagrams", *Perspecta*. 20 (1983). Cambridge, MA: MIT Press; p. 43.
- [2] Chafee, Judith with Brock, Diana (College of Architecture, University of Arizona, Tucson). "Rancho de los Amigos," *Studio Brief: MIT Department of Architecture*. Cambridge, MA: MIT, Fall 1986. p.3.
- [3] Herbert, Frank. *Dune*. New York: Berkeley Publ. Corp., 1965; p. 380.
- [4] op cit., Chafee and Brock. p.43.
- [5] ibid.
- [6] Larson, Peggy. *A Sierra Club Naturalist's Guide, The Deserts of the Southwest*. San Francisco: Sierra Club Books, 1977.
- [7] Abelson, Harold and Sussman, Gerald J. *Structure and Interpretation of Computer Programs*. Cambridge, MA: MIT Press, 1985; p. xv.
- [8] Anderson, James. "Lecture Notes: Relational Formal Language(4.182)." Cambridge, MA: MIT. 1987; pp. 3,4.
- [9] Hirschi, Suzanne and Max. *L'Architecture au Yemen du Nord*. Paris: Architectures, Berger-Levrault, 1983.
- [10] Hansen, Neil L. R. and Radford, Antony. "Living on the Edge: a grammar for some country houses by Glenn Murcutt," *Architecture Australia*. 5 (July 1986); p. 66.

APPENDIX

```
\04.23.87
\A.Bennett
\Thesis Experiments: Query Interface(q_new3)
\Kes production-rule system(KES.ps)

\*****
text:

{banner:"      ---- ARID LANDS DESIGN ADVISOR ----",
  "THIS MODULE IS SPECIFICALLY FOR SECONDARY SYSTEM DESIGN",
  "[MASSACHUSETTS INSTITUTE OF TECHNOLOGY; Department of ",
  "Architecture and Urban Planning Computer Resources Lab.]",
  "Author:A.BENNETT",
  "THIS IS A KNOWLEDGE-BASED VISUAL REFERENCING SYSTEM",
  "WHICH IS USED AS A DESIGN TOOL FOR ARCHITECTURAL",
  "APPLICATIONS SPECIFIC TO DESIGN IN ARID LANDS.",
  "THE SYSTEM IS COMPOSED OF TWO PARTS:",
  "1)AN INTERFACE SYSTEM FOR CHANNELING DESIGN QUERIES",
  "TO A VISUAL KNOWLEDGE-BASE and",
  "2)THE VISUAL KNOWLEDGE-BASE WHICH IS WHERE THE",
  "INTERACTIVE DISPLAY OF SELECTED IMAGES AND",
  "KNOWLEDGE SOURCES IS DEMONSTRATED.",
  "PART ONE OF THE REFERENCING SYSTEM ALSO MANAGES MORE",
  "IN-DEPTH DATABASE APPLICATIONS THAT ARE TIED DIRECTLY",
  "TO THE 'LIBRARY' OF AVAILABLE IMAGES. IMAGE PROCESSING",
  "AND MANAGEMENT OF VISUAL RESOURCES ARE TASKS AVAILABLE",
  "THROUGH THIS OTHER CHANNEL OF THE QUERY INTERFACE."}

%
\*****
patterns:

descript:
VERB [alternatives: "show", "Show", "SHOW", "change", "Change",
  "CHANGE", "give", "Give", "GIVE", "ask", "Ask", "ASK"],
delimiter1 [spans: " "] [atmost: 1],
OBJECT [alternatives: "image", "Image", "IMAGE", "array",
  "Array", "ARRAY", "reference", "Reference",
  "REFERENCE"],
delimiter2 [spans: " "] [atmost: 1],
MODIFIER [alternatives: "by scope", "by Scope", "by SCOPE",
  "by behavior", "by Behavior", "by BEHAVIOR",
  "by property", "by Property", "by PROPERTY",
  "by specification", "by Specification",
  "by SPECIFICATION", "by spec", "by specs", "by Spec",
  "by SPEC", "by Specs", "by SPECS", "by combination",
  "by Combination", "by COMBINATION", "by combo",
  "by Combo", "by COMBO"].

%
\*****
attributes:

sentence:str
{question:"
  "If you have any questions, TYPE the command 'WHY'",
  "to get HELP.",
  "SO, WHAT ARE YOU INTERESTED IN?",
  "Type in your query using the following examples",
  "as a model of sentence building.",
  "Show image by Scope , or Show image by combo",
```

```
s", "",
"options:  verbs      objects      modifier
"          SHOW      IMAGE      (by)SCOPE",
"          ARRAY      BEHAVIOR",
"          REFERENCE  PROPERTY",
"          SPECIFICATION",
"          CHANGE or  INFORMATION (on)IMAGE ",
"          GIVE      CATEGORY(ingres)
"          GLOSSARY(ingres)

{why: "",
  "Each of the capitalized words acts as a verb in a",
  "query sentence that you are about to build.",
  "SHOW---as in 'show me' visually any of various",
  "possibilities;",
  "CHANGE---as in 'change or append, update, or",
  "re-direct' any of various possibilities;",
  "GIVE---as in 'give me' lists of information about",
  "any of various possibilities;",
  "ASK---as in 'ask questions' about any of various",
  "possibilities."}

verb:str.
object:str.
modifier:str.
response:str.
request:sql
(scope, behavior, property, spec, combo, none)
[default: none].

respond:str
{question: "", "PLEASE TYPE THE PARAMETERS FOR THE IMAGE.",
  "Separate multiple entries by a space, an & or a |,",
  "and another space."}.

descriptor:sql
(by SCOPE, by BEHAVIOR, by PROPERTY, by SPECIFICATION,
  a COMBINATION, EXIT)
{question: "",
  "Choose a category from the following list....",
  ""}.

keycom1:sql
(yes, no)
[default: no].

single:sql
(yes, no)
[default: no].

Scope:mlt
(primary secondary
```

```

{explain: "", "The primary secondary system refers to the",
 "interrelationship or separate occurrence of the two systems",
 "in a building. The definition of the specific focus as",
 "described can be chosen during the definition of the",
 "behavior parameters.", ""},
"The primary secondary system is a parameter of Scope",
"because this particular referencing model is designed to",
"deal with the issues surrounding the design of a secondary",
"system in an arid climate setting. To refer to the",
"secondary system without acknowledging its relation to the",
"nature of the primary system is generally discouraged.", ""},
env_controls
{explain: "",
"The term env_controls refers to that group of elements that",
"might be defined as functions or components dealing with",
"issues related to climate modulation, whether it be",
"wind, moisture, temperature, or light factors.", ""},
"The reason environmental controls is included in the Scope",
"is that secondary systems are generally designed to satisfy",
"the requirements of closure from the elements. If an",
"attitude of design with Nature is accepted, then the more",
"a designer is informed of means to deal with sun, wind, and",
"moisture, the more freedom they are given to design",
"variations.", ""},
formal_organization
{explain: "",
"Formal organization is those characteristics that describe",
"qualities of basic form, ie., the square and circle are",
"centralized forms, while a column or beam is a linear form.",
""},
"The formal organization is generally more appropriate for",
"issues such as building footprint, but with skill can be",
"directed to elevation organization.", ""},
Exit
{explain: "",
"This is the normal setting for the beginning of the query",
"process; in other words, when you start the slate is clear.",
""})
{question: "",
"DESCRIBE THE scope PARAMETERS OF AN IMAGE.",
"", "If you need explanations for the values of Scope,",
"then type explain and the number of the value.", ""}.

Behavior:mlt
(primary system, secondary system, thermal barrier,
ventilation, light modulation,
relationship between primary AND secondary)
{question: "DESCRIBE THE behavior PARAMETERS OF AN IMAGE.", ""}.

Property:mlt
(operable closure, variable closure, fixed closure,
variable screens, wooden screens, light shelf,
convective cooling, edge definition,
two OR three part ordering system, masonry thermal
damping, chimney effect, concrete double wall,
rigid insulation)
{question: "",
"DESCRIBE THE property PARAMETERS OF AN IMAGE.", ""}.

Specification:mlt
(Source, Source_ref, Represents, Image_type, Orientation,
Color)
{question: "",
"DESCRIBE THE specifications OF AN IMAGE.", ""}.

fire_file:sql

```

```

(one, two, three, four).

%
\*****
classes:

\\ FOR DIRECT QUERY TO AN INGRES VISUAL DATABASE

description: [default: d1]

attributes:

SCOPE:mlt
(primary secondary
 {question: "How about something to do",
  "with PRIMARY and SECONDARY systems...", ""},
env_controls
 {question: "How about anything that might",
  "refer to ENVIRONMENTAL CONTROLS working",
  "with the climate...", ""},
form_organization
 {question: "FORM ORGANIZATION is another",
  "parameter, but it is currently disabled.", ""},
none)
 {question: "",
  "DESCRIBE THE scope PARAMETERS OF AN IMAGE.",
  "",
  "If no parameters are appropriate, choose NONE."
}, ""}.

BEHAVIOR:mlt
(primary system, secondary system, thermal barrier,
ventilation, light modulation,
relationship between primary AND secondary, none)
 {question: "",
  "DESCRIBE THE behavior PARAMETERS OF AN IMAGE.",
  "",
  "If no parameters are appropriate, choose NON
E."}.

PROPERTY:mlt
(operable closure, variable closure, fixed closure,
variable screens, wooden screens, light shelf,
convective cooling, edge definition,
two OR three part ordering system, masonry thermal
damping, chimney effect, concrete double wall,
rigid insulation, none)
 {question: "",
  "DESCRIBE THE property PARAMETERS OF AN IMAGE.",
  "",
  "If no parameters are appropriate, choose NON
E."}.

SPECIFICATION:str
 [default: "none"]
 {question: "",
  "DESCRIBE THE specifications OF AN IMAGE.",
  "", "End string with a space, then ENTER.",
  "",
  ".....multiple entries are possible",
  "", "If no parameters, type a comma and a",
  "space, then ENTER."}.

%
endclass.

```



```

%
\*****
externals:

    keyimage kb:
        [program: "ki_kb"].

%
\*****
rules:

    get verb:
        if match(descript, sentence)
        then verb = extract(descript, VERB).
        endif.

    get object:
        if match(descript, sentence)
        then object = extract(descript, OBJECT).
        endif.

    get modifier:
        if match(descript, sentence)
        then modifier = extract(descript, MODIFIER).
        endif.

    ruleshow i:
        If verb = "show" or
        verb = "Show" or
        verb = "SHOW" and
        object = "image" or
        object = "Object" or
        object = "OBJECT"
        then response = "si".
        endif.

    ruleshow a:
        If verb = "show" or
        verb = "Show" or
        verb = "SHOW" and
        object = "array" or
        object = "Array" or
        object = "ARRAY"
        then response = "sa".
        endif.

    modifier1:
        if modifier = "by scope" or
        modifier = "by Scope" or
        modifier = "by SCOPE"
        then request = scope.
        endif.

    modifier2:
        if modifier = "by behavior" or
        modifier = "by Behavior" or
        modifier = "by BEHAVIOR"
        then request = behavior.
        endif.

    modifier3:
        if modifier = "by property" or
        modifier = "by Property" or
        modifier = "by PROPERTY"
        then request = property.
        endif.

```

```

modifier4:
    if modifier = "by specification" or
    modifier = "by Specification" or
    modifier = "by SPECIFICATION" or
    modifier = "by spec" or
    modifier = "by Spec" or
    modifier = "by SPEC" or
    modifier = "by specs" or
    modifier = "by Specs" or
    modifier = "by SPECS"
    then request = spec.
    endif.

```

```

modifier5:
    if modifier = "combination" or
    modifier = "Combination" or
    modifier = "COMBINATION" or
    modifier = "combo" or
    modifier = "Combo" or
    modifier = "COMBO"
    then request = combo.
    endif.

```

\\ \\ THESE RULES DEAL WITH SINGULAR ENTRIES FOR THE CLASSIFICATION

```

rule1:
    if request = scope or
    request = behavior or
    request = property or
    request = spec
    then single = yes.
    endif.

```

```

rule1a:
    if single = yes and
    status(Scope) = known
    then description:d1>SCOPE = none.
    endif.

```

```

rule1b:
    if single = yes and
    status(Behavior) = known
    then description:d1>BEHAVIOR = none.
    endif.

```

```

rule1c:
    if single = yes and
    status(Property) = known
    then description:d1>PROPERTY = none.
    endif.

```

```

rule1d:
    if single = yes and
    status(Specification) = known
    then description:d1>SPECIFICATION = "none".
    endif.

```

\\ \\ This rule is for the running of the KEYIMAGE knowledge-base(KES.HT)

```

rule_2:
    if single = yes or
    status(description:d1>SCOPE) = known or
    status(description:d1>BEHAVIOR) = known or
    status(description:d1>PROPERTY) = known or
    status(description:d1>SPECIFICATION) = known

```

```

        then keycom1 = yes.
        endif.

rule2:
    if keycom1 = yes and
    request = scope
    then fire_file = one.
    endif.

rule2a:
    if keycom1 = yes and
    request = behavior
    then fire_file = two.
    endif.

rule2b:
    if keycom1 = yes and
    request = property
    then fire_file = three.
    endif.

rule2c:
    if keycom1 = yes and
    request = spec
    then fire_file = four.
    endif.

%
\*****
actions:

display attach banner of kb.

while response # "Exit" do
    while request # combo do
        write "com.bo", request.

        if fire file = one
        then write "resp", Scope.
        endif.

        if fire file = two
        then write "resp", Behavior.
        endif.

        if fire file = three
        then write "resp", Property.
        endif.

        if fire file = four
        then write "resp", Specification.
        endif.

        if keycom1 = yes
        then message "", "",
        "-----",
        "",
        "LINKING TO KEYIMAGE KNOWLEDGE-BASE...",
        "",
        "*** PLEASE WAIT ***",
        "",
        "When the other window appears, move the",
        "cursor there to answer any queries.",
        "",

```

```

        "-----",
        "",
        run keyimage_kb.
        else message "HOW DO EXPECT ANYTHING TO RUN IF",
        "THERE'S NO INPUT TO DIRECT THE COMPUTER?".
        endif.

message "", "", "IF YOU ARE DONE WITH THE INITIAL PASS THROUGH",
        "THE KEYIMAGE KNOWLEDGE-BASE...",
        "", "Type 'c' to continue.".
break.

erase request.
erase single.
erase descriptor.
descriptor = EXIT.

endwhile.

if request = combo
then write "com.bo", request.
message "", "",
"-----",
"LINKING TO KEYIMAGE KNOWLEDGE-BASE...",
"*** PLEASE WAIT ***",
"When the other window appears, move the",
"cursor there to answer any queries.",
"-----",
"",
run keyimage_kb.
erase descriptor.
descriptor = EXIT.
endif.

eraseclass description.

if descriptor = EXIT
then erase response.
response = "Exit".
endif.
endwhile.

break.
stop.
%

```

```

\04.24.87
\A.Bennett
\Thesis Experiments: Keyimage Knowledge-Base(im5)
\Kes hypothesis-test system(KES.ht)

\*****
attributes:

request:sgl
(scope, behavior, property, spec, combo, none, *Exit).

respond:mlt
(primary secondary, env_controls, formal organization,
primary system, secondary system, thermal barrier,
ventilation, light modulation,
relationship between primary AND secondary,
operable closure, variable closure, fixed closure,
variable screens, wooden screens, light shelf,
convective cooling, edge definition,
two OR three part ordering system, masonry thermal damping,
chimney effect, concrete double wall, rigid insulation).

sc:truth
(*false).

b:truth
(*false).

pr:truth
(*false).

sp:truth
(*false).

source:mlt
(B
{question: "B = Book"},
F
{question: "F = Folio"},
P
{question: "P = Periodical"},
OR
{question: "OR = Original"},
OTH
{question: "OTH = Other"},
NONE)
[default: NONE]
{question: "", "CHOOSE ONE OF THE FOLLOWING VALUES TO",
"DESIGNATE FROM WHAT SOURCE THE IMAGE MIGHT ORIGINATE FROM...",
""}.

source_ref:str
[default: "none"]
{question: "", "PLEASE TYPE IN THE BIBLIOGRAPHIC REFERENCE", ""}.

represents:mlt
(p
{question: "p = Plan"},
s
{question: "s = Section"},
e
{question: "e = Elevation"},
pe
{question: "pe = Perspective"},

```

```

a
{question: "a = Axonometric"},
r
{question: "r = Real"},
o
{question: "o = Other"},
NONE)
[default: NONE]
{question: "", "WHAT DOES THE IMAGE REPRESENT?", ""}.

image_type:mlt
(ph
{question: "ph = Photo"},
sk
{question: "sk = Sketch"},
d
{question: "2 = Diagram"},
ort
{question: "ort = Orthographic"},
fv
{question: "fv = Film/Video"},
o
{question: "o = Other"},
NONE)
[default: NONE]
{question: "", "WHAT TYPE OF IMAGE?", ""}.

orientation:mlt
(vertical, horizontal, NONE)
[default: NONE]
{question: "", "WHAT IS THE PRIMARY ORIENTATION OF THE IMAGE?",
""}.

color:int
{question: "0 = Black & White, 1 = Color",
"2 = Brown, 3 = Blue"}.

Scope:mlt
(primary secondary
{explain: "", "The primary secondary system refers to the",
"interrelationship or separate occurrence of the two systems",
"in a building. The definition of the specific focus as",
"described can be chosen during the definition of the",
"behavior parameters.", ""},
"The primary secondary system is a parameter of Scope",
"because this particular referencing model is designed to",
"deal with the issues surrounding the design of a secondary",
"system in an arid climate setting. To refer to the",
"secondary system without acknowledging its relation to the",
"nature of the primary system is generally discouraged.", ""},
env_controls
{explain: "",
"The term env_controls refers to that group of elements that",
"might be defined as functions or components dealing with",
"issues related to climate modulation, whether it be",
"wind, moisture, temperature, or light factors.", ""},
"The reason environmental controls is included in the Scope",
"is that secondary systems are generally designed to satisfy",
"the requirements of closure from the elements. If an",
"attitude of design with Nature is accepted, then the more",
"a designer is informed of means to deal with sun, wind, and",
"moisture, the more freedom they are given to design",
"variations.", ""},
formal organization
{explain: "",

```



```

"Formal organization is those characteristics that describe",
"qualities of basic form, ie., the square and circle are",
"centralized forms, while a column or beam is a linear form.",
"",
"The formal organization is generally more appropriate for",
"issues such as building footprint, but with skill can be",
"directed to elevation organization.", "",
*NONE
{explain: "",
"This is the normal setting for the beginning of the query",
"process; in other words, when you start the slate is clear.",
""})
{question: "",
"DESCRIBE THE scope PARAMETERS OF AN IMAGE.",
"", "If you need explanations for the values of Scope,",
"then type explain and the number of the value.", ""}.

Behavior:mlt
(primary system, secondary system, thermal barrier,
ventilation, light modulation,
relationship between primary AND secondary, *NONE)
{question: "",

"DESCRIBE THE behavior PARAMETERS OF AN IMAGE.",
""}.

\\NEED EXPLAIN

Property:mlt
(operable closure, variable closure, fixed closure,
variable screens, wooden screens, light shelf,
convective cooling, edge definition,
two OR three part ordering system, masonry thermal damping,
chimney effect, concrete double wall, rigid insulation,
*NONE)
{question: "",
"DESCRIBE THE property PARAMETERS OF AN IMAGE.",
""}.

\\NEED EXPLAIN

Specs:mlt
(Source, Source_ref, Represents, Image_type, Orientation,
Color, *NONE)
{question: "",
"DESCRIBE THE specifications OF AN IMAGE.",
""}.

\\NEED EXPLAIN

name:sgl
(window system, wall system, bldg_sect, NONE)
{question: "",
"DESCRIBE a general category FOR INDEXING AN IMAGE.",
""}.

\\NEED EXPLAIN

query:sgl
(Continue
{question: "I've had second thoughts and would like to e
xpand the description."}
{explain: "", "Choosing this action initiates the",
"inference process to include those description",
"categories you may have left out in your original",
"input, and then determines the images that",
"best satisfy your current description based on any",
"of several attributes that describe the keyimages.",
"", "The query process is set up to give you the",
"opportunity to decide what value, if any, is",

```

```

"appropriate at that moment; otherwise choose none",
"and the inference will proceed just the same.", ""},

Display
{question: "Show images based on my initial input."}
{explain: "This action, although it's not very",
"elegant yet, asks you to choose none for each query",
"that wasn't determined in your initial input. From",
"there the images that satisfy the parameters of the",
"initial description are displayed."},

Exit
{question: "I want to begin over with an entirely new de
scription."}
{explain: "This action clears the slate and begins",
"the query/inference process from the top.", ""},

End
{question: "I want to return to the main program."})
{question: "", "PLEASE DETERMINE THE NEXT SET OF ACTIONS,",
"just to make sure.", "",
"For an explanation of Continue, Display, or Exit,",
"type 'explain' and the number that corresponds to",
"that value.", ""}.

image_mnu:sgl
(no good
{question: "", "I'd like to see others.", ""}
{explain: "", "The images are not satisfactory or the inference",
"process was unable to locate images that have an <a>",
"designation."},

ingres
{question: "", "Connect me to the visual database.", ""},
tell image
{question: "", "I'd like to see the description of particular ima
ges.", ""},

return
{question: "", "Return to the top menu.", ""},
Exit)
{question: "", "CHOOSE AN ACTION."}.

keyimage:mlt
(i1 {description:
Scope = primary secondary, env_controls;
Behavior = light modulation,
relationship between primary AND secondary,
ventilation;
Property = operable closure, fixed closure,
convective cooling;
Specs = Source, Image_type;
source = OTH;},

i2 {description:
Scope = primary secondary, formal organization;
Behavior = light modulation, secondary system;
Property = fixed closure, operable closure,
two OR three part ordering system, light shelf;
Specs = Source, Represents;
source = F;},

i3 {description:
Scope = primary secondary, env_controls,
formal organization;
Behavior = light modulation, secondary system,
ventilation, relationship between primary AND
secondary;
Property = operable closure, variable screens,
two OR three part ordering system;
Specs = Source, Represents, Image_type, Orientation;

```

```

        source = P;],

i4 [description:
    Scope = primary secondary, env_controls;
    Behavior = primary system, secondary system,
        relationship between primary AND secondary;
    Property = edge definition, masonry thermal damping,
        chimney effect;
    Specs = Source, Source_ref, Image_type, Color;
    source = OR;
    name = bldg_sect;],

i5 [description:
    Scope = primary secondary, env_controls;
    Behavior = light modulation, secondary system;
    Property = variable closure, operable closure,
        wooden screens;
    Specs = Orientation, Image_type;],

i6 [description:
    Scope = primary secondary, env_controls;
    Behavior = light modulation, secondary system;
    Property = variable closure, operable closure,
        fixed closure, wooden screens;
    Specs = Source, Source_ref, Represents, Image_type;
    name = window system;
    source = B;
    source_ref = "Sun Control: An International",
        "Architectural Study, Ernst Danz; p. 131";
    represents = s, pe;
    image_type = ort;],

i7 [description:
    Scope = primary secondary, env_controls;
    Behavior = light modulation, secondary system;
    Property = variable closure, operable closure,
        wooden screens;
    Specs = Represents, Image_type, Orientation, Color;],

i8 [description:
    Scope = primary secondary, env_controls;
    Behavior = primary system, thermal barrier;
    Property = concrete double wall, rigid insulation;
    Specs = Source, Source_ref, Represents;
    name = wall system;)].

%
\*****
externals:

    image_a:
        [program: "afile1"].
    image_h:
        [program: "hfile1"].
    image_m:
        [program: "mfile1"].
    image_l:
        [program: "lfile1"].

    setmode:
        [program: "chmode"].

    clean:
        [program: "clean_um"].

    parse:

```

```

        [program: "htout.parse"].

Ingres:
    [program: "start_ingres"].

%
\*****
actions:

    trace.

    read "com.bo", request.
        while request # combo do
            if request = scope
                then sc = true.
                Behavior = NONE.
                Property = NONE.
                Specs = NONE.
                name = NONE.
            else
                if request # combo and
                    request = behavior
                then b = true.
                sc = false.
                Scope = NONE.
                Property = NONE.
                Specs = NONE.
                name = NONE.
            else
                if request # combo and
                    request = property
                then pr = true.
                sc = false.
                b = false.
                Scope = NONE.
                Behavior = NONE.
                Specs = NONE.
                name = NONE.
            else
                if request # combo and
                    status(Specs) = known
                then sp = true.
                sc = false.
                b = false.
                pr = false.
                Scope = NONE.
                Behavior = NONE.
                Property = NONE.
                name = NONE.
                else Specs = NONE.
                endif.
            endif.
        endif.

    if Specs = NONE
        then sp = false.
        source = NONE.
        source_ref = "none".
        represents = NONE.
        image_type = NONE.
        orientation = NONE.
        color = NONE.
    endif.

    \ read "resp", Specs.

```

```

while query # End do
    if query = Display and
    sc or b or pr and
    sp = false
    then read "resp", Scope.
    read "resp", Behavior.
    read "resp", Property.
    obtain keyimage.
    message "", "", "THE FOLLOWING IMAGES SATISFY THE",
    "REQUIREMENTS OF YOUR CURRENT DESCRIPTION...",
    ""
    message "", "Any image designations that are followed by"
    "
    "<a> show that the image always matches the",
    "description; if <h>, then the image satisfies",
    "the description in most cases(.75 of the input"
    "
    "values are present); if <m>, then the image",
    "satisfies half of the requirements of your",
    "description; if <l>, then the image is a low",
    "match, only satisfying a 1/4 of the description"
    "
    "requirements.", ""
    display value of keyimage.
    write "htout", keyimage.
    run clean.
    run parse.
    run setmode.
message "THERE WILL BE A SHORT PAUSE WHILE THE IMAGES ARE LOCATED",
"AND CALLED TO THE SCREEN...", "", "If you wish to see other",
"images, or if the images displayed are not satisfactory,",
"wait for the next menu and...",
"", "JUST ASK TO SEE OTHER IMAGES.", ""
    run image a.
    display value of keyimage.
    while image_mnu # return do
        if image_mnu = no good
        then message "", "THE IMAGE(S) THAT WILL APPEAR",
        "IN A FEW MOMENTS MEET YOUR DESCRIPTION"
        "
        "WITH A HIGH<h> PROBABILITY(0.75).", ""
        run image h.
        display value of keyimage.
        message "Type 'c' to see more images.".
        break.
        message "", "THE IMAGE(S) THAT WILL APPEAR",
        "IN A FEW MOMENTS MEET YOUR DESCRIPTION"
        "
        "WITH A MEDIUM<m> PROBABILITY(0.50).", ""
        run image m.
        display value of keyimage.
        run image l.
        endif.
        if image_mnu = ingres
        then run Ingres.
        endif.
        erase image_mnu.
        image_mnu = return.
    endwhile.
    if image_mnu = return
    then erase query.

```

```

endif.
erase query.
erase keyimage.
endif.
if query = Continue
then erase request.
request = combo.
    if sc = true
    then read "resp", Scope.
    erase Behavior.
    erase Property.
    erase Specs.
    erase name.
    else
    if b = true
    then read "resp", Behavior.
    erase Scope.
    erase Property.
    erase Specs.
    erase name.
    else
    if pr = true
    then read "resp", Property.
    erase Scope.
    erase Behavior.
    erase Specs.
    erase name.
    else
    if sp = true
    then read "resp", Specs.
    erase Scope.
    erase Behavior.
    erase Property.
    erase name.
    endif.
    endif.
    endif.
endif.
erase query.
query = End.
endif.
if query = Exit
then erase Scope.
erase Behavior.
erase Property.
erase Specs.
erase name.
erase source.
erase source_ref.
erase represents.
erase image_type.
erase orientation.
erase color.
erase sc.
erase b.
erase pr.
erase sp.
erase request.
request = none.
erase query.
query = Display.

```



```

endif.

erase query.
query = End.

endwhile.

if request # combo
then erase request.
request = Exit.
endif.

endwhile.

while request = combo or
request = none and
request # Exit do

    if request = none
        then obtain keyimage.
    else
        read "resp", keyimage.
    endif.

    obtain keyimage.
    display value of keyimage.
    write "htout", keyimage.
    run clean.
    run parse.
    run setmode.
    message "THERE WILL BE A SHORT PAUSE WHILE THE IMAGES ARE LOCATED",
"AND CALLED TO THE SCREEN...", "", "", "If you wish to see other",
"images, or the images displayed are not satisfactory, wait",
"for the next menu and...",
"", "JUST ASK TO SEE OTHER IMAGES.", "", "These are the images",
"that match your description. Those followed by the <a> value",
"will be displayed first, then in descending order thereafter.".
    run image_a.
    display value of keyimage.

    while image_mnu # return do
        if image_mnu = no good
            then message "THE NEXT IMAGES ARE HIGH",
"PROBABILITY MATCHES<0.75>.".
            run image_h.
            display value of keyimage.
            run image_m.
            run image_l.
        endif.

        if image_mnu = ingres
            then run Ingres.
        endif.

        erase image_mnu.
        image_mnu = return.
    endwhile.

    if image_mnu = return
        then erase query.
    endif.

    erase request.
    request = Exit.
endwhile.

break.

stop.
%
```

POSTSCRIPT and ACKNOWLEDGEMENTS

I have been a research assistant since late 1985, working on a continuing project, "The Electronic Studio"(principal investigator: Frank Miller, Dept. of Architecture), which is administered under the auspices of The School of Architecture and Urban Planning's Computer Resource Lab. Without the lab's existence and commitment to exploring the use of new tools my work never would have occurred.

Hidden behind the scenes throughout this thesis is a research environment new and unique to MIT. A five year educational experiment called Project Athena has enabled institute wide access to, and funding for projects on, a unified computing environment. Project Athena is a joint effort between Digital Equipment Corporation, IBM, and MIT to explore the educational viability of using high performance graphics workstations distributed throughout labs, libraries, staff and faculty offices, departmental research facilities, and student living areas. Technically, Project Athena supports the UNIX(Berkeley 4.3 version) operating system, Digital Micro-Vax and IBM RT workstations, a multi-tasking windowing environment developed at MIT---XWindows, and a host of commercial and developmental software tools. The software utilized in the thesis includes Software A&E's KES(Knowledge Engineering System)HT and KES PS inference engines, an IBM AT based digital imaging board from AT&T and Island Graphics, Inc.---TARGA16, a TARGA to XWindows translation program under development by Dr. Steven Wertheim(Dept. of Brain and Cognitive Sciences) and Shih-Lin Liu, and RTI Ingres database management system.

My heartiest thanks must go to Dave Nelson and Barry Stanton: the three of us have suffered each other's best and worst moments. I also want to thank my advisor, Frank Miller, who I met in the desert seven years ago. He has been an instrumental collaborator on the ideas expressed in this work. Several people affiliated with the Dept. of Architecture, Phil Thompson, Andy Garvin, and James Anderson, gave valuable time towards helping me to achieve some measure of technical coherence in the digital imaging and selective manipulation of KES output files. Special thanks goes to Dorothy Shamonsky for giving me access to the Project Athena Visual Courseware Lab, for doing my paste-up, and for giving me her spiritual and emotional support.

BIBLIOGRAPHY

- Abelson, Harold and Sussman, Gerald J. *Structure and Interpretation of Computer Programs*. Cambridge, MA: MIT Press, 1985.
- Alexander, Christopher, et al. *A Pattern Language*. New York: Oxford University Press, 1977.
- Anderson, James. "Lecture Notes: Relational Formal Language(4.182)". Cambridge, MA: MIT, 1987.
- Bastlund, Knud. *Jose Luis Sert: architecture, city planning, urban design*. New York: Frederick Praeger, Inc., 1967.
- Beazley, Elisabeth and Harverson, Michael. *Living with the Desert: Working Buildings of the Iranian Plateau*. Wiltshire, England: Aris and Phillips Ltd., 1982.
- Boden, Margaret. *Artificial Intelligence and Natural Man*. New York: Basic Books, Inc., 1977.
- Chafee, Judith with Brock, Diana(College of Architecture, University of Arizona, Tucson). "Rancho de los Amigos," *Studio Brief: MIT Department of Architecture*. Cambridge, MA: MIT, 1986.
- Clark, Kenneth N. and Paylore, Patricia. *Desert Housing: Balancing Experience and Technology for Dwelling in Hot Arid Zones*. Tucson, AZ: University of Arizona Office of Arid Lands Studies, 1980.
- Danz, Ernst. *Sun Protection: An International Architectural Survey*. New York: Frederick A. Praeger, Inc., 1967.
- Evans, Martin. *Housing, Climate, and Comfort*. London: Architectural Press, 1980.
- Evenari, Michael, et al. *The Negev: The Challenge of the Desert*. Cambridge, MA: Harvard University Press, 1982.
- Futagawa, Yukio, Ed. with Pfeiffer, Bruce Brooks. *Frank Lloyd Wright Monograph 1937-1941*. Tokyo, Japan: A.D.A. Edita Tokyo Co., Ltd., 1986.
- Golany, Gideon, ed. *Desert Planning*. London: Architectural Press, 1982.
- _____. *Housing in Arid Lands: Design and Planning*. London: Architectural Press, 1980.
- Graf, Douglas. "Diagrams," *Perspecta*. 20 (1983). Cambridge, MA: MIT Press; pp. 42-71.
- Hallet, Stanley I. and Rafi, Samizay. *Traditional Architectue of Afghanistan*. New York: Garland STPM Press, 1980.
- Hansen, Neil L.R., and Radford, Antony D. "Living on the Edge: a grammar for some country houses by Glenn Murcutt," *Architecture Australia*, 5 (July 1986), pp 66-73.
- Herbert, Frank. *Dune*. New York:Berkeley Publ. Corp., 1965.
- _____. *Children of Dune*. New York: Berkeley Publ. Corp., 1976.
- _____. *Dune Messiah*. New York: Berkeley Publ. Corp., 1969.
- Hirschi, Suzanne and Max. *L'Architecture au Yemen du Nord*. Paris: Architectures, Berger-Levrault, 1983.
- Kalay, Yehuda E. "Redefining the role of computers in architecture: from drafting/modelling tools to knowledge-based design assistants," *Computer Aided Design*. 7 (Sept. 1985), p.319-328.
- Konya, Allen. *Design Primer for Hot Climates*. London: Architectural Press, 1982.
- Krieder, Jan F. and Krieth, Frank. *Solar Heating and Cooling: Active and Passive Design(2nd. ed.)*. Washington, DC: Hemisphere Publishing Corporation, 1982.
- Larson, Peggy. *A Sierra Club Naturalist's Guide, The Deserts of the Southwest*. San Francisco: Sierra Club Books, 1977.
- Lerman, Stephen R. "Questions and Answers about Project Athena," Cambridge, MA: MIT Project Athena, 1987.
- Lynch, Kevin. *A Theory of Good City Form*. Cambridge, MA: MIT Press, 1981.
- McHenry, Paul G. Jr. *Adobe and Rammed Earth Buildings, Design and Construction*. New York: John Wiley and Sons, Inc., 1984.
- The Merriam-Webster Dictionary*. Springfield, MA: G. & C. Merriam Co., 1974.
- Miller, Frank C. "The Electronic Studio," *Symposium: Computers in Design, Emerging Research Directions*. Cambridge, MA: MIT Department of Architecture and Urban Planning Computer Resources Lab, Graphic Systems, Inc., 1986.
- Minsky, Marvin. "A Framework for Representing Knowledge," in *The Psychology of Computer Vision*. Ed. P.H. Winston. New York: McGraw-Hill, 1975.

- Mollison, Bill. *Permaculture I and II*. Tasmania, Australia: Tagari Press, 1978 & 1979.
- Olgyay, Victor. *Design with Climate: Bioclimatic Approach Architectural Regionalism*. Princeton, NJ: Princeton University Press, 1963.
- Paternoster, Paul. *Catalog/Dialog*. Masters Thesis(M.S.V.S.). Cambridge, MA: MIT, 1984.
- Rasmussen, Steen E. *Experiencing Architecture*. Cambridge, MA: MIT Press, 1962.
- Schon, Donald A. *The Reflective Practitioner: how professionals think in action*. New York: Basic Books, Inc., 1983.
- Shamonsky, Dorothy. *Scripting Graphics with Graphics: Icons as a Visual Editing Tool*. Masters Thesis(M.S.V.S.) Cambridge, MA: MIT, 1984.
- Shepard, A. G., Fidler and Assocs., Derek Lovejoy and Partners, and Mander, Raikes, and Marshall. *Settlements in Dry Countries: A Design Approach*. Bristol, England: 1977.
- Software Architecture & Engineering. *KES HT: Knowledge Base Author's Manual(version 2.3)*. Arlington, VA: Software Architecture & Engineering, Inc., 1986.
- Software Architecture & Engineering. *KES PS: Knowledge Base Author's Manual(version 2.3)*. Arlington, VA: Software Architecture & Engineering, Inc., 1986.
- Varanda, Fernando. *Art of Building in Yemen*. Art and Archeology Research Papers. Cambridge, MA: MIT Press, 1982.