The Right Tool at the Right Time

Investigation of Freehand Drawing as an Interface to Knowledge Based Design Tools

A Thesis
Presented to
The Academic Faculty

By

Ellen Yi-Luen Do

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in Architecture

Georgia Institute of Technology
July, 1998

Copyright © 1998 by Ellen Yi-Luen Do
All Rights Reserved

This research was partly supported by EduTech Project D48-566, and Defense Advanced Research Projects Agency, monitored by ONR under contract N00014-91-J-4092 and Grants from National Science Foundation DDM-93-13186 and IRI-96-19856.

Georgia Tech College of Architecture

The Right Tool at the Right Time

Investigation of Freehand Drawing as an Interface to Knowledge Based Design Tools

Approved:

Craig M. Zimring

Craig M. Zimring, Chair

Mark D. Gross

Mark D. Gross, Co-Chair

Jean Wineman

Jean D. Wineman, Director, PhD Program

Date Approved by Chair 8/19/98

GEORGIA INSTITUTE OF TECHNOLOGY
Office of Graduate Studies

CERTIFICATE OF THESIS APPROVAL

ALL INFORMATION IS TO BE TYPED Date________________

Name: Yi-Luen Ellen Do

First Middle Last

Thesis Title: The Right Tool at the Right Time - Investigation of Freehand Drawing as an Interface to Knowledge Based Design Tools

We, the below signed, hereby state our full approval of the thesis submitted by the above student in partial fulfillment of the requirements for the degree of Doctoral of Philosophy in the ~~School of~~ College of Architecture.

APPROVED:

Craig M. Zimring
Chairman, Thesis Reading Committee

Janet L. Kolodner
Janet L. Kolodner
Member, Reading Committee

Mark D. Gross
Member, Reading Committee

Frank D.K. Ching
Member, Reading Committee

Jean D. Wineman
Member, Reading Committee

____
Member, Reading Committee

FOR PHD ONLY

The above named student completed the final doctoral oral examination on July 13, 1998

Date

Jean D. Wineman
School Director/Graduate Coordinator Doctoral Program

* *

(GRADUATE OFFICE USE ONLY)

The above thesis/dissertation has been approved by the Graduate Office.

____
Signature

____
Date

7/92

ACKNOWLEDGEMENTS

First, I thank my advisor Craig Zimring, for his continuous support in the Ph.D. program. Craig was always there to listen and to give advice. He is responsible for involving me in the Archie project in the first place. He taught me how to ask questions and express my ideas. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. I also thank the other two Archie project leaders, Janet Kolodner, who as my minor advisor guided my study of cognitive science and computing in my first two years at Georgia Tech, and Eric Domeshek, with whom I explored the ideas, organization, requirements and development of Archie II in MCL. Thanks also to Jim Foley (at GVU) for teaching me Human-Computer Interaction techniques and how to do a usability study, a skill that confirmed my intuition that designers need a drawing environment to access knowledge-based systems like Archie.

A special thanks goes to my co-advisor, Mark D. Gross, who is most responsible for helping me complete the writing of this dissertation as well as the challenging research that lies behind it. Mark has been a friend and mentor. He taught me how to write academic papers, made me a better programmer, had confidence in me when I doubted myself, and brought out the good ideas in me. (More importantly, he taught me how to work hard and play hard, and how to ski to reduce stress!) Without his encouragement and constant guidance (and the Tender Loving Care for me and my plants at the Sundance Lab), I could not have finished this dissertation. He was always there to meet and talk about my ideas, to proofread and mark up my papers and chapters, and to ask me good questions to help me think through my problems (whether philosophical, analytical or computational). He also made the Sundance Lab a wonderful workplace and home for the past four years by indulging my ever expanding bookshelf space and computer equipment needs. The beautiful weather in Boulder, Colorado has kept me in good spirits. Also thanks to the folks at the Sundance Lab for interesting discussions and being fun to be with. Thanks, Sundancers!

Besides my advisors, I would like to thank the rest of my thesis committee: Jean Wineman, who asked me good questions and rescued me from various red tape crisis, Janet Kolodner, for her friendship, encouragement and hard questions, and Frank Ching, who gave insightful comments and reviewed my work on a very short notice.

During the course of this work, at Georgia Tech (1992 – 1994), I was supported in part by Archie research grants from EduTech Project D48-566, and Defense Advanced Research Projects Agency, monitored by ONR under contract N00014-91-J-4092. At Boulder (1994 – 1998), I was supported by grants from the National Science Foundation DDM-93-13186 and IRI-96-19856.

Let me also say ‘thank you’ to the following people at Georgia Tech: Bill Bullock (Industrial Design Program) and Thomas Gallowy (Dean) to have confidence in me and for supporting my research and teaching activities at the College of Architecture. Sharon Williams (Doctoral Program), for helping me at any time, and solving any unsolvable problems concerning Ph.D. students. (Sharon, you are great!) and Mercedes Saghini, who continues the support of the program after Sharon. Linda Duckworth (International Students and Scholars Program), who dedicated her precious time and a drawer in the file cabinet for all my complicated records. Thank you, Linda! Suguru Ishizaki, Aaron

Fleisher, N. John Habraken, Barbara Tversky, John Peponis, Chuck Eastman, Ömer Akin, Ray McCall, Mike Gerzso and Sheng-Fen Nik Chien for insightful remarks. I am also grateful to the designers who participated in my design drawing experiment (my friends, you know who you are, thank you!), and the 62 design students in Ray McCall's Design Methods and Theory class in winter '94. Thanks also go to Janet Kolodner and Kevin Matthews for the use of screen shots from Archie and the Great Buildings Collection. Thanks to David Craig for help editing the final draft of the dissertation.

I am also greatly indebted to many teachers in the past: Ming-Hung Wang and Chao-Ching Fu (National Cheng Kung University in Taiwan) for getting me interested in Design Methods and Theory and coming to the United States. Wade Hokoda, Stephen Ervin and Bill Mitchell (Harvard GSD), for introducing me to CAD programs and concepts, and teaching me programming skills. Wade deserves a special mention. After hearing my complaints and frustration with the abilities of state of the art CAD tools, he helped me develop my idea and vision of what computers should do for designers, as well as with writing the statement of my research agenda for the Ph.D. Program. Wade lives on in the loving memory of his friends and family.

Last, but not least, I thank my family: my parents, Philip Y. Do, and Grace Lynn Do, for giving me life in the first place, for educating me with aspects from both arts and sciences, for unconditional support and encouragement to pursue my interests, even when the interests went beyond boundaries of language, field and geography. My sister Yi-Yin Do, for sharing her experience of the dissertation writing endeavor with me, for listening to my complaints and frustrations, and for believing in me. My brother Ben Roy Do, for refusing to answer my e-mails unless I have something interesting to say, for reminding me that my research should always be useful and serve good purposes for all humankind.

CONTENTS

ACKNOWLEDGEMENTS ii
LIST OF TABLES ix
LIST OF FIGURES x
SUMMARY xvii

1. INTRODUCTION 1

1.1. Right Tool at The Right Time 1
1.2. Importance of Drawing in Design 3
1.3. Why The Right Tool at The Right Time? 6
1.4. Goals of The Thesis 9
1.5. Assumptions and Hypothesis 10
1.6. Method of investigation 13
1.7. A Scenario 16

2. RELATED WORK – DESIGN STUDIES, COGNITIVE STUDIES AND SKETCHING PROGRAMS 23

2.1. Design Studies 24
2.1.1. Studies Based on Interviews and Portfolio Reviews 24
2.1.2. Designer’s Self Reports 27
2.2. Cognitive Studies 30
2.2.1. Drawing as External Representation 30
2.2.2. Relevant Research on Design Protocols 32
(1) What is Protocol Analysis 33
(2) Critiques of Protocol Analysis 33
(3) Protocol Studies of Design Activity 35
2.2.3. Discussion 39
2.3. Computational Support of Sketching 41
2.3.1. “Sketch” Systems 41
(1) Systems with ‘Sketch’ in Their Names 42
(2) Systems that Convert ‘Sketches’ to Objects 43
(3) ‘Sketch’ Systems that Support Sketching Activities 45
2.3.2. Why Computer Systems Need to Support Sketching 48
2.4. Discussion – Representation and Unstructured Drawing 50

3. BACKGROUND – ANTECEDENT PROJECTS 53

3.1. Studies on Design Process 54
3.1.1 Phase Models in Design Studies 54
3.1.2. Activities, Media and Information in a Design Phase Model 56
3.2. Archie - a Case-Based Design Aid for Architecture 60

3.2.1. Case-Based Reasoning 60
3.2.2. The Content and Features of the Archie System 61
3.2.3. Usability Study of Archie’s Case Library 66
3.3. Electronic Cocktail Napkin, a Sketching Environment 67
3.3.1. Making and Training the Computer to Recognize Drawing Marks 69
3.3.2. Defining Symbol Configurations by Example 71
3.3.3. The Relations between Symbols and Configuration 73
3.4. Archie's Napkin - Indexing Archie's Case Library with Diagrams 76
3.5. Drawing Analogies – A Shape-Based Reminding Scheme to Support Creative Design 79
3.6. Summary and Discussion 84

4. WHAT’S IN A DIAGRAM THAT A COMPUTER SHOULD UNDERSTAND 87

4.1. Archie diagram Story Experiment 88
4.2. Definition of a Diagram 89
4.3. Experiment Setups 91
4.3.1. Goal and Reasons for Conducting The Experiment 91
4.3.2. Experiment Sequences and Test Materials 91
4.4. Experiment Results – What’s in a Diagram? 97
4.4.1. Pre-evaluation Questionnaire Results 97
4.4.2. Task 1 - Drawing Elements & Viewing Preferences in Diagram-Making 98
(1) Concepts Suggest Sectional or Plan Preferences. 99
(2) Participants Used a Limited Set of Symbols to Represent Design Concepts. 101
(3) Plan Diagrams Have More Text Labels Than Sections. 104
(4) Diagrams Included Labels and Symbols not Mentioned in The Text. 106
4.4.3. Task 2 - Consistent Interpretation of Diagrams among Designers 107
4.4.4. Task 3 - Designers can Easily Associate a Diagram with Its Pairing Text 108
4.4.5. Task 4 - Designers Agree with Archie Diagram-Text Pairs 110
4.4.6. Post-Evaluation Questionnaire Results 111
4.5. Discussion – Implications for Diagrams and Computing 113
4.5.1. Designers’ Shared Understanding of Diagrams 114
4.5.2. Future Experiments 114
4.5.3. Future Work – Diagrams as an Index to Design Information Systems 115

5. DESIGN DRAWING EXPERIMENT – DESIGN CONCERNS & DRAWING CONVENTIONS 117

5.1. The Experiment – Setups, Tasks and Program 118
5.2. Profiles of Participating Designers 121
5.3. Summaries of Design Sessions. 124
5.3.1. Roger, The Functional Designer 124
5.3.2. Noi, the 3D Sketcher 131
5.3.3. Samuel, the Philosopher 139
5.3.4. Mario, the Research Architect 144
5.4. Detailed Analysis of the Design Sessions 150
5.4.1. Functional Considerations of Roger’s Design Session – Diagram Conventions and Dimensional Reasoning 150
5.4.2. Noi’s 3D Sketches – Lines for Partitioning, Lighting and Hatching 151
5.4.3. Philosopher Samuel’s Design Process – Arrows for Lighting, Entrance and Visual Access 153
5.4.4. Research Architect Mario’s Concerns – Scale, Dimensions and How Things Work 155

5.5. Post-Experiment Questionnaire 158
5.6. Experiment Results: Design Context and Drawing Conventions 160
5.6.1. Designers Share Drawing Conventions: Symbols and Their Configurations 161
5.6.2. Designers Drew Furniture in Space and to Put Themselves in Context 162
5.6.3. Lighting Concerns are Portrayed in Sectional View with Light Rays 163
5.6.4. Attention and Focus Can be Identified through Labels and Overtracing 164
5.6.5. Dimensional Reasoning through Figure Calculations 165
5.7. Summary and Conclusions 166

6. A PROTOTYPE SYSTEM FOR INFERRING DESIGN INTENTION 168

6.1. System Overview 170
6.1.1. System Architecture 170
6.1.2. Recognition Phases 173
6.1.3. Recognition of Symbols and Configurations 174
6.2. Context Recognition 176
6.2.1. Representation of a Drawing Context 178
6.2.2. Initial Context and Current Context 180
6.2.3. Mapping Named Symbols for Different Contexts 182
6.3. Intention Recognition 183
6.3.1. Recognition of an Intention Configuration 184
6.3.2. Context and Intention 187
6.4. What the Computer Does Once It Determines Intention 188
6.4.1. Identifying the Commands to send to Design Tools 188
6.4.2. Design Tools Currently Supported by Right-Tool-Right-Time 194
6.5. How Does the Scenario Work? 196
6.6. Discussion 200

7. CONCLUSION, DISCUSSION AND FUTURE WORK 203

7.1. Summary of the Dissertation 203
7.2. discussions 210
7.2.1. Contribution of this Dissertation and the Lessons Learned 210
7.2.2. Limitations of the Current Work 214
7.3. Future Work 219

A DRAWING ANALOGIES 223

A.1. Finding and Using Visual References in Design 224
A.2. Remarks on the Scenario 228

B. SYMBOLS FOR HUMAN FIGURES FROM THE DIAGRAMMING EXPERIMENT 230

C. DESIGN EXPERIMENT QUESTIONNAIRE 232

Consent and Release Form 233
Please tell us about yourself. 234
Design Brief 235
Task 1 235

Task 2 236
Task 3 237
Task 4 238
Please tell us how you feel about this experiment. 239

D. ROGER'S DESIGN SESSION 241

E. NOI'S DESIGN SESSION 255

F. SAMUEL'S DESIGN SESSION 265

G. MARIO'S DESIGN SESSION 277

H. VIDEO TRANSCRIPT OF MARIO'S DESIGN SESSION 291

REFERENCES 341

LIST OF TABLES

Table 4-1. Years of education distribution of participants. 92

Table 4-2. Groups and their respective test variations. 92

Table 4-3. Standard Task sequence in a test (for Group A and B). 93

Table 4-4. Pre-evaluation self-assessment of diagram use. 97

Table 4-5. Pre-evaluation, self-assessment of computer program usage, applications used and expectation of application types to have diagramming tool. 98

Table 4-6. Number of plan and section diagrams drawn in Task 1. 100

Table 4-7. Between group representations of light passing through building envelope. 103

Table 4-8. Comments on Archie's text and diagram pair. 110

Table 4-9. Post-evaluation self-assessment of diagram elements used. 112

Table 4-10. Comment on how a diagram and a tool provide 112

Table 5-1. Profile of participating designers 122

Table 5-2. Response of the experiment from participating designers 158

Table 6-1. Elements of a Context 'PLAN' in four categories: 1) glyph templates, 2) configuration recognizers, 3) spatial-relations, and 4) mappings. 178

Table 6-2. A context, intention, drawing symbols and application table. 189

Table 6-3. Interaction types of RTRT to various tools: 1) activation only, 2) keyword query to a search engine, and 3) translation of drawings to different format. 190

LIST OF FIGURES

Figure 1-1. This dissertation is about design drawing. The discussion of a design drawing environment is facilitated through 1) survey of literature and related work, 2) empirical studies on design process, and 3) computational implementation. 13

Figure 1-2. Designer's diagram retrieves a relevant story from a case library, Archie. Left: Bubble diagram of spatial arrangements in a sketchbook. Right: Archie case story of a library lobby with similar configuration. 16

Figure 1-3. Designer's drawing can be cleaned up and rectified if desired. Left:Sketchy drawing of spatial arrangements. Right: Rectified geometric shapes of the sketchy drawing. Notice the check mark on the left was a gesture request for cleaning up the drawing and therefore was not translated on the right. 17

Figure 1-4. Designer's drawing of visual enclosure brings up a viewshed simulation program, Isovist. Left: Diagram to explore visual access. Right: Isovist viewshed analysis tool 18

Figure 1-5. Designer's sketches is transformed into a 3D CAD model. Left: Sketches of massing study. Right: Translated 3D model that allows changes of isometric projection angles. 19

Figure 1-6. Designer's diagram retrieves information from a multi-media database. Top left: Diagram of a triangle pediment and colonnade. Top right to bottom left (clockwise): Textual information, picture and QuickTime animation of the Parthenon from the Great Buildings Collection (Matthews, 1994). 19

Figure 1-7. Designer's diagram retrieves information of HVAC production s from Netscape. Left: Retrieved page from the World Wide Web through Netscape program. Middle: Retrieved sketchbook of a note on HVAC configuration. Right: Designers diagrams on top of a floor plan. 20

Figure 3-1. Organization drawings explore programmatic concerns. 57

Figure 3-2. Ideation drawings explore shapes and forms. 57

Figure 3-3. Fabrication drawings tend to be more definite and precise. 58

Figure 3-4. System diagram of Case-Based Design Aid Archie. 62

Figure 3-5. Archie screen shot. Top: windows for Problem, Story and Response. Bottom: Annotations (summaries) and Design (display of design artifact, e.g., floor plan). 63

Figure 3-6. Accessing Archie case library through a search window by selecting predefined keyword. (Archie screen consists of panels for problem, story, response, annotation and design floor plan). 65

Figure 3-7. The Electronic Cocktail Napkin - [a] drawing area, [b] sketchbook for storing interesting sketches. 68

Figure 3-8. Features of simple glyphs: pen path in a 3x3 grid, raw x, y points, and corners. 70

Figure 3-9. A person symbol can be defined as a circle placed directly above a triangle, while a computer monitor can be defined as two concentric boxes. 71

Figure 3-10. Defining a configuration by example: The Napkin shows relations, the user edits them. 72

Figure 3-11. A configuration can be named and define in a search window that displays and analyzes element types and spatial relations between elements. For example, shown here a 'temple' is defined with a 'roof' immediately above row of 'columns' immediate above a 'base.' 73

Figure 3-12. A hierarchy of spatial relations and element types enables flexible matching. 75

Figure 3-13. An illustration from a library lobby story with corresponding diagram. 77

Figure 3-14. Indexing Archie stories with 'bubble diagram' drawn on the Cocktail Napkin program 78

Figure 3-15. Element Type Match: pairs [a, b] and [c, d] have exactly the same element types. 80

Figure 3-16. Relations match -- all figures exhibit 'containment'. 81

Figure 3-17. Element type&relations match: [a] is similar to [b], but not to [c] or [d]. 81

Figure 3-18. Some dimensions of diagram similarity: [a] compared with [b, c, d, e]. 82

Figure 3-19. Parts of the Drawing Analogies system. 83

Figure 4-1. A case of adult and children’s sections arrangement in library indexed and retrieved by a diagram. (A for adult and C for children). 89

Figure 4-2. Material for Task 2a, Problem 1, a diagram with title. 95

Figure 4-3. Example page for pairing test, Task 3, diagrams on top row, and texts on bottom row with different sequence. 95

Figure 4-4. Material for Task 4f, Response 3, diagram-text set from Archie. 96

Figure 4-5. Diagrams to illustrate using thick walls to reduce sound transmission included 1) a simple rectangle to represent a thick wall, 2) detailed floor plan configuration, and 3) concepts not from text (restrooms). 99

Figure 4-6. Diagram examples showing view preference for descriptions, Problem 1- plan representations, Problem 2 - sectional diagrams and Problem 3 - both plan and section diagrams. 100

Figure 4-7. Diagram examples showing view preference for Response descriptions, Response 1 - plan view, Response 2 - both plan and sectional and Response 3 - sectional representation. 101

Figure 4-8. Primitives (drawing elements) used in diagrams included arrows, lines, hatches and simple geometric shapes 102

Figure 4-9. Primitives were composed to make more complex symbols. 102

Figure 4-10. Diagrams illustrating lighting use light lines penetrating the building envelope (windows and roofs). Problem 2, Task 1. 103

Figure 4-11. Diagrams use keyword “map” given in the text to label a space (Problem 1 of Task 1). 105

Figure 4-12. Diagrams use keyword “public” and “court” or “P” and “C” from given text to label space, Response 1, Task 1. 105

Figure 4-13. Sub-tasks for Task 2, create text from diagrams. (a) Music Room Issue. (b) Overheating caused by Overhead Glass. 107

Figure 4-14. One participant failed to associate this diagram with its text because parking lot division lines look like bookshelves. 109

Figure 4-15. Participants suggested changing this diagram to a three dimensional view (Response 2 of Task 3) 109

Figure 4-16. Diagram showing music room issues, given for Response 1 of Task 4. 110

Figure 4-17. Designers used conventional symbols and configurations for architectural concepts in diagrams. 114

Figure 5-1. Functional designer Roger’s process of understanding the design program: (a) part of the program was illustrated on the right side to represent site, and (b) personnel and space requirement in the program were underlined, framed by boxes, and linked by double arrowhead lines to indicate relationships. 125

Figure 5-2. Functional designer Roger’s bubble diagram showing conceptual design for spatial arrangement for Task 1. 126

Figure 5-3. Roger’s drawing for Task 2: (a) a lobby and receptionist area showing a round table with couch, and secretary area with table and counter, both have opening to inside space. (b) a chief architect's office showing 1 table with a computer on it, a chair for the architect, a side table serves as a conference table with a chair for client. (c) conference room, showing an oval shape meeting table with surrounding chairs, wall with windows, a door to the hallway, supporting facility such as a counter with a sink, a refrigerator (on the top right corner), a podium and a blackboard (on the left), and then (d) a section showing different ceiling height (10' and 8') 127

Figure 5-4. Roger’s drawing for Task 3, lighting: (a) the design of a courtyard place that allows lighting and visual access to all space surrounds it. (b) a sectional drawing depicting a sun with light ray comes in through the window to a room where a person sits and works with a computer. 128

Figure 5-5. Roger’s drawing and notes explaining design rationale for the spatial arrangement. 129

Figure 5-6. Process of calculating of space requirements and dimensioning:1) mark the width and length of the site, 2) calculate to make room of 800 sq. ft. for work space, 3) calculate the left over space, divided by 10' intervals, 4) check dimension of table (4x10) to fit inside the conference room, calculate the space. 130

Figure 5-7. 3D sketcher Noi’s understanding of the design program:

(a) thumb nail site plan illustrating dimensions, directions, and adjacent to existing building on the south wall. (b) underlines of personnel information, and use of different shapes to frame space names - rectangles for designers, draftsperson and office, triangles for kitchenette, bathroom and storage, round shapes for meeting room, and rounded cornered rectangles for secretary and receptionist. 131

Figure 5-8. Noi’s spatial layout illustrated by using partitioning lines: (a) plan with walls, rooms, and corridor space. (b) plan with space partitioning lines focuses on openings, hatched line for solid wall on the South (top), dotted window mullions on the East (left). 132

Figure 5-9. Noi's drawing of an isometric perspective that was derived from extruding floor plan. (a) the perspective view showing light ray coming in from the window. (b) a floor plan drawn to explain windows on the left side. 133

Figure 5-10. Noi’s series of layout floor plans illustrated different focus (in drawing sequence a–e) (a) overall layout, (b)defining architect's office, (c) further consideration of path, hallway and partition, (d) window and kitchen counter, (e) reception area 134

Figure 5-11. Noi’s drawings for lighting concerns: (a) top (left to right) sectional perspective, thumbnail section, bottom ceiling plans, lighting fixture. (b) skylight perspective drawn from a dome section 136

Figure 5-12. Noi’s drawings for task 4: (a) Sectional perspective showing windows, chief architect's office in the back, reception area with couches, table, painting on walls and people. (b) plan, (c) section. 138

Figure 5-13. Noi’s sectional perspective drawing show a conference table in front of the chief architect's office 139

Figure 5-14. Samuel’s drawing showing site, program notes (right), direction north (arrow N), sun light (arrow from bottom, south east), zoning (public, service, display, entrance), and level change to architect's office (bottom right rectangle with steps beside it). 140

Figure 5-15. Samuel’s bubble diagram for a spatial layout arrangement for Task 1. 141

Figure 5-16. Samuel’s floor plan showing north lighting (arrows on top), tables of work space (rectangles), lighting ceiling plan (hatched long rectangles) and a sectional drawing illustrating how lights come in and be reflected and controlled by the skylight windows and roof. 142

Figure 5-17. (a) Plan of different space arrangement. (b) trace on top of the plan and added furniture (table and counter), steps to architect's office, lighting and circulation. 143

Figure 5-18. Samuel’s testing on the spatial arrangement - detail furniture and room partitioning. 143

Figure 5-19. Research architect Mario’s drawings (a) Conversion of 25 ft and 70 ft into meters (8 and 23). (b) site plan interpretation of the program, (c) written notes of personnel and space information of the design program on the tracing paper. 144

Figure 5-20. Mario’s floor plan for Task 1 – he used detail furniture arrangement to test how the space would work. 145

Figure 5-21. Mario’s second floor plan has columns traced on top of last layer (Figure 5- 20, Task 1) 146

Figure 5-22 Mario’s responses to Task 2, lighting concerns
(a) Used gesture to explain about his design. (b) Section A-A depicting computer work stations. (C) Section B-B depicting a work space cubicle, the corridor space, and the meeting room. 147

Figure 5-23. Mario’s drawing for Task 3, visibility issues
(a) Original floor plan layout designed in the task 1 session. (b) Modification of switching the location of conference room and kitchenette (to avoid seeing dirty dishes) in the task 3 session. 148

Figure 5-24. Calculation and conversion about dimensions. 149

Figure 5-25. Function designer Roger's convention. Top
(1) bubble diagrams, (2) furniture, (3) lighting section, (4) double arrow line for relations. Bottom, (5) building elements wall, window and door, (6) underlining and framing program, (7) dimensional reasoning, (8) shorthand of the site plan as understanding and reminding of the program 151

Figure 5-26. Noi's design conventions
(1) shorthand of the site plan, (2) underline and different shape framing program requirement descriptions, (3) spatial partitioning for layout, (4) lighting penetrate building envelope, (5) label and hatching of space, (6) dots and lines as windows and walls, (7) making 3D from plan or sections. 152

Figure 5-27. Designer Samuel's convention: (1) underlined program, (2) label with text and use arrow, lines and circle in bubble diagram, (3) roof section lighting, (4) arrows of lighting, entrance and visual access, with hatching, (5) architectural elements stair, tables, windows and walls. 155

Figure 5-28. Mario's drawing conventions: (1) small site plan with direction north, and dimensions, hatched wall, (2) copy down design program, (3) drew same kinds of object in sequence, 3 monitors, 3 tables and 3 chairs, (4) door, wall and window, (5) cut section to show lighting and space, (6) use of furniture to test how the space works, (7) dimensional reasoning, conversion of number calculations, and (8) labeling the d imensions on side 157

Figure 5-29. Top: Primitives (drawing elements) used in drawing included arrows, lines, hatches and simple geometric shapes. Bottom: Symbols (architectural objects) such as walls, windows and stairs are formed by combination of primitives 161

Figure 5-30. Symbols give clues for context. Top row
different designer's symbols for the conference space (Roger, Noi, Samuel, Mario). Bottom row: lobby, office (Roger) and cubicles (Samuel, Mario). 162

Figure 5-31. Drawings illustrate lighting by using light lines that penetrate building envelope from windows and roof. The changed direction arrows indicate reflecting light from a roof skylight (second to right) and in an interior lighting fixture design (right). 163

Figure 5-32. [a] a bubble diagram for layout and zoning, [b, c, d, e] designers used overtracing (circle) to select a space, they put labels inside the shapes, or drew lines out to indicate space use, [f] hatching is another way to emphasize a certain area and distinguish among other spaces 164

Figure 5-33. Dimensional reasoning with annotated numbers: Left: dimensions labeling of site (Roger, Noi and Mario). Middle: Roger's design process of square footage requirement for Task 5. Right: Mario's calculation and conversion of feet to meter and square footage. top: 70 ft x 25 ft became 23 m x 8 m, bottom: 800 sq. ft. was converted to 20 ft x 40 ft to 6.7 m x 13 m 165

Figure 6-1. The Right-Tool-Right-Time is conceived as a drawing environment with an inference engine that monitors the design drawing and activates different tools. 170

Figure 6-2. The conceptual model of a drawing environment. Top: different drawing symbols and configurations (bubble diagram, facade, view lines, calculation and 3D solid). Middle, the Right-Tool-Right-Time system that 1) identifies drawing contexts (contexts are illustrated inside boxes), 2) design intention recognizer (rounded box) infers design intentions (from symbols and configurations, intentions are illustrated inside boxes), 3) inter-application communicator (rounded box) activates different tools based on detected intentions through drawing symbols and configurations. Bottom: different design tools – Archie case library, the Great Buildings Collection, Isovist viewshed analysis program, a calculator, and a 3D modeling program. 171

Figure 6-3. Phases of the intention recognition 173

Figure 6-4. Designers' drawing symbols recognized by the Electronic Cocktail Napkin Program. 174

Figure 6-5. A configuration called 'window' is identified by recognizing a vertical line directly connected to two long rectangles, one above and one below. (Note that the window symbol can exist in either a plan or a section context, the ambiguity remaining until further symbols are recognized.) 176

Figure 6-6. A 'ground' symbol is unique to a 'section' context. Therefore, after recognizing the 'ground' context (horizontal line directly above wiggly line), the system sets the current context to be 'section.' 177

Figure 6-7 A context specific configuration called 'ground' is identified by a horizontal line directly drawn above a wiggly line. As a result of recognition of this unique configuration, the current context is then set to be 'section.' 181

Figure 6-8. An intention configuration called 'monitor_glare' can be defined by co-existence of a lighting symbol configuration 'light' (sun and arrow) and a configuration called 'monitor' (concentric boxes) in the sectional drawing context. 184

Figure 6-9. Incremental definition and recognition of drawing symbols to configuration, to design intention. Left
a 'horizontal-line' above a 'wiggly-line' is recognized as a 'ground' configuration, this unique configuration sets the context to be 'section' 186

Figure 6-10. An 'intention configuration' can be defined by combination of a 'light' configuration and a 'window' configuration. 186

Figure 6-11. A number with neighboring arrows indicates dimensioning, while arithmetic marks together with numbers indicate a calculation intention. 188

Figure 6-12. A sample code showing the sub-functions of the command 'activate-netscape.' It first brings up a message to inform the user that the system is activating a browser, and then issue a command called 'netscape-go-to-url' to form a query to the browser with the string passed in from the recognized 'intention.' 192

Figure 6-13. A configuration of numbers and arithmetic marks (left) is recognized as an "adding" intention (middle) and RTRT activates a calculator (right). 197

Figure 6-14. A ‘monitor_glare’ intention is recognized (bottom left) and RTRT prompts a message (top left) and activates a Web Browser (right), requests a keyword search and the search returns a list of relevant information. 198

Figure 6-15. A ‘view’ intention is recognized in a ‘plan’ context (top) and the drawing is translated into wall lines (from lines) and a view point (arrows and circle) and sent to the Isovist visual simulation program (bottom). 200

Figure A-1. Designer's basic scheme for spatial arrangement 224

Figure A-2. Forms found in a handbook of the natural world. (Drawings by author, after (Gibbons 1984), and (Webb 1948; Alexander, Ishikawa et al. 1977)) 225

Figure A-3. Graphic representations used in design development. 226

Figure A-4. Tracing a magazine clipping into the design drawing (design clipping by author after Kallmann, McKinnell & Wood's American Academy of Arts & Sciences, (Mackintosh 1977)). 227

Figure A-5. Combining ideas and stretching the drawing to suit the design. 228

SUMMARY

Designers use different symbols and configurations in their drawings to explore alternatives and to communicate with each other. For example, when thinking about spatial arrangements, they draw bubble diagrams; when thinking about natural lighting, they draw a sun symbol and light rays. Given the connection between drawings and thinking, one should be able infer design intentions from a drawing and ultimately use such inferences to program a computer to understand our drawings. This dissertation reports findings from empirical studies on drawings and explores the possibility of using the computer to automatically infer designer's concerns from the drawings a designer makes.

This dissertation consists of three parts: 1) a literature review of design studies, cognitive studies of drawing and computational sketch systems, and a set of pilot projects; 2) empirical studies of diagramming design intentions and a design drawing experiment; and 3) the implementation of a prototype system called Right-Tool-Right-Time.

The main goal is to find out what is in design drawings that a computer program should be able to recognize and support. Experiments were conducted to study the relation between drawing conventions and the design tasks with which they are associated. It was found from the experiments that designers use certain symbols and configurations when thinking about certain design concerns. When thinking about allocating objects or spaces with a required dimensions, designers wrote down numbers beside the drawing to reason

about size and to calculate dimensions. When thinking about visual analysis, designers drew sight lines from a view point on a floor plan.

Based on the recognition that it is possible to associate symbols and spatial arrangements in a drawing with a designer's intention, or task context, the second goal is to find out whether a computer can be programed to recognize these drawing conventions. Given an inferred intention and context, a program should be able to activate appropriate design tools automatically. For example, concerns about visual analysis can activate a visual simulation program, and number calculations can activate a calculator. The Right-Tool-Right-Time prototype program demonstrates how a freehand sketching system that infers intentions would support the automatic activation of different design tools based on a designers' drawing acts.

CHAPTER I

INTRODUCTION

1.1. RIGHT TOOL AT THE RIGHT TIME

Imagine a workman doing household jobs. He has a tool box filled with various tools: hammer, screw driver, pliers and a drill. All these tools are useful, but only if they are applied to the right job at the right time. No matter how useful a sanding kit is, it will not drive screws or fasten nuts and bolts. A hammer will be useful for striking nails but not for drilling holes. An experienced assistant watching the job can hand the appropriate tools to the workman at the right time.

This dissertation focuses on the drawing conventions designers use that a computer might understand, and the relationship between drawing and design intent. It describes a computer–based freehand sketching environment for design that tries to deliver the right tools at the right time by interpreting a designer's drawings. Rather than asking the designer to find and select tools for specific design tasks, the thesis explores the idea of automatically invoking various computational tools based on the designer's drawing.

The goal of this dissertation is to develop a proof-of-concept computer system that automatically presents designers with useful information resources during sketching: the system gives them the right tool at the right time. Developing such a system creates both conceptual and technical problems. If a sketching environment is to present the designer with an appropriate tool, it must recognize what the designer is doing. Conceptually, this requires that there is a link between overt behavior--the marks that the designer makes on the paper or in the computer--and the designer's information needs. Hence the first part of the dissertation is devoted to literature review and empirical studies to understand if there are drawing conventions shared by architectural designers that might be incorporated into a sketching environment. It is argued that designers *do* share drawing conventions, both at the level of the individual glyph and at the level of the combination of symbols, and that these conventions map onto the tasks that the designers are engaged in. In the second section, the dissertation discusses the technical implementation of the design environment. Whereas the environment is focused on a specific problem – office design – it extends previous sketching environments in several ways. It allows designers to keep sketches in ambiguous, "sketchy," states; it operates on complex systems of symbols within sketches, including the topological relationships among symbols; it provides access to a wide range of additional information tools

The goal of many intelligent computer-aided design systems, and in particular knowledge-based systems, is to provide advice in the form of critiques, relevant cases or examples, and the results of simulations. How can these systems decide what advice might be useful and when to provide to the designer? Can intelligent design systems determine what advice to give designers by looking at their drawing? The drawing conventions that designers share in designing may be a good indication of the contexts and concerns they are interested in at the time. If so, this raises the question: what drawing conventions should a computer understand?

1.2. IMPORTANCE OF DRAWING IN DESIGN

In the early stages of a design process, designers usually draw diagrams and sketches to explore ideas and solutions. Designers—especially architects—are trained to use paper and pencil when developing conceptual designs. Architects are visually oriented and are taught to think graphically (Laseau, 1980; McKim, 1972). They draw to develop ideas graphically, and in the process of drawing, designers communicate their thinking. The iterative act of drawing on paper involves recording ideas, recognizing functions and meaning from the drawings, and finding and adapting new forms into design. The act of drawing is important not only as a vehicle for communication with others; it actually helps designers see and understand the forms they work with (Edwards, 1979).

“Drawing” in this dissertation is used to mean the as freehand diagrams and sketches[1] designers draw and use in their early design stages. Several architectural books focus on specific drawing methods and techniques. They describe and illustrate drawing conventions (e.g., bubble diagrams, concept sketches, etc.) for different objects (walls, windows, tree,

[1] Designers use the term “drawing” more generally to represent all the kinds of marks they make when designing; ranging from concept sketches to construction drawings. Here in particular, I am referring to freehand drawings instead of drafting or construction drawings. Though different in nature, diagrams and sketches are commonly used together by designers and it’s hard to restrict a designer to use only one kind. In another word, diagrams in architecture are usually mixed with sketches. For example, the diagramming experiment (described in Chapter 4) on Archie case stories specifically asked participating designers to use “diagram” instead of sketching, but still hybrids of the two kinds of drawings appeared in the results. A detailed discussion of the diagramming experiment and the difference between diagram, schematic drawing and sketches are discussed else where (Qualify paper, (Do, 1995a) and also Section 4.2). In this dissertation, I shall use “drawing” to include both diagrams and sketches designers make in the early design process.

pavement) to help novices learn how to use drawings to design. They use drawings to show how to approach and analyze a design.

For example, in “Design Drawing Experiences,” Lockard proposes that the ability to “diagram” a context depends on designers’ knowledge of related issues in a setting, such as sun, wind, vegetation, traffic and surroundings. He proposed that the value of diagramming is to explore variations of design problems and allows our mind to “see, comprehend and respond” to more visual information than one can remember from verbal notes (Lockard, 1973). Laseau’s “Graphic Thinking” (Laseau, 1980) is a guide to making drawings for working out problems and for communicating with others. He describes drawing as a means for design development and indicates that designers draw different symbols to think about different aspects of their design. Ching’s “Architecture: form, space and order” (Ching, 1979) is a thorough illustration of how different buildings can share the use of basic geometric shapes to create form and space. His detailed analytical sketches showed designers how to “read” architectural drawings and to think with a pen. He argues that the meaning of architecture can be defined by the elements of form and space, through the drawing of points, lines, shapes and volumes. He argues that the art of architecture lies in drawing, with “connotative meanings – associative values and symbolic content” (p. 386), and encourages readers to take visual notes to analyze and understand the elements and ordering of design.

These authors base their books on their own experience practicing architecture and teaching design and drawing. They do not explicitly mention the use of graphic symbols. However, illustrations in the books show many examples of graphic symbols such as ovals for space, straight lines for views, squiggly winding lines for wind, and pictograms of human figures and the sun. For example, Lockard demonstrates the practice of drawing figures, trees, and furniture to identify the content and context of the design. He calls the arrangement of drawings a “design process sentence” (p 110). Laseau uses the term “graphic language” and illustrates the idea of graphic vocabulary and grammar with shapes, lines and arrows. Graphical notation systems and visual languages have been discussed in different disciplines; Goel, for instance, provides a view from cognitive science and Thiel provides a view from the environmental behavior (Goel, 1995; Thiel, 1961). The scope of this dissertation is limited to identifying graphic symbols and configurations for specific tasks in architecture design.

In summary, although these pedagogical books do not specifically mention the use of graphic conventions in drawing, there is ample use of symbols as communication clues in their illustrations and text. And interestingly, the books employ a relatively small and consistent set of symbols, which suggests that one can learn these drawing symbols and identify them with design intentions.

This dissertation studies graphic conventions used in design drawing. In particular, the research focuses on symbols and configurations of sketchy design diagrams. Architectural diagrams are usually sketchy, and sketches also contain diagrammatic symbols. Designers use the words “diagram” and “sketch” to represent freehand drawings they make in the early conceptual design phase. This research focuses on functional aspects of these sorts of design drawings.[2] These functional aspects includes design issues of environmental forces, building components, and human behavior.[3]

[2] Architectural design involves both form and function. Most of the drawing examples I have in this dissertation is about the functional aspects in design because this research originated from the participation of the Archie system development. Archie is a case library of post occupancy stories and therefore emphasizes on functional aspects than formal aspects. However, the Drawing Analogies (see Appendix A) is an attempt to deal with formal aspects of design; it is a shape based reminding scheme for design.

[3] See section 4.2 for more detail description of these design issues.

1.3. WHY THE RIGHT TOOL AT THE RIGHT TIME?

Current Computer-Aided Design (CAD) programs only support design partially because they do not provide a task-oriented environment for design.[4] They merely support representation (e.g., 2D drafting or 3D modeling) or feedback (e.g., simulation, evaluation) after design decisions are made. None provide contextual assistance to designers by recognizing the design tasks designers are thinking about. First, these design tools need to be accessible in the early conceptual design process when design decisions are made on the basis of freehand drawings and changes are less costly. Second, these CAD tools require designers to specify their intent and all have their specific ways of interacting with the systems and thus obstruct the design flow. These tools are each aimed to support some aspect of design. However, they fail to address the big picture. To be really useful in a design process CAD tools should 1) support the design drawing environment, and 2) become available for the task at hand at the appropriate time. The following briefly discusses the need for having right tools available at the right time for design.

Since the early nineteen-sixties, many researchers have investigated using computers to help designers. Geometric modeling and drafting programs (e.g., Form-Z, AutoCAD) have been built to help speed the creation of models and working drawings, but not the early conceptual design drawings. These systems support the construction phase of the design process. Evaluation tools, information and expert systems (e.g., (Maher, 1985; Pohl, Chapman, Chirica & Myers, 1988; Rittel & Kunz, 1970)) have been developed to

[4] Yes, many computer aided design tools support design process by recording design ideas or identifying emergent shapes. However, I argue that a design process can be supported better if these tools can be activated based on the task at hand.

help design decision making by providing design rationale or design guidelines. Shape grammar systems (e.g., (Flemming, 1987a; Knight, 1981; Stiny & Mitchell, 1978)) compute geometry information to guide generation and transformation of a design artifact such as a floor plan or facade. Case-based design aids (e.g., (Kolodner, 1991; Oxman, 1993a; Zimring, Do, Domeshek & Kolodner, 1994) propose using past experience or design precedents to help designers to avoid making known mistakes and to adapt old cases to new situations.

These tools, though beneficial to drafting and design thinking, are cumbersome to use during the early stages of design (Do, 1995a). It takes designers too much time to learn and to become accustomed to their operation.[5] Designers often must learn obscure commands and type in text in order to fulfill their goals. The command and text-oriented interface obstructs the flow of the design process. Furthermore, these tools usually require designers to specify more detail than they are supposed to know in the early stage of design (Herbert, 1993). Most designers prefer to sketch their early design ideas and schemes on paper with a pen (Lawson, 1994).

Another drawback of currently available design tools is that they all propose to help solve design problems with their particular approach. Design needs different support at different times. “To the man with a hammer, the whole world is a nail.” For example, Case-Based Reasoning researchers (Kolodner, 1993; Oxman, 1993b) see everything about design as indexing and retrieving cases. On the other hand, Shape Grammarians see all design as generation by production rules (Flemming, 1987b; Knight, 1981; Stiny, 1985).

[5] But if the tools are useful, designers will do this anyway (e.g., AutoCAD and FormZ). However, designers deserve tools that are easier to use and support conceptual design process when sketching is the main activity.

However, design involves different concerns and incremental formulation and therefore call for different kinds of support at different phases.

Designers draw to design. They need tools that are seamlessly integrated with the environment they prefer to design in. This designer–preferred environment, in the early stages of design, is usually a freehand drawing environment. No matter how useful these knowledge-based design systems are, they all lack integration into a real design environment where free hand drawing is used. These systems also depend on designers to choose the right computational tools at the right time. The use of these tools, when not embedded in a design environment, can obstruct the flow of a design.[6] To make these design tools really useful, a drawing environment that supports recognition of the task at hand should be built. A computer system should support freehand drawing, recognize design intent from drawings, and activate the right tool for the right time based on the design intentions and the drawing context.

This dissertation views designing as a multifaceted activity in which no single tool can serve all tasks. The RTRT (right tool at the right time) architecture was implemented to provide an interface to a variety of knowledge-based design tools. Rather than asking the designer to select and apply different computational tools to perform specific tasks (e.g., ‘now I want to look at a case library’) RTRT explores the possibility of automatically (or semi-automatically) invoking various design tools based on a design drawing. The marks designers make reflect the task at hand. For example, an architect often draws sight lines and a view point when working on lighting and visual access, and force diagrams when considering structural stability. The “right tool at the right time” system (described in

[6] A usability study of Archie discussed in Chapter 3 revealed this finding.

Chapter 6) attempts to infer from the drawing what the user is working on, and uses this inference to provide a tool appropriate to the task at hand.

1.4. GOALS OF THE THESIS

This thesis examines how design contexts and intentions are embedded in design drawings. The inference of context and intention from a design drawing can help reveal the relationship of design sketching to design thinking, and enable the development of a digital sketching environment that invokes knowledge-based design tools at the appropriate time.

The first goal of the thesis is to determine whether, and to what extent, it is possible to infer, interpret, or guess what a designer is thinking about by looking at her drawing. The proposition is that the designer’s intent could be indicated by the drawing symbols and their spatial relations. This dissertation does not propose to explore all possible ways that a drawing, or the process of drawing can yield clues to intention. It focuses on some drawing features such as symbols and configurations designers use in different design contexts. Specifically, in this thesis an architect’s office space design was used as an example domain in experiments for finding the graphical vocabulary for design tasks involving lighting, space arrangement and visual access. The guess is that the designer’s attention to, and interest in, these various tasks can be determined by examining the drawing symbols and their spatial arrangement.

If it is possible to associate symbols and spatial arrangements in the drawing with the designer’s intention, or task context, then the second goal is to propose how a computer might be programmed to recognize these drawing conventions. In other words: can these design drawings be made computable? The last chapter of the dissertation describes the implementation of a Right-Tool-Right-Time (RTRT) prototype system that tries to invoke appropriate design tools at the right time by interpreting a freehand drawing to determine which tool is needed. The "right tools" are proposed based on finding from the empirical studies of design intentions and drawing symbols.

1.5. ASSUMPTIONS AND HYPOTHESIS

The hypothesis is that designers consistently draw certain symbols and configurations when they are thinking about certain aspects of their design problems. These drawing conventions,[7] which include symbols and the arrangements of symbols in a design drawing, serve as a representation for designers to communicate with each other, or perhaps even as a language to think about design problems. Furthermore, these conventions may include a consistent mapping of symbols onto intentions in their associated contexts. If this hypothesis is true, then a computer can be programmed to infer the designer's intentions from the drawing and, based on this inference, suggest appropriate computational tools for the task at hand.

This research makes use of several assumptions. The first assumption is that designers engage in various activities involving information in the early, conceptual phase of design.[8] Design being an information intensive activity, the second assumption is that

---

[7] Conventions may come from a variety of sources such as common approaches to education (Laseau, 1991), the media or the nature of drawing itself; however exploring the sources of conventions is beyond the scope of this thesis.

[8] The observation of designers engaged in a courthouse design described in Section 3.2.3 yields evidence that designers look for information in their early design process.

knowledge-based design tools would be helpful to designers.[9] The third assumption is that designers sketch different symbols and configurations when thinking about different design concerns.[10] Given the third assumption, the fourth assumption is that these freehand drawing conventions can be programmed[11] into a computational drawing environment (i.e., Right-Tool-Right-Time) and be used to activate different design tools or to retrieve useful design information. There are, of course, two implicit assumptions: (1) people will come up with some conventions if they need to communicate a concept[12] and (2) it is possible to teach one's conventions to the system.

In addition to these assumptions, this dissertation takes the position that designers would find a system that works like the one described in the scenario (section 1.7) useful, and that it would enable designers to make better use of knowledge based design tools. No such systems have existed in the past, therefore there has been no way to test this assumption. The results of a usability study of the Archie tool described in section 3.2.3, and answers to the post-test questionnaire in the study described in section 4.4.6 suggest that designers would indeed find a system like RTRT useful. If, in the future, the RTRT approach is developed into a practical design system (as opposed to a research prototype) this assumption can be put to the test.

---

[9] This is an interesting assumption. Of course, many knowledge based systems are developed based on this assumption. However, to prove whether these knowledge based systems are useful or not for the designers is beyond the scope of this research.

[10] The evidence of this assumption is described in Chapters 4 and 5.

[11] Chapter 6 shows how these conventions can be programmed and used to activate different knowledge based design tools.

[12] This assumption relates to the question about where the conventions come from. One possibility is that people are coming up with the conventions out of the blue if they need to communicate. However, as one might suspect, (demonstrated in Chapter 2) that most of these conventions are in fact learned as part of professional training.

In order for RTRT (see section 1.7) to really work, a computer system needs to have several capabilities. First, the system should allow input of freehand drawing from designers. Then these drawing symbols and their spatial relationships should be recognized by the system. Having identified the drawing symbols and configurations, the system should further identify different drawing types (e.g., plan and section) to setup a context for the existence of certain symbols and configurations. Furthermore, the system should recognize the combinations of symbols, configurations and context that indicate particular design concerns and recognize them as intentions. Then, the system should be able to communicate with different design tools such that the inferred design intentions can be used to activate appropriate design tools. In another word, design tools would be activated when certain configurations and context are identified by the system.

In summary, to realize the scenario of Right-Tool-Right-Time, a system needs to 1) accept freehand drawing input, 2) recognize drawing symbols and configurations, 3) recognize drawing contexts, 4) infer design intentions from symbol configurations and contexts, 5) establish communication with other knowledge based design systems, and 6) use inferred design intentions to activate different design tools.

This dissertation investigates and explores issues 3 to 6. The Right-Tool-Right-Time prototype system is built on top of a freehand sketching program called the Electronic Cocktail Napkin (see Chapter 3) that handles the input and recognition of freehand drawing symbols and symbol configurations.

1.6. METHOD OF INVESTIGATION

This dissertation explores the questions outlined above in three major parts: through 1) case studies of related work, 2) empirical experiments, and 3) software implementation. (Figure 1-1)

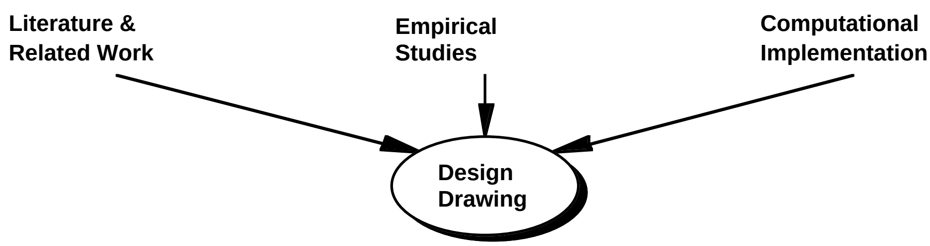


Figure 1-1. This dissertation is about design drawing. The discussion of a design drawing environment is facilitated through 1) survey of literature and related work, 2) empirical studies on design process, and 3) computational implementation.

The first part, which presents case studies, includes a survey of the literature on the use of drawing, design protocols and computing research on sketching. It briefly describes the role of drawing in design through the work of famous architects, pedagogical books on design, cognitive science studies of design representations, and protocol analyses of design process. From discussions of drawings in different design studies, one can identify various uses of diagrams and drawings in design thinking. The case studies include what designers said about their drawing and protocol analyses of design drawing. The design protocol studies show that previous research has not fully explored the role and use of graphic symbols in design drawing. The review of computer aided sketching program and the description of two systems, a case based design aid Archie and a freehand drawing environment called the Electronic Cocktail Napkin set up the background, motivation and development platform for this thesis that led to the Right Tool at the Right Time computer implementation.

A second way to understand the question of inferring intention from drawing is to conduct cognitive science style, empirical experiments. This second part of the dissertation presents a series of original empirical studies on design drawing and intentions conducted to examine the mapping of drawing symbols to design intention in their associated context. Chapter 4 describes pilot studies of diagram–making with material from an Archie case library. Chapter 5 describes a further investigation about drawing in the context of design, with particular focus on the relations between drawing conventions and design concerns.

The third part, design computing, demonstrates a possible consequence of part I & II. This part of the dissertation reports the implementation of the Right-Tool-Right-Time mechanism, an “intelligent” application manager that provides design tools targeting the task at hand by attempting to recognize intentions from design drawings. For example, when the machine detects that the designer is working on lighting concerns, it may provide relevant cases, specifications for lighting devices, or an interactive lighting simulation. The purpose of the software implementation is to demonstrate the practical consequences of the research.[13]

The Right-Tool-Right-Time scheme shows that if a designer can train the computer to recognize a set of drawing symbols, then the computer can retrieve information based on

[13] RTRT is only a prototype to demonstrate that it is possible to recognize design intention from drawing symbols and configurations, and based on the contextual intention, relevant design tools can be activated. It does not intend to claim a full implementation of all the findings from empirical studies. However, it shows the promise of a fully functional system to support various drawing activities.

the previously defined context of the symbols and the relevant databases. The designer can remain engaged in drawing instead of interrupting design activities to activate the tools.

The rest of the thesis is organized into six chapters. Chapter 2 describes the use of drawing in design activities based on a survey of the literature, and case studies of architects, design protocols and sketch systems. Chapter 3 describes antecedent projects and pilot implementations that serve as background information for the Right-Tool-Right-Time concept. Chapter 4 presents a preliminary study of drawing and design activities, the diagram–making experiment using stories from Archie case base. Chapter 5 describes a design experiment set up to explore the relations of drawing conventions to design intentions that other studies did not address. The experiment aims to show that different symbols are used in different design drawing contexts. This chapter discusses experiments and identifies symbols and relations for RTRT's implementation. Chapter 6 describes the Right-Tool-Right-Time prototype system implemented for inferring design intentions. It explains the phases of recognizing an intention: 1) symbol recognition, 2) configuration recognition, 3) context identification, and 4) intention inference. It also describes the communication protocols sent out by RTRT to different design tools once an intention is identified. Chapter 7 concludes with a summary of this dissertation, discussion of the findings and an outline of future research directions.

1.7. A SCENARIO

The following section discusses a scenario[14] of how a designer might use the Right-Tool-Right-Time. RTRT is portrayed as an intelligent design assistant named ‘Arty.’ The scenario presented here was initially designed to motivate the development of the RTRT system. In fact, the prototype system developed in the course of the dissertation research implements the functionality described here as a scenario. An overview of the implementation of this program is presented in Chapter 6.

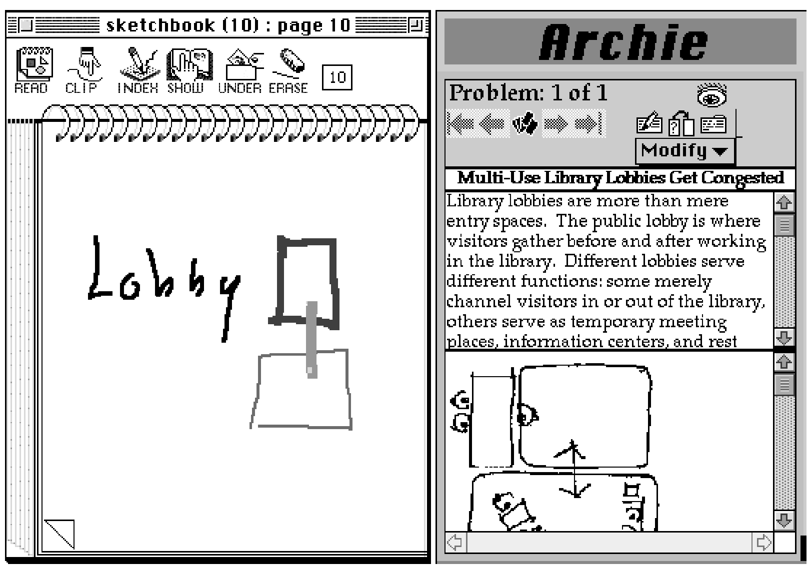


Figure 1-2. Designer’s diagram retrieves a relevant story from a case library, Archie. Left: Bubble diagram of spatial arrangements in a sketchbook. Right: Archie case story of a library lobby with similar configuration.

---

[14] Though this is a scenario describing how Right-Tool-Right-Time system might work in a design process, the figures shown here (Figure 1-1 to Figure 1-6) are all actual screen shots of the current working system.

A novice designer Lynn is designing a public building. Her assistant Arty (RT) sits quietly watching and ready to help. Lynn starts the design from scratch by making a bubble diagram to think about spatial arrangements. Reading Lynn's diagram (Figure 1-2a), Arty recognizes that she is working on the spatial relations of lobby and other spaces, and calls up a story from a case library, Archie, (Kolodner, 1991; Zimring, Do, Domeshek & Kolodner, 1995) that discusses lobby arrangements (Figure 1-2b).

Lynn finds the case provided by Arty useful, and she incorporates the arrangement into her design. Reminded by the Archie case story that a lobby can serve as an information center, Lynn adds another space bubble into the diagram.

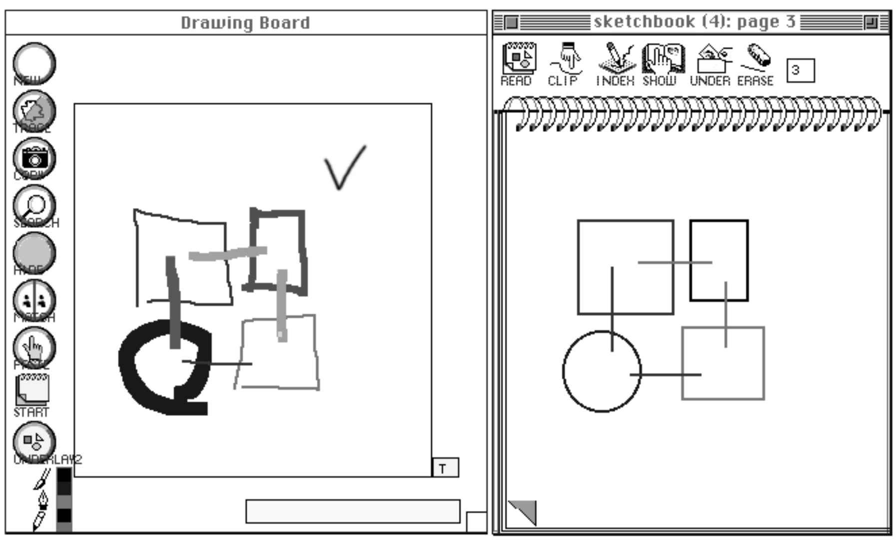


Figure 1-3. Designer's drawing can be cleaned up and rectified if desired. Left: Sketchy drawing of spatial arrangements. Right: Rectified geometric shapes of the sketchy drawing. Notice the check mark on the left was a gesture request for cleaning up the drawing and therefore was not translated on the right.

She continues to draw, adds another space, links them together, and draws a check mark gesture to indicate that she is pleased with the layout and wants the layout carefully redrawn. When she stops, Arty recognizes the meaning of the check mark and hands Lynn a hard line version of her sketches (Figure 1-3).

Meanwhile, Lynn draws sight lines from a viewpoint on a floor plan to get a sense of visual enclosure in her design. Arty calls up a visual simulation program, Isovist (Do, 1993; Do, 1995b; Do & Gross, 1997) to help her with this analysis (Figure 1-4).

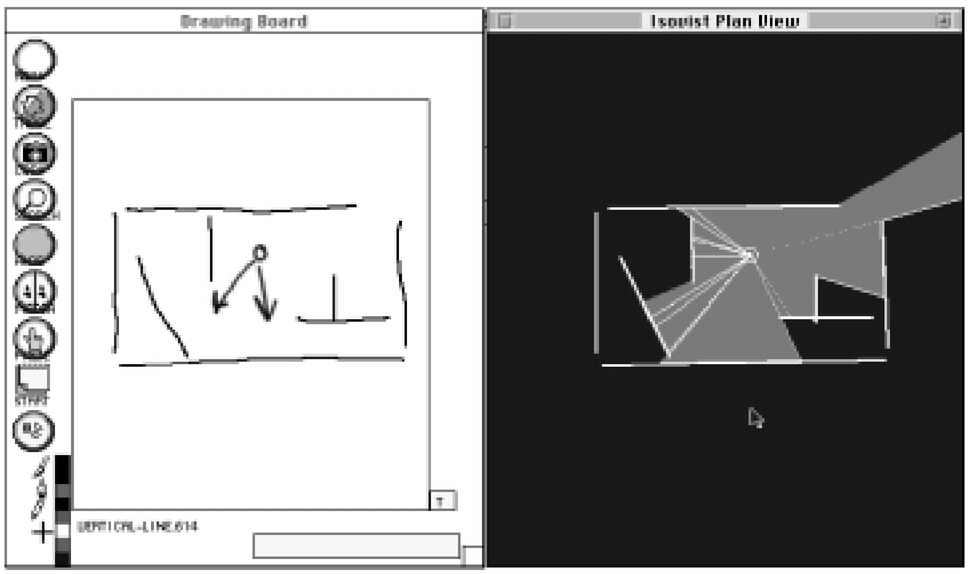


Figure 1-4. Designer’s drawing of visual enclosure brings up a viewshed simulation program, Isovist. Left: Diagram to explore visual access. Right: Isovist viewshed analysis tool

Satisfied with her basic spatial arrangement, Lynn decides to work on form. She sketches several cubes to examine the massing of her design. Based on Lynn’s sketch, Arty quickly builds three-dimensional models (Figure 1-5) so that Lynn can explore viewing perspectives and perhaps create a walkthrough animation.

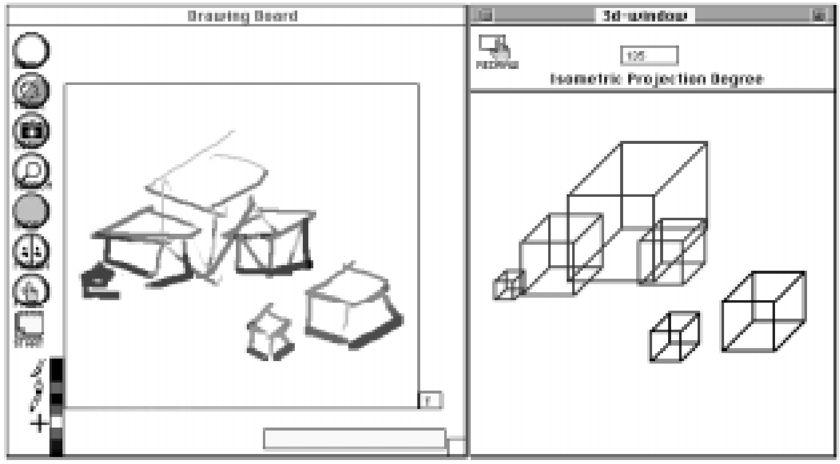


Figure 1-5. Designer's sketches is transformed into a 3D CAD model. Left: Sketches of massing study. Right: Translated 3D model that allows changes of isometric projection angles.

To make her building look more monumental (!), Lynn draws a facade composed of a triangle pediment and colonnade. Arty recognizes the drawing as a temple and finds for Lynn slides and a QuickTime animation of the Parthenon from a catalog of famous architecture (Figure 1-6).

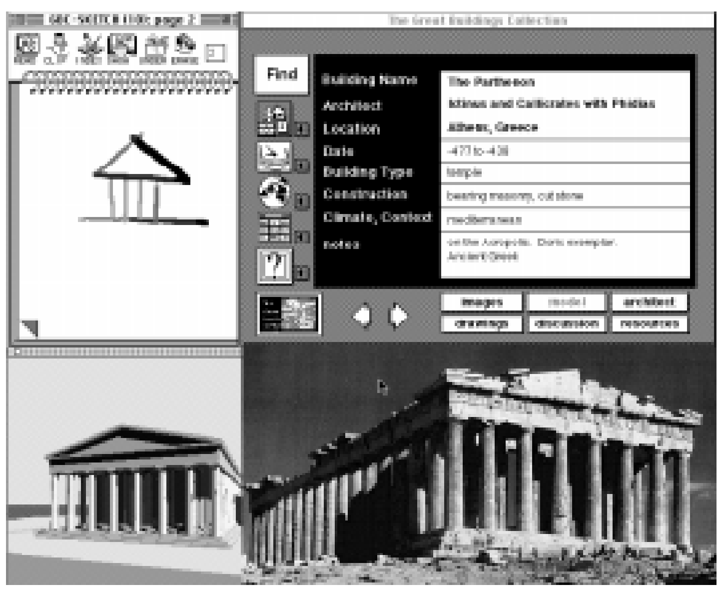


Figure 1-6. Designer's diagram retrieves information from a multi-media database. Top left: Diagram of a triangle pediment and colonnade. Top right to bottom left (clockwise): Textual information, picture and QuickTime animation of the Parthenon from the Great Buildings Collection (Matthews, 1994).

At this time, Lynn gets a call to solve an emergency. Lynn pulls out a floorplan, overlays it with tracing paper, and starts designing mechanical wiring. As Lynn works, her assistant Arty observes that Lynn is drawing HVAC devices and wires, and quietly finds a sketchbook page that has a similar configuration. Arty goes on finding product catalogues of thermostats from the World Wide Web (Figure 1-7).

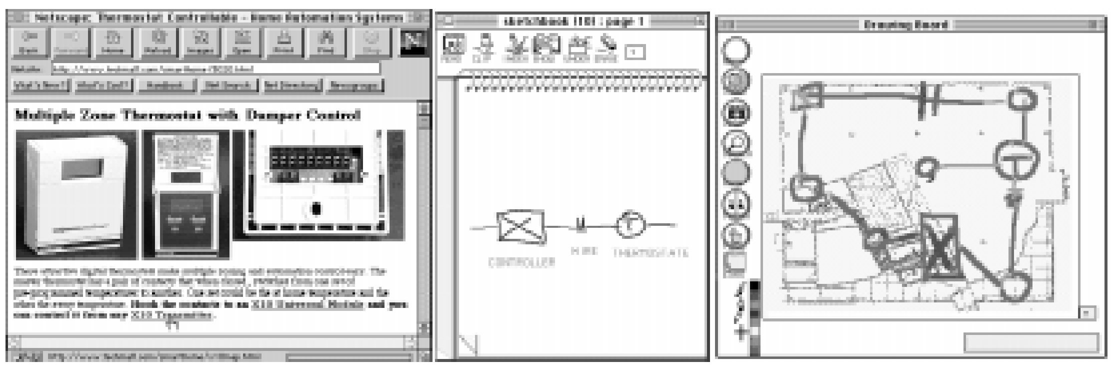


Figure 1-7. Designer's diagram retrieves information of HVAC production s from Netscape. Left: Retrieved page from the World Wide Web through Netscape program. Middle: Retrieved sketchbook of a note on HVAC configuration. Right: Designers diagrams on top of a floor plan.

And so the story goes.

Though different designers may approach design differently, all design processes have two things in common. The scenario described above shows a good example of these two aspects and how a design assistant can play a part in the design process. First, the designer engages in a variety of activities during design: finding references, functional analysis, modeling, and visualization. Designers use tools such as design case libraries, drafting and modeling, and visual analysis to support these activities. Her assistant Arty understands her design concerns and supplies her with useful tools.

Second, the designer makes various kinds of drawings as she works (e.g., check mark, sight lines, 3D cube, etc.). Her assistant Arty (who could be a computer-based agent) recognizes what the designer is doing by looking at these drawings and then suggests an appropriate tool.

This dissertation attempts to realize the above scenario. It demonstrates RTRT, a system architecture and a prototype application for a freehand drawing environment that can support design by making knowledge-based systems available at the right time based on design drawings. The interactive tool (Right-Tool-Right-Time, RTRT) supports design activities such as exploring ideas, finding references and images, and information retrieval, while allowing designers to sketch their ideas with a digitizing pen and tablet. The system allows rough drawings to be produced quickly, preserving the important properties of pencil and paper. However, unlike a paper sketch, this electronic sketching environment is interactive: it recognizes a designer's intention by looking at the drawing, and it can be trained or modified to fit personal design needs and different design contexts

Of course, one can question the possibility of making computers “experienced” or “intelligent” assistants for design. Researchers (Bijl, 1989; Winograd & Flores, 1986) have cautioned us the use of personification language of this kind (e.g., understand, expert, intelligent, knowledge, etc.) for computers might be misleading. However, this dissertation takes the approach that, in order to make computer programs work better for people, one might try to observe people more closely, to externalize knowledge, and therefore, to build more reasonable expectations for computers. The use of personification language throughout this dissertation is a way to convey this idea – if a human can understand what another human does, maybe a machine can be taught to do the same.

This dissertation takes the view that computers can be “taught” with human knowledge and chooses to personify computers to explain how computers can work for people. Therefore, “intelligent” means that a computer could have "knowledge.” “Knowledge” means organized data. “Understanding” is then “pattern recognition” of a predefined model of some sort (e.g., if–then rules). Schodek argues that design is a process of both acquiring and utilizing knowledge (Schodek, 1994). In terms of Schodek’s views, this thesis deciphers design so that computers can capture domain knowledge and support design representations from drawings.

CHAPTER II

RELATED WORK – DESIGN STUDIES, COGNITIVE STUDIES AND SKETCHING PROGRAMS

This part of the thesis seeks answers to three questions: 1) What is the role of freehand sketching and diagramming in design? 2) How can one study the reasoning processes of designers so as to further our understanding of sketching in design? and 3) What computer programs have already been built to support ‘sketching’ activities?

To answer these questions, this chapter reviews existing case studies related to design and sketching. The first part of this chapter looks at design studies that focus on the importance of drawing in design. The second part describes cognitive science and protocol analysis studies of the relationship between drawing and design thinking. Finally, the last part reviews relevant research of computational sketching programs. The purpose of the chapter is ultimately to establish a knowledge base and rationale for the empirical studies described in Chapters 4 and 5 and the system building described in Chapter 6.

2.1. DESIGN STUDIES

Drawing plays an important role in architectural design. Designers use the act of drawing to help them discover and explore ideas. They draw to think about design and to remind themselves of possible design alternatives. Although drawing styles may vary, many designers acknowledge the use of sketches as an integral part of their design process. The sections below review the design studies literature and report informal accounts given by designers of their own use of design drawings.

2.1.1. Studies Based on Interviews and Portfolio Reviews

Several recent design studies focus on the connection between design and drawing. They present case studies of famous architects based on interviews, observations, and analysis of their portfolios.

Lawson's "Design in Mind" presents interviews with ten famous architects and analyzes their design approaches in practice (Lawson, 1994). His book provides abundant examples of drawing, mostly freehand drawing, with occasional photographs of sites and models for comparison. Lawson reports interviews in which designers talk about their design process. These designers (his subjects) all stressed the importance of drawing. For example, Herman Hertzberger argued that drawing is crucial in his design process because it is a "communication of my brain and my paper" (p. 38). Santiago Calatrava described his sketches as a "discovering" journey of ideas. He called sketching a "dialogue" (p. 26) between what is on the paper and what is in his mind. Denise Scott Brown pointed out that though Robert Venturi's sketches are beautiful and expressive, they are never drawn "as works of art but as communication with self" (p. 98). Lawson concludes that the act of drawing plays an important role for the designers: "They find it hard to think without a pencil in their hand" (p 141).

Fraser and Henmi's "Envisioning Architecture" looks at how techniques used to make different drawing types influence the making of architecture (Fraser & Henmi, 1994). Their examples come from archives of architects' drawings, which at first sight seem to vary widely in type and style among different architects. However, upon closer inspection, several common shorthand notations can be identified, such as those for human figures, trees, movement directions, and dimensioning. For example, many architects (e.g., Le Corbusier, Böhm, Maybeck, Kahn, Stirling, Rudolph, et. al.) used a human figure in section and elevation drawings to serve as a scale indicator and many drew the symbol of sun and light rays (e.g., Maki, Scarpa, Predock). In their chapter on diagrams, Fraser and Henmi note that architects "symbolize . . . intangible factors such as movement, access, sound, view, function, and time" (p 110) in diagrammatic form to represent the abstraction and reduction of information. In their chapter on design drawings they quote Carlo Scarpa's reason for drawing – "I want to see therefore I draw" (p. 113) and argue that design drawings facilitate design evolution by carrying out "conversation" to study, find, and test ideas. They also argue that designers use "visionary drawing" as a means to expand and explore imagination in their design process.

Herbert's "Architectural Study Drawings" examine the graphical media and design processes of six practicing architects (Herbert, 1993). He argues that drawings are more than just a convenient strategy for solving design problems, suggesting that they are "the designer's principal means of thinking" (p 1). He argues that a designer "must interact with the drawing" (p 121). Among the drawing and diagrams selected from the interviews and archives, symbols for axial lines, views, entrances, and numerical calculations can be identified. Herbert argues that architectural graphic conventions[1] serve two functions: 1) they provide a basic framework for a designer to draw, and 2) they enable communication between different designers. Using a quote from the preface of "Architectural Graphics Standards" that remarked "those trained to grasp a drawing at a glance can find their desired information immediately," (p. 87) he argues that designers can read each other's drawings because they follow the same graphic conventions – not only in finished working drawings, but in study drawings as well. His interview with Stanley Tigerman illustrates that "drawings are a vehicle for interaction" and that architects use graphic conventions to communicate with each other (p. 96).

Robbins's "Why Architects Draw," like Herbert's book, examines the work of well-known professional architects (Robbins, 1994). However, Robbins focuses on the social role of drawings in architectural practice, using descriptions given by architect themselves of their own personal design processes. He quotes Renzo Piano, who says that drawing is a "pure instrument of a circular process between thinking and doing," (p 126) and Alvaro Siza who says that sketching "provoke(s) a change of ideas" (p 153) and helps "verify" things to see if they work (p 157). He concludes that drawing serves to support the

[1] Herbert used the term "conventions" in two ways. First, he referred to "five architectural conventions" as different kinds of "graphics projections" -- plan, elevation, section, perspective and axonometric drawing -- that designers use in construction drawing and study drawing. He observed that all drawings in the book "Architectural Graphic Standards" fall into one of these categories. Secondly, he also used the word "conventions" to mean "standardized ways of showing dimensions, notations, signs, or symbols" (p. 89). My use of the word "conventions" in this dissertation is closer to the second kind but broader. "Drawing conventions" means not only the symbols that designers draw but also the configuration of the drawing symbols. I report in Chapter 4 the conclusions from my drawing experiment that designers exhibit "drawing preference" or "view preference" of using plan versus sectional drawing to illustrate different design concepts.

designer's internal dialogue and critique. Drawing is a disclosure mechanism that expresses and tests an architect's intent.

These books all argue that drawing is necessary for design thinking. None of them discusses the kinds of graphic symbols designers use in design drawing. However, several drawing symbols consistently[2] appeared throughout the design drawings of various architects: those for trees, entrances, the sun, columns, view lines, compass direction (north), human figures, walls, doors, and dimensioning. One might look at these design drawings and guess the context and intention by identifying their key symbols and configurations.[3]

2.1.2. Designer's Self Reports

Many architects express the importance of diagrams and drawing in their design process. They describe their experiences using drawings to think about design, to explore and record ideas, and to communicate with others.

For example, Graves explains that his "referential sketch" serves the purpose of a "diary" or record of his observations and discoveries (Graves, 1977). He describes these sketches as "shorthand" notes of architectural ideas drawn to be remembered, refined and combined with other sketches in later compositions. He also says that he and his partners usually collaborate through a "conversation" by exchanging and adding to the drawings.

---

[2] Interestingly, these designers used similar diagrams and symbols even though they came from different countries -- UK., Spain, Holland, Czechoslovakia, and USA. Could it be that there might be an international convention in architectural drawing? Or could it be the symbols are in some way "natural?"

[3] It would be an interesting project to collect and classify the drawing conventions from the illustrations of these books.

He calls this exchange of drawing a game of idea exploration facilitated through a common understanding of a set of explicit “principles or conventions.” He further claims that drawing marks on paper are a “language” that can “play back to one’s mind and bring forth further elaboration” (p. 384). He argues that they are “speculative” and therefore plays an important role between designers’ mind and action.

Louis Kahn in “The Value and Aim in Sketching” mentions that sketches are as important to him as design problems (Kahn, 1931). He says “drawing is a mode of representation” (p. 10). Regardless of the medium used, the value of a drawing is in the “purpose” of making. He argues that designers need to interact and work with a sketch, not just “crystallize” thoughts on paper.

Many books on design also give examples of the importance of drawing. For example, Rowe’s “Design Thinking” describes the procedural and normative aspects of sketching in design thinking (Rowe, 1987) He examines how architects and planners use drawings to inquire about shapes and ideas of buildings and the use of public spaces. The freehand drawings in his book include diagrams with text annotations for entrance, lighting, and numeric calculations. As another example, Eisenman’s “House of Cards” documents his use of diagrams and drawings for a series of house design projects (Eisenman, 1987). In the preface he states that the book is a “record of evolution” of his design thinking. The drawings vary from extremely unstructured and informal sketches to rigorous and definitive hard-line technical drawing. Similarly, Birkerts in his “Process and Expression in Architectural Form,” shows, like Eisenman, examples of his many different design projects with drawings from various stages to illustrate the development of a design (Birkerts, 1994).

These books and the architects' portfolios themselves illustrate the importance of freehand drawing in design. They insist that freehand drawing is an integral part of the design process and cannot be neglected.

One can identify several consistent ways of expressing design concerns through the use of symbols in drawings. Many architects explain in the text what the drawing is about and many include hand–written annotations. For example, a human figure in a sectional view is usually a circle above a triangle, often with arrows shooting out from the head accompanied by the word "view." A sun symbol appears in many drawings when designers mention or explore lighting, often with lines starting from the sun and penetrating the building envelope. Architectural details such as walls, windows and doors and furniture such as tables, chairs, beds and sinks appear in plan view as simple lines and shapes.

Many drawings from these books use lines and numbers to show dimensions, human figures to indicate scale, the letter "N" and an arrow for the direction of North, numerical calculations for space requirements or material budgeting, and simple shapes for space. However, none of the researchers or designers mentioned above specifically refer to the use or meaning of graphic symbols in the drawing.

2.2. COGNITIVE STUDIES

2.2.1. Drawing as External Representation

Several cognitive scientists have argued that drawing is important because it is an external representation that helps in solving problems and generating ideas. They argue that drawing facilitates thinking and supports emergent ideas.

For example, Larkin and Simon's "Why a Picture is (Sometimes) Worth Ten Thousand Words" argue that a diagram is a representation created to externalize and visualize problems (Larkin & Simon, 1987). They view diagrammatic reasoning as information processing. They argued that drawing helps people to recognize information, to see problems and to find solutions. For example, a diagram for a pulley helps one to see the distribution of pulling forces. People draw diagram to utilize spatial metaphors in the process of solving problems. Larkin and Simon conclude that diagrams help grouping information together with spatial locations and support perceptual inferences.

Blackwell's "Diagrams about Thoughts about Thoughts about Diagrams" (Blackwell, 1997) describes the views from experimental psychology literature that diagrams can be seen as a notation system (e.g., Goodman 1969; Bertin 1981; Ittelson 1996). He describes diagrams as practical artifacts created to 'resemble,' to make 'metaphors,' and to 'frame problem.' He argues that diagrams are important because they provide information and intention in a visual form through simple shapes and their topology.

Suwa and Tversky report that architectural drawing is important in that drawings visualize a designer's thinking and facilitate problem solving and creative efforts (Suwa & Tversky, 1996b). They observed design sessions and reported the self reflections of designers. They argue that designers draw to externalize their concepts and that drawings provide visual cues for revision and refinement of ideas.

Verstijnen, in "Sketches of Creative Discovery" argued that 'idea-sketches' are important in the early stages of design (Verstijnen, 1997). In particular, these sketches are important in the creative process because they function as a means by which the designers can interact with their mental imagery. She further proposes that the act of sketching helps deliver visual information and serves the function of both analysis and synthesis. Through a series of experiments, she argues that those who perform tasks with the aid of sketches achieve 'higher transformations' of their ideas, solve and find more problems than those who did not use sketches. She concludes that sketching is an externalization that can enhance designer's creative mental imagery.

Fish argues that in order to make computer systems that really work for design, one should have a theory of sketching first (Fish, 1996; Fish & Scrivener, 1990). In "How Sketches Work" he reviews the literature from cognitive science, art history and design to argue that sketches are representations of "visual thought" that help facilitate perception and translation of ideas. Furthermore, sketches help designers attend to thought and help trigger short term memory. He also argues that sketch supports indeterminacy such as generality, ambiguity and vagueness for implications of possible meanings, and is a controlled and skilled act to stimulate complex and infinite mapping to visual alternatives.

Mezughi argues that sketching is "the principal means of visualising design solutions and crystallizing the thinking process" (Mezughi 1996). His "Integral Role of Drawing in Architectural Conception" investigates the relations between design and sketching through a series of interviews, surveys, protocol studies and retrospective data analysis.

Goel, in "Sketches of Thought" argues that drawings[4] are important for designers because they are used as 'external symbol systems' to represent real world artifacts which can be manipulated and reasoned with (Goel, 1995). He argues that different symbol systems correlate to different design problem solving activities and phases.

Van Sommers's "Drawing and Cognition" describes experimental studies of graphic production processes (Van Sommers, 1984). He argues that the "charms of doodling," whether simplistic or complex, lie in the fact that it is a graphic play that represents an accumulation of actions. He further argued that the act of drawing is a "graphic engine or a production system" that helps people generate concepts.

2.2.2. Relevant Research on Design Protocols

Many researchers have performed protocol studies of designers to understand problem solving in design. In the following sections, the definition of protocol analysis is first given, followed by a summary of the contributions by many researchers who have used this technique. It then concludes by listing the implications of these studies. This section sets the background to the later chapters (4 and 5) on empirical studies.

[4] Goel also argues that sketchy quality adds value to the reading of the drawing representations. The sketchy quality of the drawing represents the ambiguous state of the thinking.

(1) What is Protocol Analysis

Cognitive psychologists Ericsson and Simon published a book “Protocol Analysis: Verbal Reports as Data” to describe a technique of studying the extraction of knowledge (i.e., understanding the knowledge and reasoning used during problem solving) from verbal reports (Ericsson & Simon, 1984). Newell used the technique of protocol analysis to study human information processing (problem solving) mechanisms (Newell, 1968). He applied protocol analysis to subjects who were in the process of solving problems. The subjects worked on a real or simulated problem situation and were encouraged to “think aloud.” They were asked not to rationalize or justify their decisions but to directly report their actual moves (or strategies and goals) when attempting to solve the problems. The record of the verbalization is a protocol, and the analysis and notation of the protocol is therefore called protocol analysis.

Protocol analysis is used to obtain information about people’s individual problem solving strategies and data collecting methods. If one could make inferences about what is going on inside people’s heads based on their step-by-step performance of solving a problem, maybe that understanding could be used to construct a problem–solving system (computer). In sum, protocol analysis, or the analysis of verbal protocols, is used to externalize a subject’s internal problem–solving strategies.

(2) Critiques of Protocol Analysis

The use of protocol analysis in cognitive psychology has increased in recent years, suggesting that this method of knowledge acquisition is gaining credibility. Recent protocol studies of design include not only verbal reports but also other kinds of data produced by the subjects during problem solving, in particular, design drawings. However, one remains concerned that the use of protocol analysis might not truly or fully represent the information processing involved in design process .

Two major issues have been raised about using protocol analysis to understand a design problem. First, a design protocol can only reflect part of the real design process, it cannot capture everything. Ericsson and Simon pointed out that the accuracy of the verbalization in a protocol might be task–specific (Ericsson & Simon, 1984). If the task is responsive, involving only short term memory, the verbal report would be closer to the real mental process. On the contrary, if a task is more retrospective, involving the use of long term memory, then wrong or missing data might occur. Therefore, a de-briefing interview or post-protocol questionnaire can be constructed after the protocol to reduce lower erroneous readings.

A second concern about protocol analysis is that the think-aloud protocol might distort the real design process. As Lloyd, Lawson and Scott pointed out the methods of protocol analysis might interfere with the act of designing (Lloyd, Lawson & Scott, 1995). Real design is usually "considered," designers have time to digest the design brief or architectural program. Designers would not normally be forced to work out a design in the artificially short period set up by a protocol analysis section. A real design process would be in a real setting (e.g., in a studio, using a drafting table) instead of in isolation in a laboratory. An alternative to protocol analysis is "discourse analysis," which suggests using transcripts of actual interactions involving domain experts and their clients in a real setting (Belkin, Brooks & Daniels, 1987). There are also concerns about how the verbal protocols might impair visual reasoning. Cognitive Psychologists Schooler and Engstler-

Schooler conducted experiments that show verbal reasoning interferes with visual reasoning in visual memory tests (Schooler & Engstler-Schooler, 1990). Similarly, Wilson in his experiments shows that people often misstates what they are thinking about in talk-aloud protocol studies (Wilson, 1994). These studies present arguments that verbal protocol studies could obstruct the reasoning process. However, it also supports the argument of this research that designers should not have to propositionize their intentions in order to retrieve needed information.

Though protocol analysis might not represent internal problem solving strategies completely, it can still yield valuable information about a designer's internal thought process and thus might help further our understand of design.

(3) Protocol Studies of Design Activity

Cross, et. al. suggested that protocol analysis is a useful research technique for analyzing design activity (Cross, Christiaans & Dorst, 1996; Dorst & Cross, 1995). In 1994, they reported an international research workshop on protocol analysis in design held in Delft. The twenty papers presented in the workshop presented numerous analyses of the same protocol data drawn from two design sessions (one of a team of designers and the other of a single designer, both working on fitting a backpack to a bicycle). The views ranged from identifying different modes in the design activity, to how knowledge and actions are embedded in design. Though the approaches vary widely, the workshop brought together many research technique of design activity.

This section describes several design research studies that use protocol analysis. All these studies collected both verbal and visual data. Eastman (Eastman, 1968) and Akin (Akin, 1986) conducted protocol analyses of designers in action and proposed that design representations are related to design problem solving and design thinking. Goldschmidt (Goldschmidt, 1989) and Schön (Schön, 1988) use protocol analyses to propose design reasoning as visual perception, looking at the acts of “seeing” and their functions in design. Suwa and Tversky (Suwa & Tversky, 1996b) argue that seeing drawing marks helps architects to refine their design ideas. These studies are outlined below.

Eastman in his “Analysis of Intuitive Design Processes” engaged six subjects in a simple task of improving a bathroom layout (Eastman, 1968). He collected and analyzed the protocols and annotated them as problems and behavior. Eastman views the design of a physical environment as a problem solving task in an information process model following Newell and Simon (Newell & Simon, 1963). In his protocol study, he documents the design operations used, the objects manipulated during design and the control mechanisms employed by the designers. He then developed a model to account for the behaviors observed in the study. The model portrayed design process as a process of identifying the design problems and testing design alternatives. Eastman’s study showed that different types of representation such as words and drawings done by designers correlate with the problems they find and solve (Eastman, 1968).

Akin’s “Psychology of Design” also follows the theory of viewing human problem solving as information processing (Newell & Simon, 1972). He conducted design protocol studies of architects sketching in order to analyze their chunking of design actions and their attention shifts (Akin, 1986). His experiment on recall looks at the time interval between the drawing of lines and the identification of groups of architectural elements in memory.

His examples revealed several chunks: the wall and window segments, steps, furniture of similar size that are closely located together. However, he did not identify the symbols or the configurations that were used by designers when they performed the recall tasks through drawing.

In a recent study, Akin and Lin observe that previous protocol research mostly emphasized recorded verbalizations (Akin & Lin, 1995). They note that little has been written about the role of drawings produced in the protocols although drawing is essential in the design process. They discuss symbolic encoding of different modes such as drawing, thinking, examining and speaking. They designed an experiment with two parts: 1) subjects were asked to reproduce a drawing from a printed transcript, and to 2) to predict the verbal data from a video of the design drawing process that has no sound track. They point out that novel design decisions usually occurred when the designer was in a "triple mode period": drawing, thinking, and examining. They also conclude that the transcripts and drawings echo each other.

Schön analyzed protocols of practicing architects to investigate design reasoning. He argues that design reasoning is a pattern of the use of design rules (Schön, 1988). His protocol experiment asked designers to make guidelines for selecting the entrance of a branch library. He argues that the data from the protocols showed that design rules are derived from types, and may be "subjected to test and criticism" (p. 183). He argues that designers frame a design problem, "set its boundaries, select particular things and relations for attention, and impose on the situation a coherence that guides subsequent moves" (p. 182). Schön, in "The Design Studio," (Schön, 1985) uses protocols to illustrate the idea of "reflection-in-action." He argues that designers "see" and then "move" the manipulated design objects. He further describes the kinds of seeing and their functions as 1) literal visual apprehension of marks on a page, 2) appreciative judgments of quality, and 3) apprehension of spatial gestalts (Schön & Wiggins, 1992).

Goldschmidt takes a different approach to using protocol studies. Like Akin, she believes design protocol should include not only verbalization but also drawing. Furthermore, her study does not view the protocol as merely a problem solving process; instead, she argued that a designer interacts with a drawing with “seeing as” and “seeing that” reasoning modalities[5] (Goldschmidt, 1989). In “Dialectics of Sketching,” Goldschmidt further proposes the use of sketching activities as visual thinking and imagery as a conceptual framework for investigation (Goldschmidt, 1991). She views sketching as an operation of design moves and arguments, an “oscillation of arguments” that brings about a gradual transformation of images. Recognizing the process of sketching as a systematic dialectic between the “seeing as” and “seeing that” modalities, she plots the “as-that” modalities with notation on sketching acts for various design protocols (p. 131). She concludes that the systematic structure of design reasoning is a “ping-pong” process.

Suwa and Tversky claim that seeing different types of information in sketches drives the refinement of design ideas (Suwa & Tversky, 1996b). They video taped architects sketching a design for an art museum. While watching the tape, the participants then reported what they had been thinking about as they drew. Suwa and Tversky classified the information in the protocols into different categories such as spaces, things, shapes, views, lights and circulation. They then proposed that a computational tool should “superimpose stimuli” on the design sketches to stimulate design thinking.[6]

[5] Goldschmidt’s modalities further identifies the type of reasoning involves in Schön’s “see” and “move” mode. Designer takes different moves after “seeing as” or “seeing that.” For example, seeing the circle as a pond and seeing that the circle is too big are two different modes of thinking.

[6] Though their paper was titled “What Architects See in Their Sketches: Implications for Design Tools,” they did not implement a system nor propose what the “stimuli” should be. In a later paper (Suwa &

Ullman, Wood, and Craig argue the importance of drawing in design process through protocol studies of five mechanical design engineers of varying background and experience (Ullman, Wood & Craig, 1990). Each design session took six to ten hours, and involved industrial design problems such as designing a plastic envelope, electrical contacts for batteries, and a mechanism for coating objects with chemicals by dipping and flipping them. Verbal protocols, drawings and gestures were video taped and transcribed. They concluded that each marking action is an external representation of a chunk of information. They classified all the “marks-on-paper” into different types. “Draw” marks include freehand “sketch” and “draft” with mechanical devices such as a straight edge and a ruler. “Support” marks include “text,” “dimension,” and “calculate.” Furthermore, they propose that the “marks-on-paper” can be classified according to whether their purpose is to “add,” “patch,” “refine,” or “recall information.”

2.2.3. Discussion

All the above protocol analyses agree that design drawing is associated with design thinking and can be interpreted through verbal descriptions. For example, Eastman's design protocol shows that both words and drawings are design representations and they correlate with the design problems (Eastman, 1968). Yet he did not identify the mapping between drawing and design problems. Akin’s study reveals that architectural elements (e.g., walls, windows and furniture) that are drawn spatially close are likely to form chunks in a design recall task (Akin, 1986). However, he did not identify the drawing

Tversky, 1996a) they described their vision to “animate sketched elements by giving them fluctuating movements according to certain principles.”

symbols or their configurations. Schön argues that designers “see” and “move” the design objects they draw to reason about design (Schön, 1985), but he does not identify the relations between the drawing objects and the design reasoning. Goldschmidt argues that “seeing as and seeing that” of a design drawing is an “oscillation of arguments” (Goldschmidt, 1989). Nevertheless, she did not identify the relationship between the arguments and the design drawing. Suwa and Tversky collected “reflected” protocols on architects and found that seeing information from the sketches helped the designers to refine their ideas (Suwa & Tversky, 1996b). They classified the information into different categories (space, shapes, things, views, lights) but did not identify the relations between the information types and the drawing symbols designers made.

In summary, the design protocols reviewed here acknowledge that design drawings are deeply connected to verbal protocols and design thinking. They discuss several important issues about design drawings: first, they identify that designers use freehand drawings when thinking about design concerns; second, they conclude that design reasoning is related to design drawings; and third, they suggest that different types of information are embedded in design drawings. This conclusion implies that a design drawing may employ different symbols to represent different types of information. However, none of the studies identify the graphic symbols designers use in design. They mainly look at the verbal descriptions of design problems and solutions, or the state shift or chunking of the thinking. They use their protocols to argue that design process involves the act of drawing and thinking. Therefore, this dissertation takes on the tasks of 1) investigating freehand sketching, 2) verifying the relationship between design drawing symbols and design intention, and 3) identifying the drawing symbol universes that designers use.

2.3. COMPUTATIONAL SUPPORT OF SKETCHING

With the observation that drawing plays an important role in design thinking and reasoning, the question then is whether computational design media can support design better, and if so, how? Can a sketching program provide an enhanced environment for design? What sketching programs have been built to support design activities and what are their strengths and weakness? This section first briefly describes related work on computational tools that aim to support sketching and then further discusses why it might pay to support sketching with computers.

2.3.1. "Sketch" Systems

Several researchers have used the term "sketch" in naming their design support systems. They generally use the word "sketch" to advocate the idea of drawing with a pen or having an easy to use interface, though, in many cases, they only support hard line drawing (strictly speaking, these hard line drawing or palette input systems should not be called "sketch" systems). However, many interesting ideas such as using constraints in a drawing environment (e.g., Sketchpad (Sutherland, 1963), SKETCH (Zeleznik, Herndon & Hughes, 1996)), interpreting sketches to be straight lines (e.g., STRAIT (Taggart, 1975), SketchIT (Stahovich, 1996)) have been explored in these system building efforts. The following discusses several systems that either have "sketch" in the naming of their systems, can convert sketches into straight lines, or are able to support freehand drawing inputs.

(1) Systems with ‘Sketch’ in Their Names

One of the first computer based drawing system is Ivan Sutherland’s Sketchpad (Sutherland, 1963). Sketchpad developed several important interface concepts that are common in today’s computer systems (e.g., constraints, copying and editing, grouping). The Sketchpad system enabled users to draw primitive graphic objects such as points, lines and circular arcs with a pen. Users drew directly on a graphic display with a light pen, and issued specific “constraint” commands with a set of push buttons and toggle switches to operate on the graphic objects. For example, user could draw a straight line by instructing Sketchpad to connect two points. To draw a hexagon (p. 14-17), one could start with an arbitrary six sided polygon, ask the Sketchpad program to move all points of the polygon to lie on a circle, and constrain the polygon to have equal length sides. The Sketchpad system maintained constraints such as a structure of a pattern, equal length, or line alignment. Sutherland claimed that Sketchpad provided a new man-machine interface by eliminating typed commands “in favor of line drawings.”

Zeleznik, Herndon and Hughes’s more recent SKETCH project explored the idea of “sketching” gestures as an interface for three dimensional geometric modeling (Zeleznik et al., 1996). SKETCH was designed to allow users to specify different modes of input in a 3D scene by using a three button mouse to draw gesture commands. For example, to construct a cube, user can click and drag three lines along the x, y, z axes to specify the dimension. To make a freehand curve, user can click with shift key and then draw. By using the third mouse button combined with click, pause, shift and drag, user can directly manipulate the zooming and panning of the camera. User can edit a volume by “oversketching” (i.e., in the editing mode) to reduce the height, or “make an opening” of an object. The user can also apply constraints to objects for “hinged” rotation, and to copy and group objects.

Stahovich’s SketchIT program (Stahovich, 1996) reads a mechanical “sketch” and transforms it into multiple interpretations of workable designs in the domain of springs and kinematics joints. He used the word “sketch” to represent the kind of mechanical drawing that if taken literally, might not work properly. The “sketcher” in the SketchIT project does not support freehand sketches. Instead, it provides a tool palette with objects such as face, pivot and slider. For example, to design a circuit breaker, a user can add a hook, spring, push rod and stop from the palette to the “sketcher” to illustrate a structure diagram. SketchIT first analyzes the geometry and generalizes a design representation in a “qualitative configuration space” (QC-Space). Then SketchIT compares the behavior of the parts through a library of motion types and interaction and generates several working designs.

(2) Systems that Convert ‘Sketches’ to Objects

James Taggart (Taggart, 1975) argue that sketch recognition is the way to “communicate architectural intentions and ideas” with a computer because designers will enjoy the comfort of “familiar (pencil) tool” instead of “input protocol” demanded by the systems. His HUNCH system has two modules: Draw and Show, and STRAIT and STRAIN. Draw is used to record and save input data from a light pen, while Show is used to redisplay the data with the recorded information such as sampled points of the pen path and drawing speed. STRAIT is used to “infer” intended corners by comparing changes of direction between adjacent line segments to “latch” close points (reducing the amount of data to be stored) in order to extend straight lines. STRAIN (STRAIghten with No latching) is a variation of STRAIT that avoids over-enthusiastic latching of corners. Besides this “local” approach of storing sketch data as a structure of endpoints and lines, HUNCH could also map the sketch to a grid (array) in which filled cells indicate the line path. The gird can be examined with varied resolution (e.g., 10x10, 200 x 200). HUNCH was good at interpreting certain types of sketches (e.g., it correctly inferred sharp corners from round ones) but was poor at interpreting others (e.g., it inferred false closure from extensions of an arc that was meant to be open).

Sivaloganathan’s “Sketching Input for Computer Aided Engineering” describes a system that uses isometric freehand sketching as input to solid modeling (Sivaloganathan, 1991). The designer first sketches an isometric (e.g., x and y axes at 30 degree to the horizontal while the z-axis remains vertical) view of a three-dimensional solid block. The system “Sketch-Solid” provides an isometric grid as a sketch area and a menu offers items for sketch registration: “visible line,” “hidden line,” “centre line,” “construction line,” “erase line” and “redundant line.” Sketch-Solid first reads the endpoint coordinates of the sketched lines, then maps them to geometric objects such as straight lines, circles, ellipses and arcs. The sketches first undergo two dimensional processing to merge end points, and to find equations of line segments, vertices, and edges. The data then go through three-dimensional processing to produce three-dimensional vertices, construction lines (e.g., a rectangle to “box-in” and guide the drawing of a “circle” inside) and edges.

Baudel argues that designers and artists prefer to redraw sketches to modify them and therefore he proposed a “mark-based interaction” technique for editing free-hand drawings (Baudel, 1994). Instead of using the current computer graphics convention of editing a curve with control points and tangents, Baudel’s technique allows users to edit a drawing by adding new gesture strokes as editing commands. The system “reparametrizes” the curve to generate a new smooth curve in place of the original one. There are four modes of interaction. In "create" mode, any stroke the designer draws is a new curve. In "edit" mode, the new stroke acts as curvature specifications to correct the adjacent original stroke, and a new modified (e.g., rounder or flatter) curve is generated. The "delete" command gesture applied to the tail of a curve will erase that portion of the stroke. A smooth oval shape can be obtained by adding a line connecting two end points of a drawn letter "C" in a "join" mode.

(3) 'Sketch' Systems that Support Sketching Activities

Recent advances in digitizer and pen computing technologies have induced research in computational environments for freehand sketching. These systems used pen-based interfaces as a way to input and edit design drawings. Unlike the "sketch" projects described above that use the word "sketch" to mean an easier interface for drawing or the ability to translate sketches into structured drawings, the systems described below supported the display and manipulation of freehand sketches as they are entered.

Saund and Moran's PerSketch program (Perceptually Supported Sketch Editor) presents an image processing approach to support extracting visual images from freehand sketches as would be perceived by people (Saund & Moran, 1994). They argued that the perception of the structure of a sketch is an emergent, dynamic and interactive process and that users should thus be able to "walk up and draw" instead of dealing with menus. They explore the idea of manipulating "digital-ink in an Electronic Whiteboard or Electronic Sketchpad application." For example, when a user draws a circle that overlaps its diameter on the edge of a rectangle, PerSketch's analysis routine creates multiple readings of possible shapes (e.g., segments of half circle arcs, a rectangle with one round end, etc.).

The user can then gesture to indicate an intended selection of a visual object by drawing a rough trace stroke whose bounding box is equivalent to the object.

Kramer's "translucent patches" and "Architect's Electronic Sketchboard" argue that irregular shapes are an important feature in a sketchbook and therefore should be employed as infrastructure for information selection, presentation and manipulation (Kramer 1994; Genau and Kramer 1995; Kramer 1995). For example, a designer meeting with a client might use a freehand circle to select a certain area on a floorplan for discussion and to bring attention to that particular space. She may then draw a quick section of the space to illustrate how it might work. Kramer's Sketchboard would create patches for these drawings and link them together. The designer might write down a list of material with estimated prices to discuss with her client. After she draws double lines to request a calculator, the patches will add the numbers and print the total sum below the double bar. The client might decide to eliminate one item by circling it and applying an erase gesture. The patch will then eliminate that item from the list and recalculate the sum. The user can also draw gesture commands to move and edit a patch (delete, clear content, grow bigger, shrink or dissolve) with other patches. Freehand sketches and handwriting are all preserved in the translucent patches and can be manipulated.

Landay's SILK project – (Sketching Interfaces Like Krazy) – is an interactive interface builder that allows one to design and test interfaces by sketching them (Landay, 1996; Landay & Myers, 1995). Interface designers can use SILK to sketch and mockup an interface with a limited set of interface components and to explore the look and behavior of an interface. SILK provides gesture commands such as delete, move, copy and group as well as a predefined library of interface components (widgets) in sketch form. SILK uses a single stroke recognition algorithm (Rubine, 1991) to recognize objects. The system recognizes a wiggly line as a “text” item, a circle as a “radio” button, a long rectangle as a “slider bar” and a small rectangle as an “elevator” in a “slider bar.” Once the objects are recognized, interaction can be “played”: the elevator can be dragged up and down along the slider bar, a radio button can be pressed in a “run” mode. When designer is satisfied with the design, the SILK sketch can be replaced with real widgets and graphics objects. The SILK sketch interface was built to support easy iteration and evaluation of an interface prototype in the early stages of design before time consuming programming takes place to build the interface with the real graphic components (buttons, slider, text, etc.).

In conclusion, these “sketch” systems support different modes of design drawing. The first kind support structured drawing. The drawings are presented as clean and rectified objects – no sketchy lines, either as input or output, are involved. For example, with Sketchpad (Sutherland, 1963), designers use a light pen to digitize primitive hard-lined objects such as points and lines; with SKETCH (Zeleznik et al., 1996), users input a three dimension model by drawing three straight line “gestures”; and with SketchIT (Stahovich, 1996), users select objects from a tool palette, such as pivots and sliders, and place them in a mechanical drawing.

The second group of “sketch” systems take sketchy, rough drawings and convert them into straightened objects or clean curves. For example, Taggart’s system converts sketchy lines into “intended” straight lines and sketchy curves into pointed corners (Taggart, 1975). Sketch-Solid (Sivaloganathan, 1991) turns sketchy lines on an isometric grid into three dimensional objects. “Mark-based editing” (Baudel, 1994) uses overtracing sketches to modify clean spline curves.

The third group of “sketch” systems engage real freehand sketching abilities. They accept sketchy input, maintain the rough and sketchy presentation of the drawings and support editing or manipulation. For example, PerSketch (Saund & Moran, 1994) allows the user to select portions of a freehand drawing by overtracing. “Translucent patches” (Genau & Kramer, 1995; Kramer, 1994; Kramer, 1995) maintains handwriting and sketchy objects and supports moving irregular sketched shapes. Finally, SILK (Landay, 1996; Landay & Myers, 1995) will interpret freehand drawn objects as interface objects and allows the user to interact with the objects (e.g., move the slider, push a button, etc.).

To support early, conceptual design, this dissertation follows the strategy used by the “sketch” systems in the third group above. The Right-Tool-Right-Time system supports the input, display and manipulation of freehand sketches. It supports recognition of sketched objects. Unlike the second type of “sketch” systems that turn freehand drawings into straightened objects, it maintains the sketchy presentation and allows the designer to name drawing symbols according to her personal drawing style and preference. The details of sketch recognition are discussed in Chapters 3 and 6.

2.3.2. Why Computer Systems Need to Support Sketching

In architectural design today, though many CAD programs are available for drafting, most conceptual and creative work is still done using traditional media—paper and pencil. Pencil on paper is more flexible and easy to use compared with conventional CAD software. Using a pencil to draw allows the designer to explore more freely and quickly. With paper the designer can mark directly on the design drawing, indicating shape, line weight, color, and position without stopping to type commands or select menu items.

Designers draw what they want without the burden of learning obscure commands or being forced to specify a defined shape when they are only concerned about a rough object or space. Therefore, to support creative design work, computer aided design systems should emulate paper-based media and enable the designer to work in an unstructured way with a pen.

But what is the advantage to using computers to support sketching if the machine only mimics paper media? Some sophisticated painting programs (e.g., Painter, Photoshop, etc.[7]) allow freehand pen input and display a result that is similar to a scanned image. But to edit the painting, user must work at the pixel level. Such programs are not "intelligent" with respect to what is drawn. To support creative design, a computational sketching environment might offer additional capabilities. An obvious need is more powerful editing, allowing the user to reshape lines, to delete, group, and duplicate figures in sketch form as with conventional (structured, menu-based) CAD drawing tools.

Beyond tools for making and editing sketches, computing environments for creative and early conceptual design should recognize sketch and diagram elements and provide simulations, critiques, constraint maintenance, and knowledge based editing. More sophisticated enhancements could contribute even more useful functions to a computer-based sketching environment with advantages over "dumb" paper. They could provide access to relevant information during the earlier stages of design when changes in strategy are less costly. Designers might use the computer sooner in their design process and thus gain valuable feedback earlier in design. In short, a pen-based computational drawing environment that supports and preserves sketch would offer designers not only a way to

[7] These are commercial products. Painter is available from www.fractal.com and Photoshop is from www.adobe.com.

move smoothly and incrementally from conceptual sketches to more schematic design drawings, but also from conceptual drawing to hard line mechanical drawings, providing supportive design information (e.g., case library, simulation, estimation, etc.) along the way.

2.4. DISCUSSION – REPRESENTATION AND UNSTRUCTURED DRAWING

As described in previous sections, freehand drawing is an essential part of the design process. However, most systems that attempt to support drawing are actually “structured” drafting systems. Sutherland, although he built a drawing program that provides “clean drawing” admits that it is difficult to use Sketchpad to “sketch” because of its “structured nature” (Sutherland, 1975).

CAD supposedly focuses on systems that support design. However, current CAD system are generally limited to drafting, modeling and rendering. CAD is used by design professional to create presentation and construction drawings, or near-complete designs. Those who claim to use CAD for design such as Eisenman and Gehry (Gehry & Associates; IDOM, 1997; Giovannini, 1993) are actually using CAD to manipulate or represent the distortion and transformation of geometric shapes. Using CAD systems did not help these designers with their design reasoning. Most designers continue to develop solutions with paper and pen. They sketch to design.

What then, should a computer program recognize from a drawing? Schodek suggests that an object in a computational representation should not just carry one class inheritance but multiple classes for different purposes (Schodek, 1994). For example, a column in a facade and a column in structural analysis would have different meanings and should therefore have different representational foci. The interpretation of objects should change based on the different design purposes, yet most systems only offer one standardized description for an object. Schodek points out that when applying AI techniques to architectural problems we need to emphasize intent rather than merely problem solving. This suggests the need to detect design intention in a drawing.

If designers do share drawing conventions, then a computer program should be able to recognize these intentions. However, it is also possible that some conventions and design knowledge could be so obvious or well known that they would not be represented in a drawing. This is when a knowledgeable computer tool can help supply the missing knowledge. Tools that perform visual field analyses, evaluate energy consumption, present past failures or success stories, help with scheduling and budgeting, or estimate material costs are important because they serve different purposes in design. In a design environment these “intelligent” tools and techniques could help as reminders, consultants or could offer alternatives. If the designer’s intention are detected during drawing engagement based on the context, then the missing knowledge or the relevant tools could be supplied. Designers could make better use of these tools if they are made available at the right time with an appropriate interface.

This chapter has given an overview of related research. Section 2.1 described design studies of the importance of drawing in architectural design. This literature includes case studies based on interviews and observations of famous architects and on their self reports.

These researchers and designers argued that drawing is important for exploring and communicating ideas. Yet none of them reported what the drawing symbol universe of designers' might be. Section 2.2 examined the role of drawing from the view points of cognitive studies and protocol analysis. These studies identified drawing as representation of mental ideas and used "think aloud" protocols to highlight design thinking. They argued that drawing serves as external representation for design reasoning and design thinking. However, they did not identify what kinds of drawing symbols designers use to think about design concerns. Section 2.3 surveyed computational "sketch" systems that were designed to support drawing activities. These are pen input systems. Some of these systems supported real freehand drawing activities. Some supported conversion from sketchy lines to two-dimensional straight lines and geometric objects or to three-dimensional forms. The conclusion drawn from this chapter is that the best representation to link the design process with design knowledge should be a drawing interface.

This chapter provides the necessary background for the empirical studies and computational implementation in Chapters 3 through 6. The next chapter – Chapter 3 – reports on some pilot, or antecedent projects that led to this dissertation.

CHAPTER III

BACKGROUND – ANTECEDENT PROJECTS

This chapter reviews the systems that led up to the Right-Tool-Right-Time project. Two systems that aimed to support early conceptual design are of particular importance: 1) the Archie case library and 2) a freehand sketching program called the Electronic Cocktail Napkin. These two systems are brought forth for discussion because they introduce concepts fundamental to development of Right-Tool-Right-Time (RTRT). It should also be pointed out that the initial idea for RTRT came from an involvement in the development and usability testing of both systems.[1]

---

[1] A personal history note here. I joined the Archie team (lead by Janet Kolodner, Eric Domeshek and Craig Zimring) in 1992 when the system was on Symbolic Lisp machines with lots of text stories and little graphics. Together with the team, we moved and rebuilt Archie to Macintosh Common Lisp environment, classified post occupancy evaluation information data into categories of problems, responses and stories and supplied Archie with bountiful graphic material and indexing features. After performed two usability studies (total of ten designers) of Archie, I concluded that for Archie to be really useful for designers, it needs a graphic interface. Therefore, in 1993, I worked with Anna Zacherl (Do & Zacherl, 1993) on a diagram editor for Archie. In 1994, I started working with Mark D. Gross on building links to accessing Archie cases through diagrams made from his freehand sketching program called the Electronic Cocktail Napkin. After successfully linking Napkin diagrams with Archie stories, I implemented a shape similarity comparison scheme and added a sketchbook module to the Napkin environment to serve as a front end to different database and simulation programs. I also performed empirical studies to test the feasibility of using diagrams to access Archie case library, and identified designers' drawing conventions. Recognizing that drawing links to intentions, the idea of Right-Tool-Right-Time originated. Therefore, this thesis have the components of design studies, empirical studies and software implementation based on both Archie and Napkin programs.

The rest of this chapter is organized as follows. First, a brief review of phase models found in the design studies literature is given and a particular model is proposed for use in this study. The proposed phase model argues that different design phases involve different design activities and therefore require different kinds of support tools. The second part of the chapter describes a case-based design aid system called Archie. Involvement in the development of Archie led to the RTRT concept and thus Archie provides the first link to RTRT. The third part describes a freehand sketching environment, called the Electronic Cocktail Napkin, that supports conceptual design with trainable recognition of drawing symbols and configuration. The Napkin program eventually became the backbone for the development of RTRT. The fourth part describes a pilot implementation that supports finding case story in Archie through Napkin drawings. The fifth part describes a shape-based reminding scheme called Drawing Analogies. The Drawing Analogies prototype explored communication to different programs. Finally, the chapter concludes with a summary and discussion.

3.1. STUDIES ON DESIGN PROCESS

3.1.1 Phase Models in Design Studies

Many design studies researchers break design into different phases. For example, Asimow's "Introduction to Design" breaks primary design into three phases: 1) feasibility study, 2) preliminary design and 3) detailed design (Asimow, 1962). The purpose of a feasibility study is to identify design problems and to find possible, useful solutions. The preliminary design phase involves the analysis and synthesis of design solutions discovered in the feasibility study phase, leading to the identification of the best design alternatives for the given problem. Finally, the detailed design phase takes the concepts explored in the preliminary design phase and produces, through a master layout plan, a specification of components on paper. Asimow calls this chronological pattern the “morphology of design,” though each design project would have its own peculiar history. He argues that all projects follow the same sequence of development from initiation to final design. He proposes that a useful methodology can be derived by carefully examining the phases a design project goes through. Rigorous methodological patterns can then be used to transform engineering resources into useful, physical objects.

Sanoff, on the other hand, describes design as the transformation of information through the stages of 1) analysis, 2) synthesis and 3) evaluation (Sanoff, 1977). He views design as a kind of problem solving activity. In the process of design, each individual design idea is transformed through the same set of design strategies. A complex design problem will first be broke down into parts in the analysis phase so that the relationship between the component parts can be studied. The synthesis phase puts the parts together to form a solution and to integrate diverse concepts into a whole. In the evaluation phase, designers test and compare alternatives, leading to the selection of a good design solution. Evaluation might also generate new design problems for further analysis and synthesis in the design cycle. He argues that all designers go through these different stages to reduce design errors and the need for redesign, and to refine design solutions. Similarly, Markus describes design as decision making sequence of 1) identification, 2) finding relationships, 3) model building and 4) optimization (Markus, 1969). He calls these “appraisal activities” to represent the induction – deduction cycle of a design process.

Another phase model, based on observations of designers in action, describes the design process as composed of sketching concepts and images, exploring design ideas, investigating design themes, mapping site constraints, resolving conflicts, and developing concepts (Rowe, 1987).

These models all suggest that the design development process has different stages that involve different kinds of activities.

3.1.2. Activities, Media and Information in a Design Phase Model

Architectural design involves different activities. A particular phase model of design is proposed here based on interviews with designers about their design processes, the observation of a two architects' working on a design (Do & Zacherl, 1993), and a case study of Le Corbusier's design process of Carpenter Center (Do, 1993). The proposal is a three phase model of design and hence classifies designers' activities into three categories, 1) organization, 2) ideation, and 3) fabrication. Each activity is carried out using different drawing techniques, employs different kinds of external information, and therefore calls for different computational support (as discussed later in this section and also Chapter 6).

The first two phases involve the consideration of different issues, and usually appear as iterative interplay with no specific sequence. At certain times the designer's main concern is function, particularly the manipulation of objects within the context of the architectural program. This phase is called 'organization.' Drawings in this phase – usually crude sketches and diagrams – depict spatial layouts, circulation, cost and construction concerns, and zoning studies (Figure 3-1).[2]

---

[2] This suggests that sketching activities come after the initial definition of the problems by the designer.

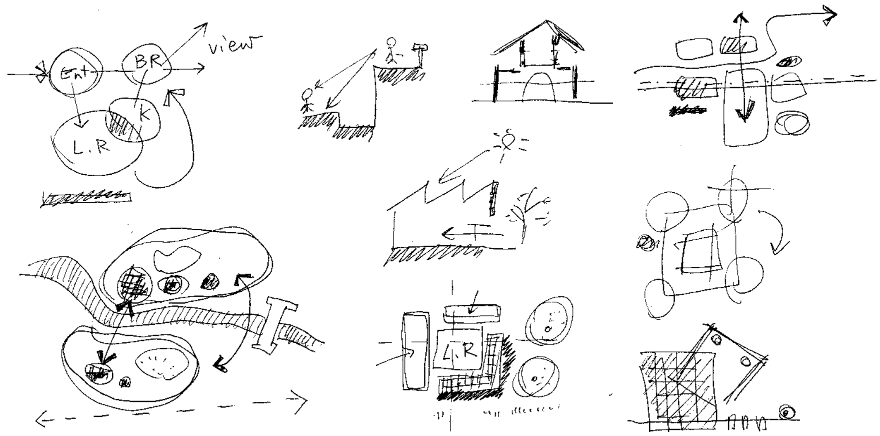


Figure 3-1. Organization drawings explore programmatic concerns.

More or less interposed with organization activities, which aim to resolve functional concerns, the designer also searches for creative images and solutions. In this second design phase, called 'ideation,' the designer sketches to explore ideas, employs metaphor and images, and often makes 'gestural' drawings of artifacts (Figure 3-2). Designers typically explore geometry and consider principles of form such as rhythm and harmony. They also look for ideas in design precedents (symmetry, axis, interplay of masses) or in other domains (such as natural objects or artifacts like plants and animals).

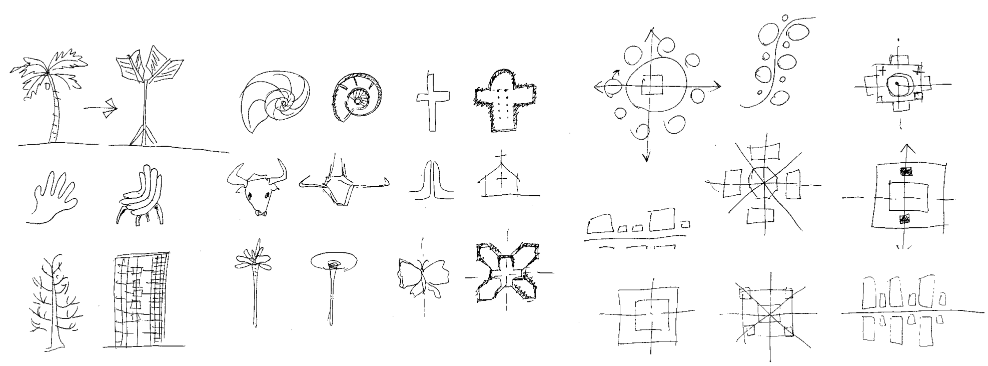


Figure 3-2. Ideation drawings explore shapes and forms.

Usually during a later phase the design is developed further and prepared for realization, at which point construction drawings are needed. These activities take place in the ‘fabrication’ phase of design. During fabrication, detailing and grids are usually more definite and precise than in the first two phases, and fabrication drawings (Figure 3-3) lead directly to the production of hard-line, mechanical, drawings.

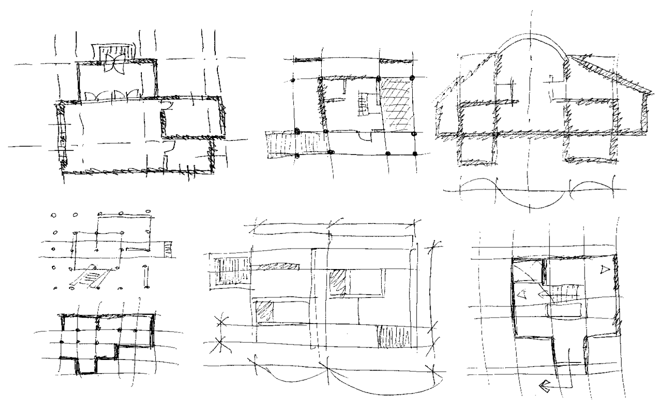


Figure 3-3. Fabrication drawings tend to be more definite and precise.

To sum up, it is proposed that early architectural design consists of three distinct phases: organization, ideation and fabrication. These phases iterate with no particular sequence and each incorporates many different design contexts and intentions. The three-phase model serves as a framework to identify what different types of design support might be needed for different activities. For example, in the Organization phase, designers usually manipulate entities that are abstracted with respect to function. They make bubble diagrams to explore adjacency concerns; they make calculations to check space requirements. In the Ideation phase, designers are often concerned with finding relevant visual references for proposed configurations and with testing their design proposals. They produce diagrams and sketches to analyze, for example, visual access and lighting conditions. In the Fabrication phase, designers trace over old sketchy lines and use tools for hard line drafting. These different design activities can be facilitated by different kinds of computational support.

In short, the proposed design process model argues that different activities and different kinds of drawings are employed in different design phases to deal with different concerns. The number of the phases is not critical for this thesis. The primary objectives is to identify different features of design drawings, so that future computational aids can be designed to have different modes serving different design concerns. The tool (the Right-Tool-Right-Time manager) proposed in this research deals with different modes of design activities that focus on different design intentions (Chapter 6).

Computational tools have been built to support these activities, especially organization and fabrication tasks. Functional analysis, spatial layout, and knowledge based evaluation programs support organizational decision making. Commercial CAD drafting programs provide libraries of geometric elements and building components, and they support fabrication. However, few systems have been built to support "ideation" activities.[3]

Databases of design information can aid designers in each of the design activities mentioned above, but the nature of the needed information, as well as an appropriate means of finding it, may vary widely. For example, for organizational decision making, catalogs of similar designs, precedents, and post-occupancy evaluation stories (as in the case-based design aid Archie (Zimring, Do, Domeshek & Kolodner, 1994)) may be most useful, whereas for fabrication decisions, a catalog of building components and details may be

---

[3] Therefore, I have proposed and build a "Drawing Analogy" prototype (Do & Gross, 1995a; Do & Gross, 1995b) to employ shape similarities as a way to access visual reference for creative idea exploration (Do & Gross, 1995a).

more appropriate. For ideation activities, the designer may employ more widely diverse visual references.

Embedding information tools into drawing environments is a key strategy for getting designers to use them. If information is not ready-to-hand, a designer will be reluctant to stop drawing to look it up. For example, a usability study (see 3.2.3) of Archie's key word lookup scheme (which is not embedded in a drawing or design generation environment) revealed that architects found the keyword (features) search mechanism cumbersome and that using it while designing interfered with their design "flow" (Do, Or, Carson, Chang & Hacker, 1994). McCall and Fischer (McCall, Bennett & Johnson, 1994; McCall, Fischer & Morch, 1990), in their Phidias and Janus systems, follow the strategy of embedding knowledge-based critics and access to design argumentation in structured CAD environments. However, structured CAD environments are problematic for creative designing. Especially for ideation activities, but also for organization, freehand sketching is the traditional, and arguably the most natural, medium; hence, it would be helpful if needed information were accessible through sketches and diagrams.

3.2. ARCHIE - A CASE-BASED DESIGN AID FOR ARCHITECTURE

3.2.1. Case-Based Reasoning

Case-based reasoning (Hammond, 1989; Hinrichs, 1992; Kolodner, 1993) means reasoning based on previous experiences. A case is an experienced situation. Case-based researchers believe that people solve new problems by adopting and adapting previous experiences. By referring to past experiences, one can understand new situations and evaluate proposed solutions. Those who have adopted the Case-Based Reasoning (CBR) concept have proposed CBR systems that would help designers to retrieve previous solution strategies and use them to frame their current problem situations.

To support design decision making, architectural knowledge bases have been built. These projects, such as Precedents (Oxman, 1993), CADSYN (Maher & Zhang, 1991), and Archie (Domeshek & Kolodner, 1992; Kolodner, 1991; Zimring, Do, Domeshek & Kolodner, 1995) concentrate on the applications of case-based reasoning to design and argue the use of case-based reasoning would help decision making, education, and creative problem solving.

3.2.2. The Content and Features of the Archie System

Archie is a case-based design assistant (CBDA) for architectural design. Like other CBDA programs, Archie contains a case base (case library) of designs including both good and bad exemplars annotated with stories that describe key design features and how they function in the building.

Archie consists of stories, problems and responses from post occupancy evaluation data collected in field studies. Currently Archie's case library contains post-occupancy evaluation data about courthouses, libraries, and tall buildings. This data is organized into categories (see Figure 3-4) of problems, responses and stories to cover a range of different design concerns.

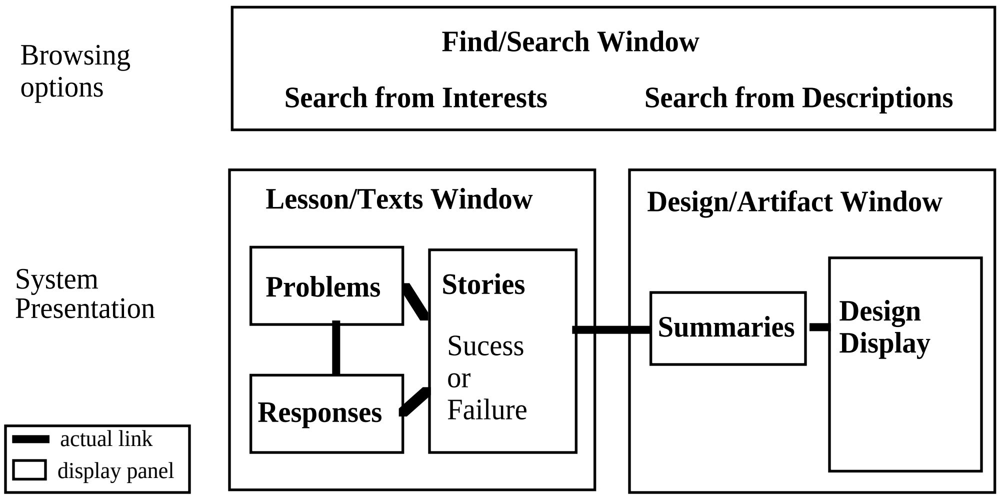


Figure 3-4. System diagram of Case-Based Design Aid Archie.

Each problem, response, and story is indexed using a set of key features organized along the dimensions of systems, components, design issues, stakeholders, and life cycle concerns. All related items are cross-linked. Each text item is clearly and concisely written, labeled with an explanatory title, and summarized in a one-paragraph synopsis along with scanned-in graphic material (plans, photographs, and drawings) obtained from the architect, from visits to the building, and most importantly, from post-occupancy evaluations (POE's).

A major part of the work in Archie has been to structure post-occupancy evaluation and other data in a format amenable for inclusion in Archie's case library. Both text and graphic data had to be entered into the library. Each problem, response, and story is indexed using a set of features developed specially for the Archie case library, and related items are cross-linked for browsing. Text items in each case are classified as "problems," "responses," or "stories." Each item had to be clearly and concisely written, labeled with an explanatory title, and summarized in a one-paragraph synopsis. Graphic material (plans, photographs, and drawings) associated with each text item is scanned and manipulated so that it appears clear on the screen (Figure 3-5).

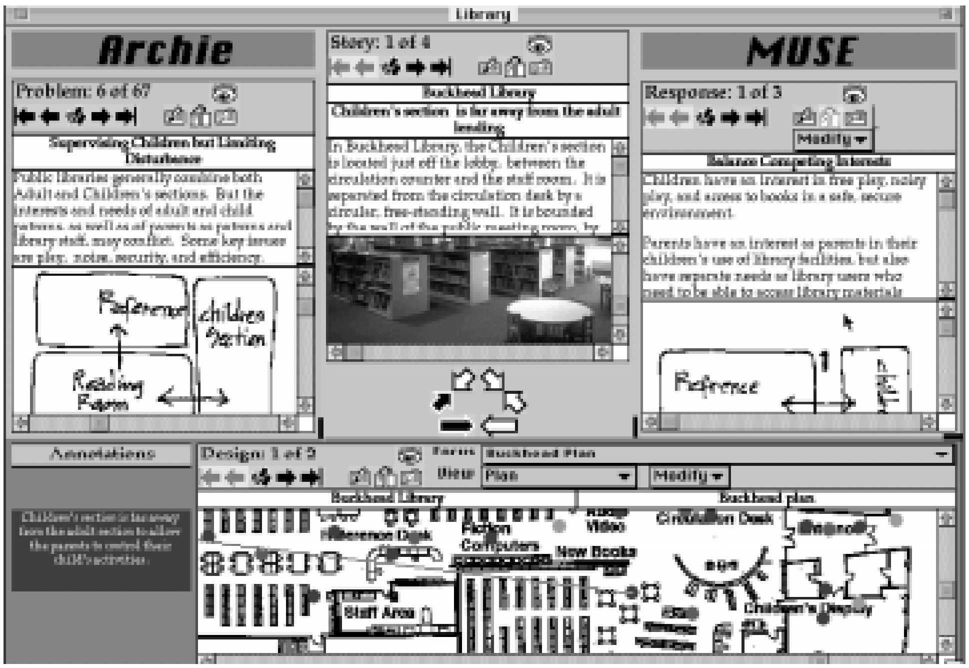


Figure 3-5. Archie screen shot. Top: windows for Problem, Story and Response. Bottom: Annotations (summaries) and Design (display of design artifact, e.g., floor plan).

Post-occupancy evaluation (POE) is a conventional method for evaluating the performance of a building design (Preiser, Rabinowitz & White, 1988; Zimring, Wineman & Carpman, 1988). POE proceeds by structured interviews with building users, and it frequently reveals ways a building is actually used that were not considered by the designer. Unfortunately, POE's are usually not carried out by the architect, but by a separate consulting firm. POE data is collected and presented to the client, and perhaps an immediate problem is solved; however the lessons learned from evaluation often do not reach the designer for consideration in future work. Therefore, Archie is aimed at making knowledge gained in post-occupancy evaluations more accessible to designers who are in the early stages of design and are working on similar design problems. A case-based design assistant like Archie can provide a mechanism for incorporating memory in a design firm; it can also help also coordinate various stakeholders interests in a complex building design.

The two main ways to find information in Archie's case library are (1) access through an index of the features of stories (Figure 3-4, top), and (2) browsing from story to story, or from a graphic image such as a floor plan (Figure 3-4, bottom, left and right). Initial access to items in Archie is through an index of key features to be found in the cases. The index is organized along five major dimensions: systems, components, design issues, stakeholders, and life cycle concerns (Figure 3-6). These dimensions, employed also in other case based design assistants, have specific interpretations in Archie. 'Systems' include the circulation, HVAC, and structural systems; 'components' identify specific elements such as rooms, doors, elevators; design issues include noise, privacy, access for physically challenged users; 'stakeholders' include the client, building management, and various classes of users; and 'life cycle concerns' include maintenance and repair.

A particular story is likely to be indexed by several categories of features. For example, in one case a story explains how the high internal pressure of the HVAC system results in the rear private staff doors hanging open (if staff occupy a separated, isolated zone), causing serious problems for the circulation system.

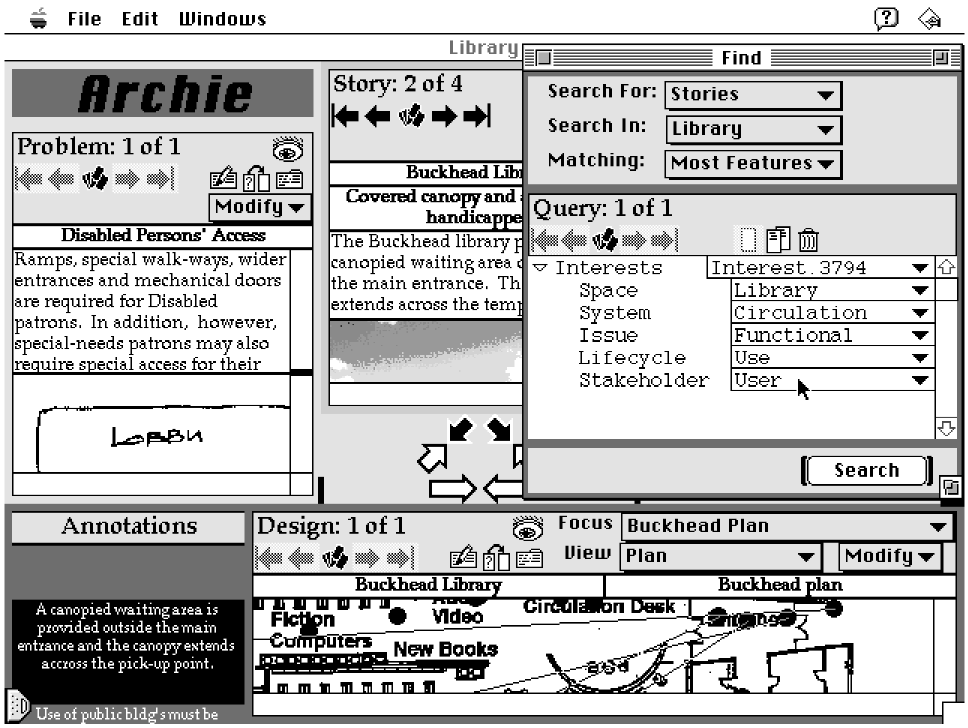


Figure 3-6. Accessing Archie case library through a search window by selecting predefined keyword. (Archie screen consists of panels for problem, story, response, annotation and design floor plan).

In the standard interface to Archie, the designer identifies retrieval keys by choosing them from a set of menus. For example, the designer can identify issues such as “access to courtrooms,” or “security & safety”; systems such as “circulation” or “building structure”; or provide descriptors to limit a search, such as “urban setting,” or “area greater than 200,00 square feet.” Archie’s retrieval machinery then provides items in the case library that have been tagged as potentially relevant to these concerns. In addition to the index that identifies major features of each story, links between items in the case library provide a way for a designer to browse from case to case without revisiting the index each time.

However, to access Archie's design information, one must use specific keywords assigned by the system developer. For visually-oriented, visual thinking designers, (Laseau, 1980; McKim, 1972) this text-based interface may be an obstacle to effective use.

3.2.3. Usability Study of Archie's Case Library

A way to access the usability of a system is to conduct usability studies. Together with a team of Computer Science students, a usability study of Archie was conducted using the paper mockup and lab approach. Paper prototype and monitored laboratory testing of computer systems are common used techniques in usability testing (Hix & Hartson, 1993; Laurel, 1990).

In the mockup version usability study (partly reported in (Do & Zacherl, 1993)), designers were asked to design a courthouse while a paper mock-up of the Archie interface was available on their desks. The whole design process was observed and video taped. The two designers were asked to speak aloud to explain what they were doing. Based on the experiment session and the debriefing after session, it appears that designers look for information and references from various sources in a design process. They would look for case stories from Archie if they knew what was in the database, or if the paper version happened to be handy on the desk.

Both designers expressed the importance of finding useful information to help make design decisions. However, they both enjoyed the act of drawing to explore ideas and design solutions instead of stopping to find useful information. One suggested it would be great if there was an assistant to help find all this useful information for him while he was designing. He then could remain in the design flow, not be interrupted by having to stop to look for material, and yet still have useful information available at the right time.

The other usability studies were conducted with five designers separately in a closely monitored lab (Graphics, Visualization and Usability Lab, College of Computing, Georgia Tech) (Do et al., 1994). The studies observed and video taped designer using the Archie program for design and information searching tasks. Designers were asked to compile a report from the case library for special topics, to use a case to help design a library or courthouse, or to simply find useful information from Archie.

The usability of Archie verified the obvious: designers need a drawing environment to access design information more easily and in a way that is naturally tied into design. This experiment found that designers tend to draw when designing and they dislike using keyboard commands to find information. They commented that even though the Archie system is useful, they would not use it in their design process. The reason is that in order to find useful information, they have to stop drawing and the design flow is interrupted. Therefore, it is concluded that a sketch-based interface for accessing knowledge based design tools like Archie would be useful for designers.

3.3. ELECTRONIC COCKTAIL NAPKIN, A SKETCHING ENVIRONMENT

The Electronic Cocktail Napkin (Gross, 1994; Gross, 1996) is an environment for making hand drawn sketches and diagrams (http://wallstreet.colorado.edu/Napkin). The

Cocktail Napkin program, as the name suggests, aims to support the kind of scribbling activity that happens on the back of an envelope or cocktail napkin. Designers use a digitizing tablet with a cordless pen or a mouse, or a PDA (personal digital assistant, e.g., an Apple Newton) for their design drawing. Designers can diagram and sketch freely on the drawing surface, and can customize the program to recognize personally defined symbols by combining drawing elements.

The program supports the recognition of drawing elements, records drawing pressure and pen paths, registers author information and drawing speed and creates time stamps. It retains inking information (raw x-y points and pressures) for display and for possible later re-interpretation. It recognizes and parses hand drawn diagrams into relational propositions. In addition to the trainable recognition of hand drawn glyphs and the parsing of more complex figures built from combinations of glyphs in certain spatial relations, the Cocktail Napkin supports storage and retrieval of sketches and diagrams (with sketchbook, Figure 3-7), tracing, multiple users, as well as standard CAD-like editing of raw stroke data. Later when the designer returns to a drawing, the system retrieves previously saved records.

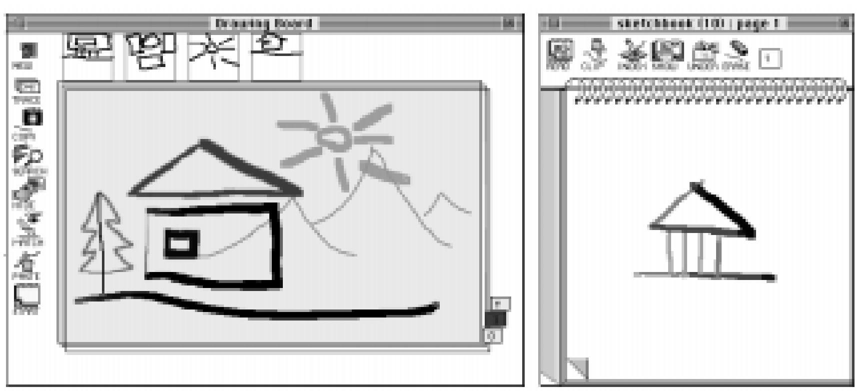


Figure 3-7. The Electronic Cocktail Napkin - [a] drawing area, [b] sketchbook for storing interesting sketches.

3.3.1. Making and Training the Computer to Recognize Drawing Marks

The Napkin program identifies the marks drawn by the designer as “glyphs. It attempts to recognize drawing marks as simple geometric shapes (such as Box, Circle “and Line, etc.) by comparing to the stored, customized and trained (by designers) data of previously defined training sets. When the designer is drawing, the name of the glyph (Box, Circle, Line) could be displayed or remain invisible (depending on switches) once they are recognized by the program. The Napkin maintains and displays the as-inked representation rather than automatically cleaning up the drawing (it does not, for example, turn sketchy lines into straight lines). Of course, the option to rectify a drawing can be turned on when in the ‘fabrication’ phase that is, when a construction drawing needs to be produced from concept sketches.

A glyph is recorded in a certain period of time (time-out function, that can be adjusted) between when the designer’s pen touches the drawing surface and when the pen is lifted up. Typically a ‘glyph’ is a single-stroke object such as a circle or a line. However, it can also include multiple strokes made in rapid succession. For example, a box glyph can be a one-stroke object or an object composed of four strokes.

The first process involved in the recognition of glyphs is identifying the bounding box – that is, the smallest rectangle that contains all the recorded coordinate points. Once the bounding box is identified, its aspect ratio and size are classified and a 3x3 grid (see Figure 3-8) is overlaid on top of the point coordinates. The pen path of the glyph then can be represented simply as a sequence of the grid square numbers (1-9).[4] Corners are identified when the direction (change in heading) of line segments between raw points exceeds 45 degrees. As shown in Figure 3-8, a clockwise circle and a box could have same pen path (e.g., [1 2 3 6 9 8 7 4]) but would have different numbers of corners.

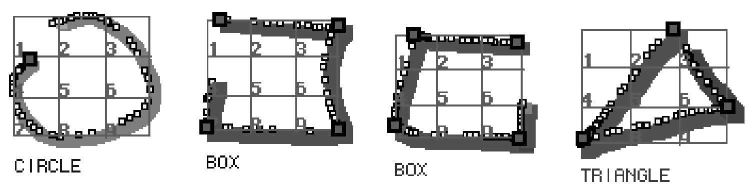


Figure 3-8. Features of simple glyphs: pen path in a 3x3 grid, raw x, y points, and corners.

A designer can train and introduce new ‘glyph’ templates to the Napkin system by drawing examples of the shape and entering its name. The Napkin program constructs a new template for the glyph, adds the features and the pen paths of the training samples to the new template. If such a template already exists, the training set of this glyph template will be expanded by adding in the features (such as pen path, corner, and aspect ratio) from the sample shape. For example, a designer may add a 4-stroke Box to a training set containing only 1-stroke Boxes. The “strokes” slot of the “Box” glyph template would then be expanded from the list [1] to the list [1 4]. Thus, constructing and refining training sets may occur at any time during use. These newly stored features from the new samples expanded the glyph training set and later would be used for recognition comparison of any new drawing marks designers make.

[4] The sequence list of a shape is stored for quick matching. Raw points are also stored for the processing of finding corners and measuring drawing speeds.

3.3.2. Defining Symbol Configurations by Example

Once simple glyphs are recognized, combinations of these glyphs can be define as a configuration of symbols. A drawing usually consists of not just the primitive marks of simple geometric shapes, but also symbols in higher level configurations. A configuration is a group of elements (glyphs or sub-configurations) that are arranged in a certain way to represent design concepts or objects. To identify configurations, the program runs recognizer functions over all pairs of elements in the diagram to look for these patterns.

The Napkin program provides support for recognizing configurations in drawings. Designers can define their shorthand symbols and patterns by giving example to the program. For example, a designer can define a configuration of a circle directly above of a triangle as an instance of a "person" symbol (Figure 3-9). A "computer screen" symbol could be defined as a box that is concentric to another box. A symbol configuration is recognized when the Napkin finds this spatial arrangement [concentric] combined with the element pair [box box]. It assembles the elements into configurations which are then named as symbols by the designers.

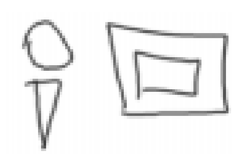


Figure 3-9. A person symbol can be defined as a circle placed directly above a triangle, while a computer monitor can be defined as two concentric boxes.

In the process of drawing, if the designers pauses more than five seconds, the Napkin's configuration recognizes looks over the drawing for patterns. This delay ensures that the recognizer won't interrupt the designer and prematurely parse a partially completed configuration.

By defining special configurations of symbols and glyphs, a designer can build up a whole drawing symbol universe. The Napkin program provides a special window (Figure 3-10) where it identifies the elements and spatial relations both graphically and as symbolic expressions. Napkin uses these patterns as matching criteria for the symbol recognizer. Designers can edit this pattern, deleting unwanted or incidental relationships and making the identity and spatial relation constraints more general or more specific. They can also name the configuration and provide an abstract view to replace the more detailed view of its parts or vice versa. For example, a table surrounded by four chairs can be replaced with a letter "D" inside a "Box" as a shorthand for "dining table configuration." On the other hand, the sketchy table layout can be replaced with a pattern that consists of "cleaned-up" and rectified rectangles (which could be useful in the fabrication phase).

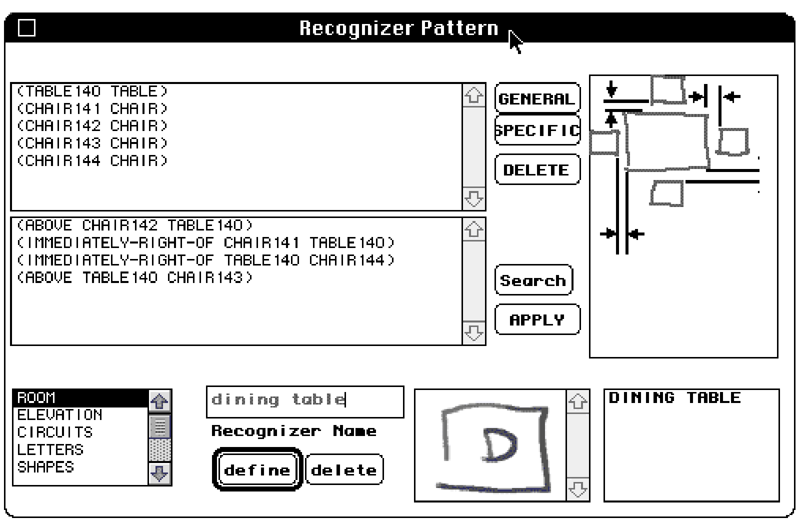


Figure 3-10. Defining a configuration by example: The Napkin shows relations, the user edits them.

The user can program the Cocktail Napkin's graphical search routines to recognize new, higher-level configurations by defining replacement rules that identify certain combinations of element types and spatial relations. For example, a user can define a collection of vertical lines as a "row of columns" which can in turn become part of a larger figure. A "temple" can be defined as a triangle "roof" on top of a "row of columns" with a rectangular "base" (Figure 3-11). The collection of recognition rules for configurations makes up a set of graphical rewrite rules for parsing a visual language of diagrams.

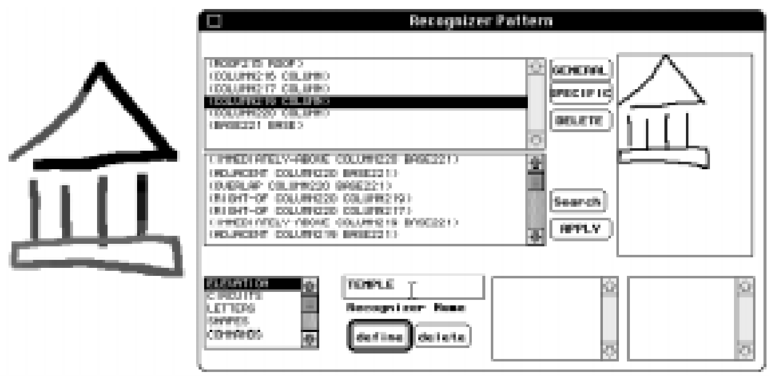


Figure 3-11. A configuration can be named and define in a search window that displays and analyzes element types and spatial relations between elements. For example, shown here a 'temple' is defined with a 'roof' immediately above row of 'columns' immediate above a 'base.'

3.3.3. THE RELATIONS BETWEEN SYMBOLS AND CONFIGURATIONS

Every symbol and configuration contains several constraints that describe the relationship between its own elements. For example, at the symbol's level, a designer may define a section of a table as a 'horizontal line' supported by two perpendicular 'vertical lines.' The width of the table could be defined in the range of 3 to 5 ft, and stored as a constraint on the table's width value. The Napkin maintains this constraint and permits the designer to size the box only within this range.

Spatial relations among drawing elements can be represented as constraints as well. These constraints include spatial relations such as above, below, left-of and right-of, contains, overlaps, connects, and intersects. Spatial relations belong to classes; for example ‘topology’ constraints include line connections and containment, and ‘spatial layout’ constraints include the above, below, right-of and left-of relations. These relations are ordered by their degree of specificity. For example, “concentric” is a more specific version of “contains” which in turn is more specific than “overlaps.” When identifying spatial relations among drawing elements, the Napkin chooses the most specific version that applies.

The Napkin provides an interface to adjust the constraints in the configuration recognizer. In the configuration window (see Figure 3-11 above) the designer can view and explicitly adjust the constraints of the element types and the spatial relations among them. The interface allows the designer to select a specific symbol element or relation and to adjust it to be more specific or more general, or even to delete it. For example, performing the General operation on the type constraint [Circle1 Circle] converts the description to [Glyph1 Shape]. On the other hand, performing the Specific operation on [Glyph1 Shape] the designer can choose a specific type from the various Shape glyphs.

To enable flexible matching, spatial relations and element types may be identified more or less specifically in the search pattern. For example, a search for the pattern [A concentric B] finds only pairs of elements A and B that have roughly the same center point (actually, whose bounding boxes are centered); [A contains B] is a more general version of the relation that simply requires B to be inside the bounds of A; and [A overlaps B] requires only that the two figures share some spatial extent (Figure 3-12). Similarly, elements A and

B can be identified in the search pattern more or less specifically as in, for example, ‘a letter M,’ ‘any letter,’ or ‘any object.’

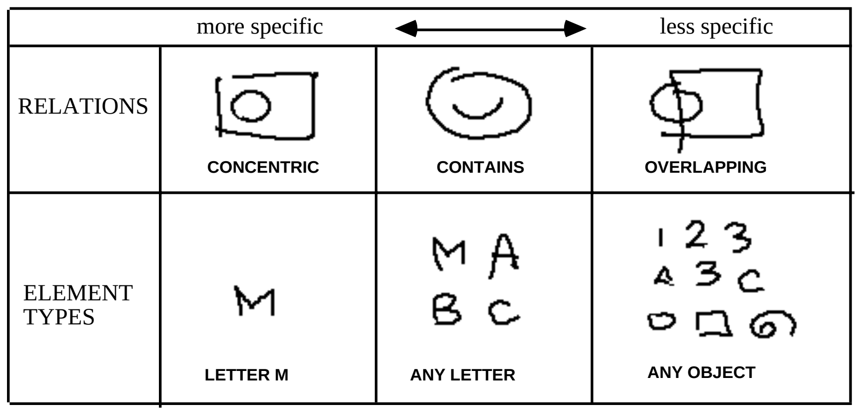


Figure 3-12. A hierarchy of spatial relations and element types enables flexible matching.

Designers can also adjust relation descriptors. For example, a spatial relation of a space bubble that represents an ‘adult’s room’ in a public library could be drawn with a circle with a letter A as a label. If the circle and the letter A happen to be draw as ‘concentric,’ then the spatial relation constraint will read [Concentric Circle1 LetterA]. Designers can perform the General operation to change this to [Contains Circle1 LetterA] so that future space labeling would get recognized even when the letter and bubble were not concentric. A designer can make a set of diagrams and assemble them into a collection called a ‘sketchbook,’ browse previous sketchbooks, or use the Cocktail Napkin to search for combinations of elements in certain spatial relations.

3.4. ARCHIE'S NAPKIN - INDEXING ARCHIE'S CASE LIBRARY WITH DIAGRAMS

The sections above described the case-based design aid Archie and reviewed the usability studies that conclude that a drawing and diagramming interface is necessary to make Archie more accessible to designers. Also discussed were the tools that were built to support freehand drawing and the advantages that computer recognizable drawing has over paper sketches. Furthermore, a freehand sketching program called Electronic Cocktail Napkin, its features and functions for recognizing simple drawings and configurations was described. The obvious next step is to link the Archie case stories with Cocktail Napkin’s diagrams. The section below briefly describe efforts in accessing Archie case library through hand drawn diagrams (Gross, Zimring & Do, 1994).

Most of the stories in Archie’s case library include some graphic information – a plan fragment or a small diagram, for example, that illustrates or explains the principle described in the story. These illustrations are made using a drawing program or are scanned in from printed building documentation and sketches obtained from the field studies. To facilitate access of stories from hand drawn sketches, each story must be augmented with a simplified diagram of the illustration[5] composed of a small number of basic glyphs. These diagrams bear the same relation to the illustration that the title and keywords bear to the story text – they summarize and highlight an important point or relationship, but they do not tell the whole story. Many of these diagrams are simple ‘bubble diagrams’ made of crude ovals, linked by short line segments or arrows, and often labeled. In bubble

[5] ‘Illustration’ here refer to the more detailed picture and ‘diagram’ to refer to the crude, simplified representation.

diagrams, certain relationships are particularly important like adjacency, containment, overlap, and connections with lines, whereas size and shape are often incidental.

As Figure 3-13 shows, the diagram keys are simpler than the figures used to illustrate the story. The illustration (at left) aims to provide both a visual summary of the problem described in the story and to portray a specific layout from the case. The illustration often adds information not found in the text. On the other hand, the diagram used as a key or sketch-based index shows only one aspect of the story – in this instance, the concept of two connected zones serving different functions. The diagrammatic elements are simple – a line and two bubbles, with the label "lobby."

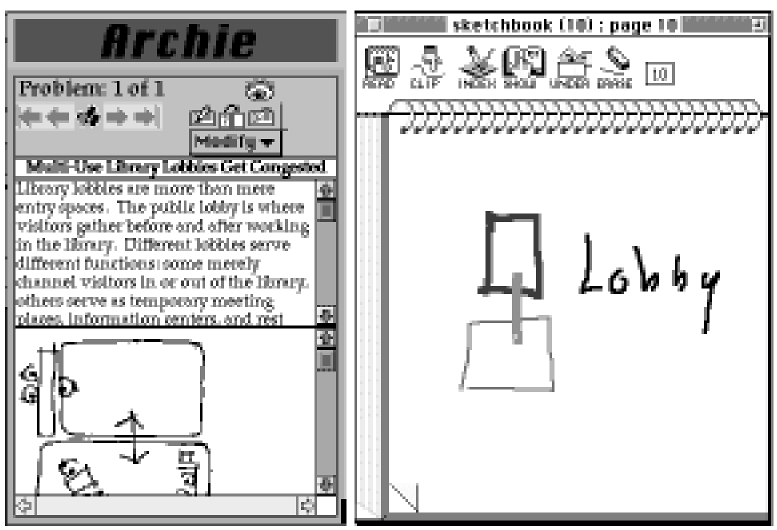


Figure 3-13. An illustration from a library lobby story with corresponding diagram.

Another example, shown in Figure 3-14, illustrates a concern about "spatial relations between Adult and Children's section." In this story, the locations of the adult and children's sections in a library are discussed. Adult patrons and library staff of a public library are mostly concerned about security of the children as well as noise disturbance.

They need easy supervision of the children while maintaining easy access to their own reading and reference area. The goal of the children’s section is to make a separate area that has a play atmosphere with open bookshelves and safe furniture. Designers here illustrate the concerns about conflicting interest in a spatial layout diagram. The illustration portrayed the location of an adult's section with respect to a reference area and a children’s section. Figure 3-14 shows both the formal illustration (at left) for the story as well as the simple diagram used for indexing it with respect to sketch input. This diagram includes ‘space bubbles’ that have letters to indicate functional space name (e.g., A, R, C for Adult, Reference and Children).

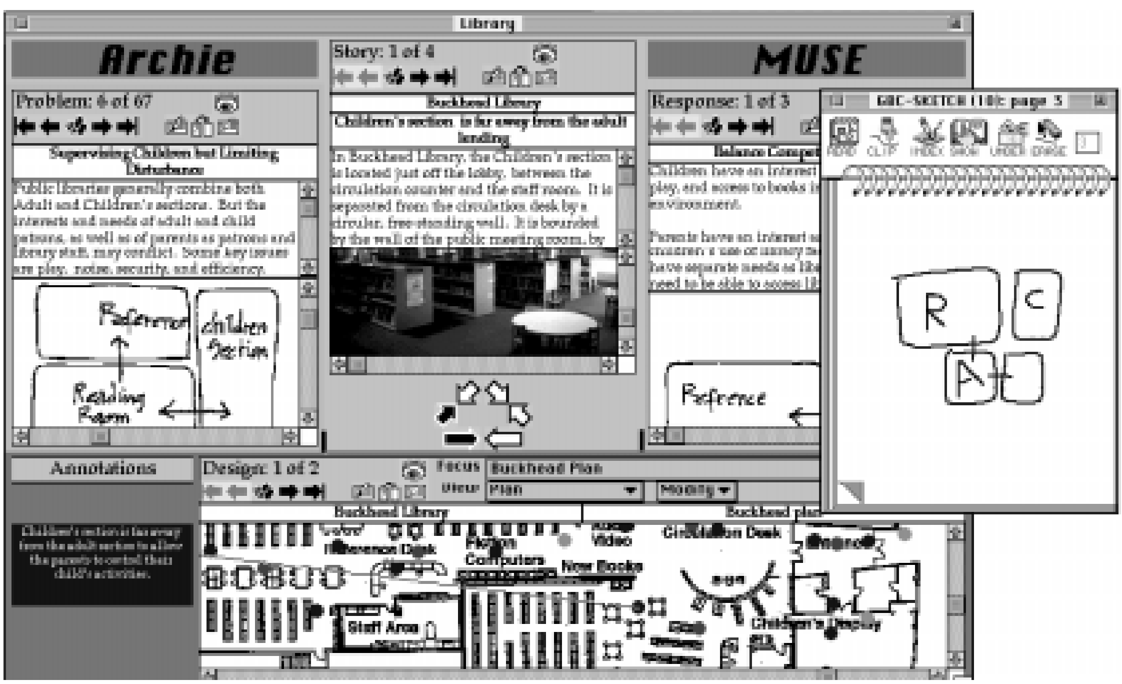


Figure 3-14. Indexing Archie stories with 'bubble diagram' drawn on the Cocktail Napkin program

3.5. DRAWING ANALOGIES – A SHAPE-BASED REMINDING SCHEME TO SUPPORT CREATIVE DESIGN

When designers work with ‘conceptual sketches’ and are thinking about metaphors and form making, a scheme for retrieving drawings and information based on drawing similarity might be of interest. A pilot implementation of the Right-Tool-Right-Time called “Drawing Analogies” explores ways to compare diagrams and sketches, using element type and spatial relation information for scoring similarity (Do & Gross, 1995a; Do & Gross, 1995b; Do & Gross, 1995c). The scheme links the Napkin to visual databases to serve as a shape-based reminding device for designers. A use scenario of Drawing Analogies is described in Appendix A.

The Drawing Analogies module uses the features of a query drawing to find similar drawings in the sketchbook, which are linked with images in visual databases. The problem of searching for similar shapes is not limited to the domain of design. The computer vision and image processing communities have built image retrieval systems. They typically use extracted features from pre-processed images as retrieval keys (Grosky, Neo & Mehrotra, 1992; Nakakoji, Reeves, Aoki, Suzuki & Mizushima, 1995; Stein & Medioni, 1992). Furthermore, many image processing projects focus on finding boundary features and solving occlusion problems. The reason for the focus on occlusion is that in real life photographs, foreground objects often occlude other objects behind them. Therefore, these retrieval mechanisms only examine specific line features of a shape, for example, concavity and convexity, and do not treat the images as compositions of objects in which spatial relationships are important.

The scheme in Drawing Analogies for assessing drawing similarity depends not only on simple shape profile features, but also on spatial relationships among drawing elements. Drawing elements may include simple shapes such as lines, circles, and triangles, as well as symbols and configurations such as columns, windows, and trees that are composed from simpler shapes. Relations are identified from a predetermined list, including adjacent, above, contains, concentric, and overlapping. The following list discusses how one can assess the similarity of drawings.

(1). The element count similarity measure compares the number of elements. Two drawings are identical with respect to this measure if they have exactly the same number of elements.

(2). The element type similarity measure compares the element types in the two drawings. Two drawings are identical with respect to this measure if they have exactly the same element types (irrespective of the number of elements). For example, two drawings composed only of boxes, circles, and lines (Figure 3-15) are identical with respect to element type matching, even though they are arranged in entirely different configurations and have different numbers of each element type.

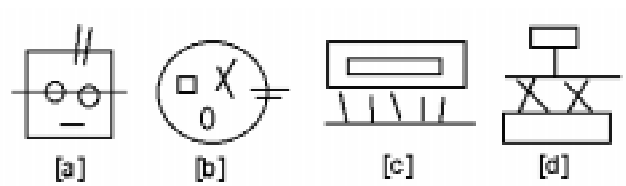


Figure 3-15. Element Type Match: pairs [a, b] and [c, d] have exactly the same element types.

(3) The relations similarity measure compares spatial relations only. Two drawings are identical with respect to this measure if they contain exactly the same spatial relations, regardless of the elements involved. For example, a ‘box containing a circle’ and a ‘triangle containing a line’ are identical (Figure 3-16, b), because in both drawings one element contains another. The same rules apply to configurations as well, so all the drawings in Figure 3-16 have the same 'containment' relation.

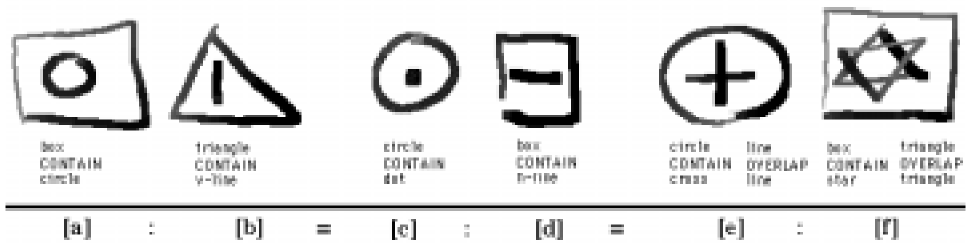


Figure 3-16. Relations match -- all figures exhibit 'containment'.

(4). The element relations&type similarity measure combines the previous matching primitives to provide a finer comparison of drawings that considers elements and their spatial relations together. Two drawings are identical by this measure if they comprise exactly the same elements arranged in exactly the same spatial relations. For example, the pair of drawings in Figure 3-17 a and b are identical; 3-17 c has a different relationship and 3-17 d has different element types.

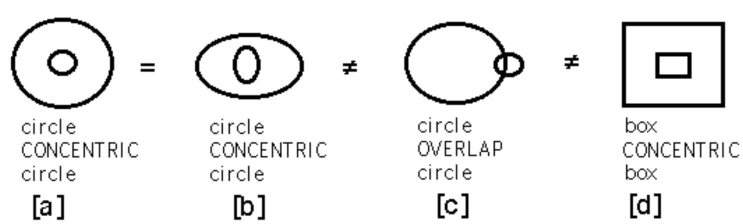


Figure 3-17. Element type&relations match: [a] is similar to [b], but not to [c] or [d].

Figure 3-18 shows the similarity scores of one pattern (Figure 3-18a) compared with several others (Figures 3-18b-e) according to the element type, relations, and relations&type measures. For example, when comparing Figure 3-18b with 3-18a, 2/3 of the element types match (both have a triangle and a horizontal line), the relations match entirely (both have two ‘directly-above’ relations) but the combined relations&type measures scores 0, since none of the matching relations have the same element types. On the other hand, comparing Figure 3-18c with Figure 3-18a, the combined relations&type measure scores 1/2, because one of the two relations in the former matches exactly with that of the latter (box directly-above line).

FIGURE

	[a]	[b]	[c]	[d]	[e]
ELEMENTS	triangle box horizontal line	triangle row of columns horizontal line	half circle box horizontal line	box box box	line triangle box
RELATIONS	tri d-above box box d-above line	tri d-above row-col row-col d-above line	half above box box d-above line	box above box box d-above box	line d-above tri tri d-above box
SIMILARITY	**element type** **relations** **relations&type**	**2/3** **1** **0**	**2/3** **1/2** **1/2**	**1/3** **1/2** **0**	**1/3** **1** **0**

Figure 3-18. Some dimensions of diagram similarity: [a] compared with [b, c, d, e].

The Drawing Analogies program added a ‘sketchbook’ to the Cocktail Napkin program. The sketchbook is where the designer collect their diagrams and associates them with items in different reference databases. Sketchbook pages serve as visual bookmarks. The designer can copy pages between drawings and the sketchbook. Figure 3-19 below shows the relationships between the design drawing, the scratch pad, the sketchbook, and the visual reference libraries.

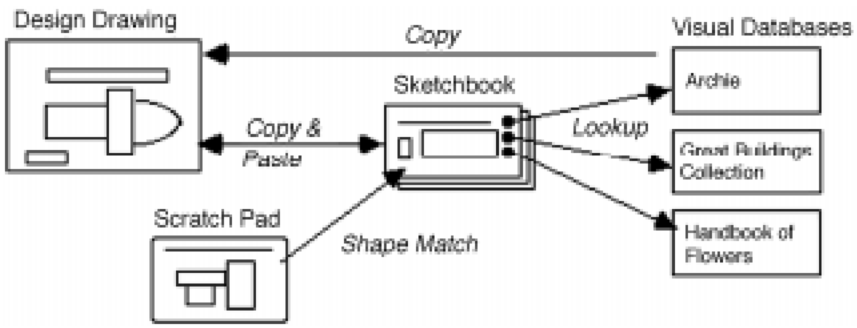


Figure 3-19. Parts of the Drawing Analogies system.

The Drawing Analogies program provides two basic functions. First, the designer can link a sketchbook page with an item in a visual database. Second, the designer can query the system for visual references by drawing a diagram sketch. To store a link, the designer locates an item in one of the visual databases that she wishes to index with a sketch. She draws a sketch on the scratch pad, and copies it to her sketchbook. By performing a ‘link’ command, the Drawing Analogies program finds an item in the most recently referenced database with the current sketchbook page. What it retrieved depends on the database. For example, if the link is made to the Great Buildings Collection (Matthews, 1994) it may entail turning to a particular card in the HyperCard stack, opening image files, and displaying a QuickTime animation or a three-dimensional model. If the link is made to an Archie case library, the sketchbook would find a design story whose illustration resembles the drawing on the current page of the sketchbook.

The Drawing Analogies module is useful for the Ideation phase (as described in Section 3.1.2 earlier in this chapter) – that is, when designers are in search of ideas and concepts. This module provides a shape-based reminding scheme for finding visual references based on shape similarities. This scheme supports explicitly specified intentions such as 'show me a building facade looks like my drawing.' Besides these explicit intentions, there are implicit ones embedded in other kinds of design phases (e.g., Organization and Fabrication) that can be derived from the empirical studies reported in the next Chapters (Chapters 4 and 5). Based on the success of this scheme, a 'semi-automation' and 'automation' scheme is developed called Right-Tool-Right-Time, as investigated and reported in Chapter 6.

3.6. SUMMARY AND DISCUSSION

There are two premises for the prototype described above in section 3.4.5. The first is that sketching is a natural medium for exploration during early design. The second is that case-based reminding of important issues and relationships at the early stage can save time and trouble later on in the design process. The prototype system combining Archie and the Napkin shows how stories from Archie's case library can be retrieved in response to simple hand drawn diagrams. As the designer sketches, the Napkin program identifies key elements and relationships, matches them with diagrams in the case library, and brings relevant stories to the designer's attention. Of course, one could argue that the sketches are not needed; merely a specification of the elements and relations they embody. However, as stated in the beginning of the previous chapter (section 2.1-2.3) drawing is important in design development, supports multiple information and activities, helps designers see and refine. Furthermore, the usability studies of Archie systems suggests that designers like to draw in the process of designing. To make use of the information in a system like Archie, they prefer a natural, non-intrusive drawing interface connected to the drawing environment. Therefore it is believed that allowing the designer to express information requests through the natural medium of sketching will add more value to the system.

When “reminding” is turned on, the sketch program searches the case library for diagrams that match parts of the designer’s evolving sketch. The match can be made more or less exact. For example, suppose a designer is thinking about spatial arrangements in a public library and draws a diagram for “adult’s section” and a “children’s section” (see Figure 3-14). If inexact matches are permitted, a space bubble with the label ‘A(dult)’ in the designer’s sketch will recall all diagrams containing the “adult section” and retrieve their associated stories. On the other hand, in a more exact match, only diagrams that contain both an adult section and a children’s section in a library will be retrieved. Likewise, a bubble with an arrow pointing into it is often used to indicate “entrance.” If inexact matching is permitted, then all diagrams containing this configuration will be retrieved along with their stories; on the other hand, if the label “lobby” is included and an exact match is sought, then the sketch will only retrieve stories about lobby entrances.

Archie and Napkin are two already-built, stand-alone programs. The integration of diagram making and the Archie program is a simple and crude demonstration of how a sketching interface might work with an intelligent knowledge based design tool. In this linking scheme, each story that is diagrammatically indexed points to one or more diagrams in a separate catalog. The sketch program’s graphical search routine is set up to search the catalog of diagrams. When it finds a matching diagram, the sketch program signals Archie to retrieve the associated story. Thus the sketch program and Archie remain distinct modules sharing a data structure and they communicate only in the simplest possible way.

Linking Archie with a sketch program has clarified the importance of integrating AI tools and techniques with environments that support and enhance regular design activities such as making drawings, diagrams, and sketches. If AI tools are to be effective design assistants, they must be able to communicate with designers with as little interference as possible. It is believed embedding tools in a sketch-based environment offers the best hope that they will actually get used.

In summary, this chapter form a basis for the empirical studies and computational implementation that is described in the following chapters (Chapters 3, 4, and 5). This chapter has: 1) reviewed existing phase models of the design process and proposed a new one; 2) described a case-based design aid called Archie; 3) described a freehand sketching program called the Electronic Cocktail Napkin; 4) described Archie’s Napkin, a prototype of accessing case library from sketching diagrams; and 5) described Drawing Analogies, a shape-based reminding scheme to support creative design.

The obvious next question is whether it is feasible to use diagrams to access design stories in a case library like Archie. To address this question, the following chapters (Chapters 4 and 5) report on empirical studies to find out the relations between sketching diagrams and design intentions.

CHAPTER IV

WHAT'S IN A DIAGRAM THAT A COMPUTER SHOULD UNDERSTAND

To better understand how drawings are used in design, this chapter and the next one describe controlled protocol studies of designers conducted by the author. Specifically, these chapters attempt to answer two questions: 1) What different activities and drawings consistently appear in the conceptual stages of design? and 2) How are symbols and drawing conventions associated with different design activities. The ultimate goal is to determine how one might identify design contexts and intentions simply by looking at a designer's drawing – the benefit being that one could then use diagrams to access appropriate design information automatically.

The hypothesis of this chapter is that designers represent architectural concepts in a consistent and conventional way, using a limited set of diagrammatic elements. In order to generalize how different designers actually use design diagrams in different contexts and to check for consistency, an empirical study was designed to see how diagrams are related to certain design concerns (represented as problems, responses and stories from Archie's case library). The experiment asked design students to both draw and interpret a variety of diagrams representing architectural issues derived from post-occupancy data. The purpose of the experiment is to test the feasibility of using diagrams as a way to access design information. Chapter 5 follows with a second design drawing experiment set up to more generally investigate the relations between drawings, their configurations and design intentions in the process of design.

4.1. ARCHIE DIAGRAM STORY EXPERIMENT

The experiment reported here focused on using diagrams to access cases from Archie, an existing case-based design aid (Domeshek & Kolodner, 1992; Kolodner, 1991; Zimring, Do, Domeshek & Kolodner, 1995). Archie has a database library filled with post-occupancy evaluation cases (see Section 3.2.2). Archie's case library contains stories, problems and responses from post-occupancy evaluation data collected in field studies of about ten courthouses and libraries. All related items are cross-linked. However, to access Archie's design information, one must use specific keywords assigned by the system developer. For visually oriented, graphic thinking designers (Laseau, 1980; McKim, 1972), a text-based interface may be an obstacle to effective use. Therefore, a prototype of a diagramming tool was developed. It helps users to find cases in the Archie database using only hand drawn sketch diagrams (see Sections 3.2.3, 3.4). Figure 4-1 shows a case story about the arrangement of adult and children's sections in a library that was retrieved by a diagram in which annotated shapes indicate spaces. To further develop this prototype, it is important to explore how diagrams and elements that make up diagrams are used in general.

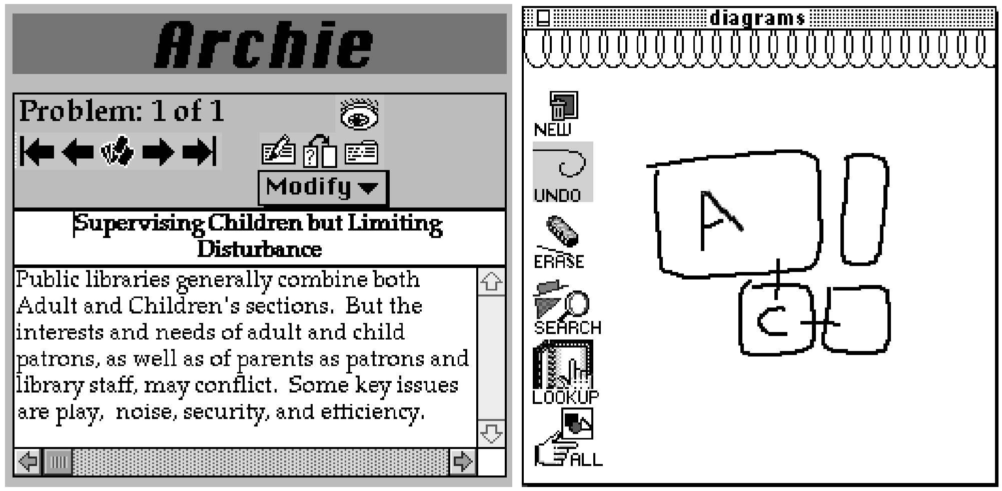


Figure 4-1. A case of adult and children's sections arrangement in library indexed and retrieved by a diagram. (A for adult and C for children).

If different designers use diagrams in a consistent way, then different designers should be able to access Archie cases through a common sketch index. Specifically, if designers are found to use a limited set of symbols in diagrams, then these symbols can be used to index all the cases with diagrams, and the computer can be trained to recognize diagrams. Therefore, these experiments were designed to test whether designers draw and interpret diagrams in a consistent way.[1]

4.2. DEFINITION OF A DIAGRAM

Before describing the experimental setup and task sequence, some basic terms need to be clarified. In particular, the words "diagram," and "sketch" may mean different things to

---

[1] I reported this experiment in detail in my qualifying paper (Do, 1995), titled The Feasibility Study of Using a Diagram Interface to Access an Architectural Knowledge Base. It is also partially published in CAAD Futures 95, titled "What's in a Diagram that a Computer should Understand".

different people. The distinction between different kinds of architectural drawings and their uses in design is the subject of several recent books (Fraser & Henmi, 1994; Herbert, 1993; Robbins, 1994). This study focuses on diagrams that explain design issues of environmental forces, building components, and human responses. The term “diagram” is used here to mean a drawing that uses geometric elements to abstractly represent natural and artificial phenomena such as sound, light, wind, and rain; building components such as walls, windows, doors and furniture; and human behavior such as sight, perception of privacy, and circulation, as well as territorial boundaries of spaces or rooms.

A diagram differs from a sketch in that it contains symbols. For example, a diagram might have an arrow indicating directional “force,” which is unlikely to appear in a sketch. A diagram also represents information without giving detailed descriptions of scale or realistic pictorial representations. It can be spatial, for example, showing relative positions and approximate sizes of rooms, or it can be non-spatial, showing the sequence of building construction.

A diagram indicates spatial relationships only in approximate form, using indefinite shapes. For example, a bubble diagram represents functional spaces in a floor plan with rough sizes, adjacencies, containment, and connections. In contrast, a sketch does not use symbols or abstractions and mainly about spatial form. For example, a perspective sketch provides three dimensional information about a scene, indicating physical elements and their spatial relationships. A plan or elevation sketch may be concerned with the proportions of a building or its components. A schematic drawing has features of both a diagram and a sketch. It uses conventional symbols to represent building components but has the spatial feel of a sketch. It is more complex and more precise than a diagram, and is drawn to scale. Despite these general differences, there are no clear cut distinctions between diagrams, sketches and schematic drawings.

4.3. EXPERIMENT SETUPS

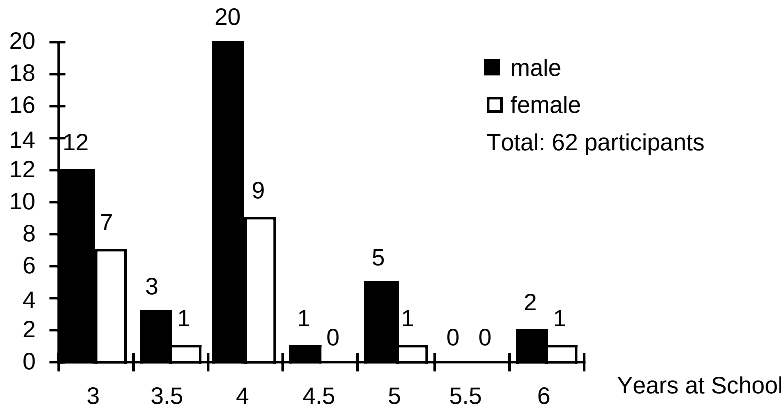
4.3.1. Goal and Reasons for Conducting The Experiment

The purpose of the experiment is to find out what can be read in diagrams, what symbols are used the most, and how they are used. It was performed to explore whether there are common features among diagrams drawn by different designers. To determine what a drawing might implicitly or explicitly mean, the following questions were asked: 1) Will different people construct similar images when attempting to illustrate the same text? and 2) Will they describe in similar words the same drawing or abstract diagram, with and without textual hints? The experiments drew material from Archie's case library and asked participants to first illustrate written concepts with diagrams and second, develop stories from given diagrams.

4.3.2. Experiment Sequences and Test Materials

Sixty-two undergraduate design students participated in the experiment. There were nineteen (31%) female students and forty-three (69%) male students ranging from twenty to thirty-three years old. All participants for the experiments were students enrolled in the Design Theory and Methods core course offered by Environmental Design Department of the University of Colorado at Boulder.

Table 4-1. Years of education distribution of participants.



The experiment took place in the classroom where the class was taught, during regular class hours. There were approximately thirty-five students in each class, all architecture majors. Most were seniors with between three and five semesters of design studio experience (see Table 4-1). They were thus assumed to have basic understanding of the design process, to know how to illustrate design concepts and to be able to interpret diagrams drawn by other designers.

Table 4-2. Groups and their respective test variations.

Group	Task Sequence	Titles	Participants
A	1 - 2 - 3 - 4	yes	17
B	1 - 2 - 3 - 4	no	17
C	2 - 1 - 3 - 4	yes	11
D	2 - 1 - 3 - 4 *	no	17

* Task 1 – illustrating texts, Task 2 – interpreting diagrams, Task 3 – diagrams and texts pairing, Task 4 – commenting Archie diagram-text pairs.

The participants were divided into four groups, each of which had a different task sequence (Table 4-2). The tasks included making diagrams from stories, writing stories from given diagrams, pairing diagrams and stories and commenting on existing Archie diagram-story pairs regarding library and courthouse design (Table 4-3).

Table 4-3. Standard Task sequence in a test (for Group A and B).

Task Sequence	Tasks
pre-evaluation questionnaire	background information
Task 1	illustrating texts
Task 2	interpreting diagrams
Task 3	pairing diagrams and texts
Task 4	commenting Archie diagram-text pairs
post-evaluation questionnaire	comments and suggestions about the experiment

In each task, the first three sub-tasks were concerned with architecturally problematic situations (Problems), and the second three sub-tasks with possible design responses to an architectural problem (Responses). Participants were asked to spend no more than ten minutes on each task, although the timing restriction was not enforced. The time spent by participants on all tasks ranged from thirty minutes to one hour.

The first Task (1) asked participants to illustrate given text describing either an architectural problem or a design response. The actual sub-tasks (Tasks 1a through 1f) used in Task 1 are shown below, where the first three describe a problem and the second three describe an architectural response:

Task 1a (Problem 1) Maps in Possible Special Collection

> *In a community library, a separated special map room allows control over access to valuable material but if out of sight of normal staff areas may require an additional staff member to supervise.*

Task 1b (Problem 2) Natural Lighting May Create Glare on Video Screens

Natural daylight is pleasing in offices, but may create glare on video screens.

Task 1c (Problem 3) Fun Spaces for Children in Library

The Children's section in a library must be fun for child patrons too young to sit and read for long stretches, but the noise and activity they create can disturb children old enough to read to themselves.

Task 1d (Response 1) Walls as Acoustic Insulation

Use thick walls between courtrooms and public areas to reduce sound transmission.

Task 1e (Response 2) Visual Contact to Children's Area

Buffer noise from Children's areas with special materials, boundaries, and physical separation from noise sensitive Adult areas, yet maintain visibility between Staff and children, or parents and children, through use of cutouts, glass walls, or overlooks.

Task 1f (Response 3) Use Frosted or Block Glass along Corridors

Use frosted glass on corridor windows to allow lighting from the atrium to come into the office yet so that passers by in the corridor can not see inside, allowing the office to remain private.

The second Task (2) asked participants to write a brief text description for a given architectural diagram (3 Problems and 3 Responses). They were given brief guidelines for writing text descriptions with examples. Figure 4-2 below shows the diagram from the first sub-task in this task with its title.

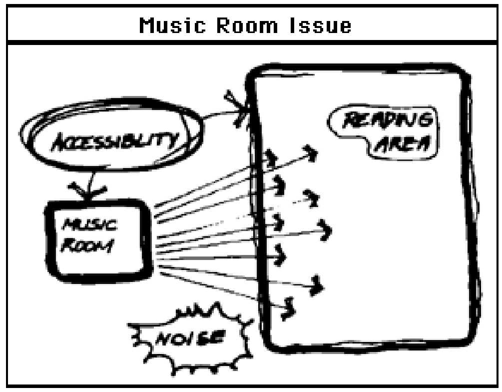


Figure 4-2. Material for Task 2a, Problem 1, a diagram with title.

The third Task (3) asked participants to match six diagrams with six textual descriptions. Two sets of three diagrams and three text paragraphs were arranged in rows and the participants were asked to pair them (Figure 4-3).

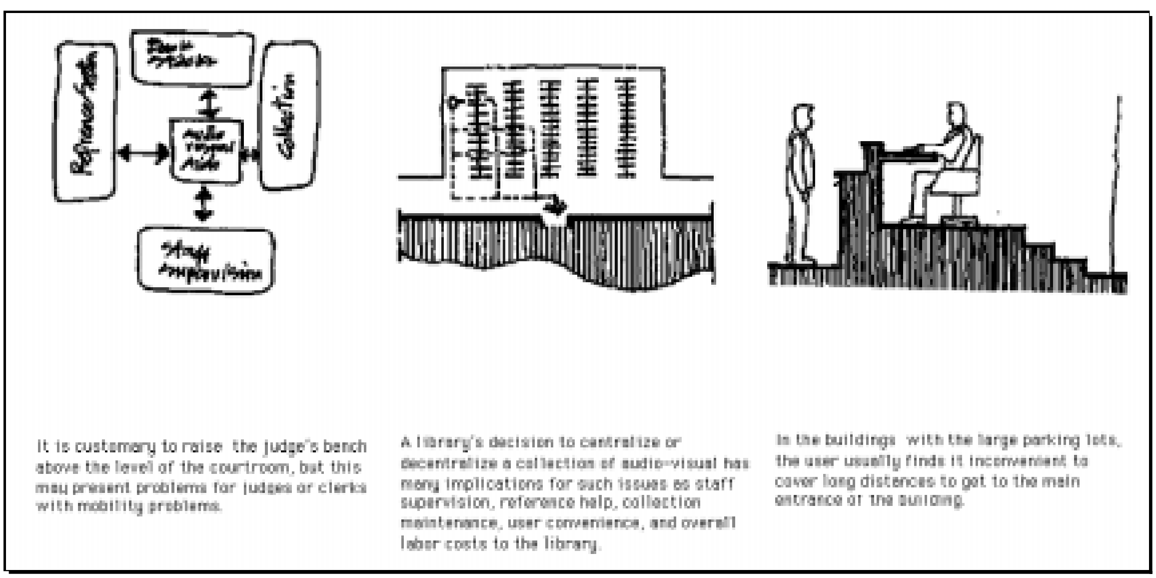


Figure 4-3. Example page for pairing test, Task 3, diagrams on top row, and texts on bottom row with different sequence.

The fourth and final Task (4) asked participants to simply comment on six diagram-text pairs (3 Problems and 3 Responses) taken from the Archie case library. For example, Figure 4-4 shows Task 4f from this task.

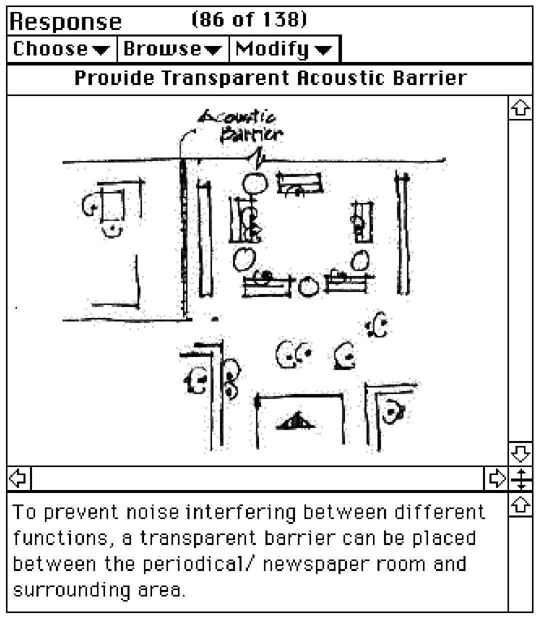


Figure 4-4. Material for Task 4f, Response 3, diagram-text set from Archie.

To determine whether the task set was feasible and to establish expected completion times, a pilot study was conducted individually with two doctoral students in architecture at Georgia Tech. Each participant was given the entire task sequence including the pre-and post-evaluation questionnaires with no time limits. The pilot participants took the same test as those in Group A. The whole task sequence took less than an hour, and the pilot participants were asked to discuss what they did and to make suggestions for future study. They suggested adding diagram examples to allow them to make better diagrams, and removing the title text from diagrams to make the test more difficult.

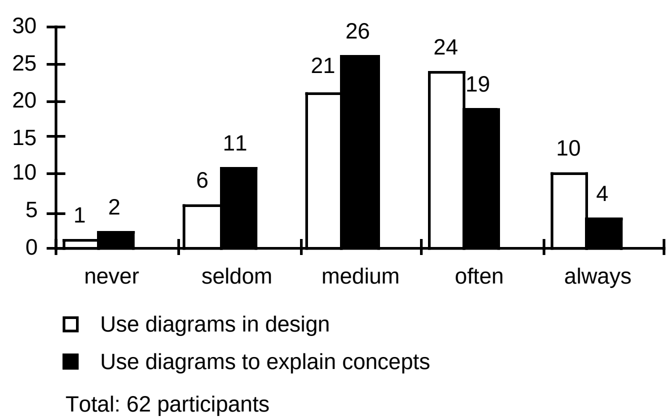
4.4. EXPERIMENT RESULTS – WHAT'S IN A DIAGRAM?

This section describes the findings from the pre-evaluations, the main experiment (Task 1 through 4), and the post-evaluations. The discussion emphasizes Task 1 since it is a diagram-making experiment. Tasks 2 - 4 were all designed as supplementary tasks.

4.4.1. Pre-evaluation Questionnaire Results

The pre-evaluation questionnaire confirmed that most participants were juniors and seniors in architecture, and had at least three semesters of design studio experience. When asked to rate how often they relied on diagrams in their design process, twenty-six participants (42%) chose the "medium usage" range, only two (3%) chose "never," and four (6%) chose "all the time." The remaining answers were spread along the midrange of the selection scale: eleven (18%) chose "seldom," sixteen (26%) chose "sometimes," and nineteen (31%) chose "often" (Table 4-4). The pre-evaluation also asked participants how often they used diagrams to explain concepts to others. Most participants said they used diagrams more frequently to explain concepts than to design.

Table 4-4. Pre-evaluation self-assessment of diagram use.



When asked about their use of computers, sixty-one participants (98%) reported using word processing programs, and forty-six (74%) reported using CAD programs. When asked what computer application they would like to see have diagramming abilities, thirty-six (58%) mentioned drawing programs, and thirty-two (52%) mentioned CAD.

Table 4-5. Pre-evaluation, self-assessment of computer program usage, applications used and expectation of application types to have diagramming tool.

	Word	Spreadsheet	Drawing	CAD	blank
applications used	61 (98%)	21 (34%)	24 (39%)	46 (74%)	1 (2%)
to have diagramming	19 (31%)	17 (27%)	36 (58%)	32 (52%)	9(15%)

4.4.2. Task 1 - Drawing Elements & Viewing Preferences in Diagram-Making

In Task 1, participants were asked to draw diagrams based on six different pieces of text, each either a description of a problematic situation or an architectural design response. There was wide variation in the level of completeness, detail, and drawing competence among the sixty-two sets of diagrams made for Task 1. For example, as shown in Figure 4-5, when asked to illustrate a text about using thick walls to reduce sound transmission between courtrooms and public areas, several participants simply drew a rectangle to indicate a thick wall. At the other end of the spectrum, some participants interpreted the task as a design task, and produced highly detailed configurations to solve the problem. Some even introduced concepts that were not given explicitly, such as putting restrooms beside the courtroom.

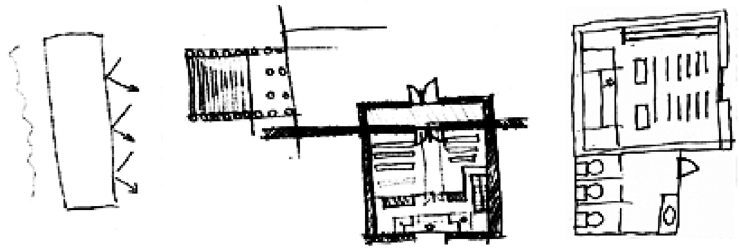


Figure 4-5. Diagrams to illustrate using thick walls to reduce sound transmission included 1) a simple rectangle to represent a thick wall, 2) detailed floor plan configuration, and 3) concepts not from text (restrooms).

An analysis of the diagrams generated in Task 1 data reveals four significant features of diagram making. First, most participants chose a plan view to illustrate relationships between different functional spaces and acoustics, but chose a sectional view to illustrate lighting conditions and sight lines. Second, participants used a limited set of symbols to represent design concepts. Third, participants annotated their diagrams with key words from the text they were given, and plan diagrams had more text annotations than sections. Fourth, diagrams frequently included labels and symbols for objects that were not explicitly given in the text. These four findings are discussed in detail below.

(1) Concepts Suggest Sectional or Plan Preferences.

The first finding of Task 1 is that participants seem to share a preference for using plans or sections to illustrate certain architectural concepts. For example, most participants chose a plan view to illustrate relationships between different functional spaces and acoustics, but chose a sectional view to illustrate lighting conditions and sight lines. Table 4-6 below shows that (except for Task 1e, Response 2) each of the six sub-tasks in Task 1 resulted in a strong preference for either plan or section diagrams.

Table 4-6. Number of plan and section diagrams drawn in Task 1.

#		Summary of text	Plan	(%)	Section	(%)	Other	(%)
1a	P1	map room location	**58**	94	0	0	4	6%
1b	P2	natural lighting create glare	3	5	**52**	84	7	11
1c	P3	fun children's section	**45**	73	14	23	3	5
1d	R1	thick walls to reduce noise	**40**	65	8	8	14	23
1f	R2	visual contact to children's area	25	40	31	50	6	10
1g	R3	glass windows for lighting	7	11	**52**	84	3	5

* # – sub-task number, P1 - Problem 1, R1 - Response 1
total participants = 62 (100%)

As shown in Figure 4-6, most diagrams illustrating Problem 1, which dealt with functional placement of a map room in a library, used plan-based representations. Also, for Task 1c, which was about fun spaces for children, plan views were most often employed to illustrate the functional arrangement issue. In contrast, the issue in Problem 2, natural lighting and glare) seemed to lead the use of sectional diagrams with symbols for light and sectional walls. Interestingly, the few sectional diagrams that were made for Problem 3 all represent attempts to solve the design problem using a change in level. In summary, a pattern of preference for plan and section diagrams was found from the experiment results. Functional arrangement problems suggest the use of plans, whereas lighting or sight-line problems suggest the use of sections.

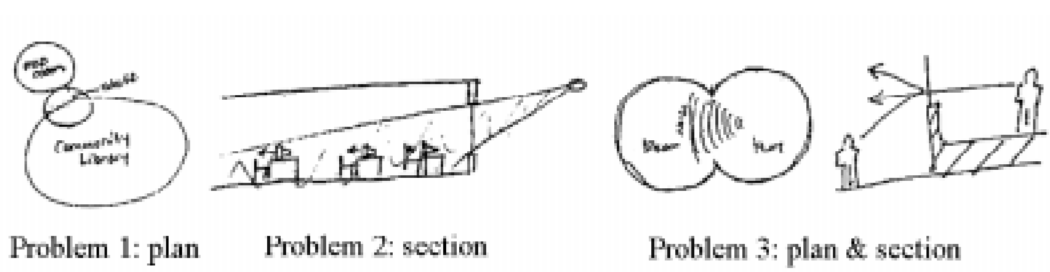
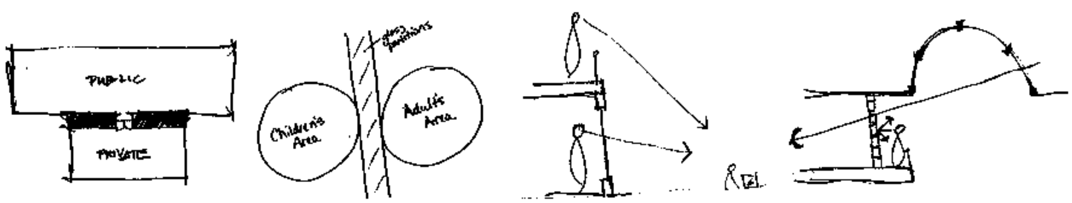


Figure 4-6. Diagram examples showing view preference for descriptions, Problem 1- plan representations, Problem 2 - sectional diagrams and Problem 3 - both plan and section diagrams.

A similar pattern of preference for plan and section diagrams was also found in the three Response sub-tasks (Tasks 1d - f). Response 1 discussed the arrangement of functional spaces and elicited primarily plan views. Response 2 was about visual contact to the children's area, which involved both the arrangement of functional spaces and consideration of visual sight lines, and hence resulted in a mix of plan and section diagrams. The diagrams used to illustrate Response 3 were also primarily sections, perhaps reflecting the reference in the text to lighting and visual contact (Figure 4-7).



Response 1: plan Response 2: plan & section Response 3: section

Figure 4-7. Diagram examples showing view preference for Response descriptions, Response 1 - plan view, Response 2 - both plan and sectional and Response 3 - sectional representation.

(2) Participants Used a Limited Set of Symbols to Represent Design Concepts.

The second finding is that participants chose primitives from a limited universe of geometric shapes and symbols to draw their diagrams, and composed these in highly conventional ways. They predominantly used lines, ovals and blobs, rectangles, and hatching. Each of these shapes and symbols were used with some small variation. For example, lines were drawn either solid or dashed, with or without arrowheads, and occasionally wiggly or curved. Ovals and rectangles were drawn with varying size and aspect ratios. Hatching was drawn in two main variations: as a set of closely spaced parallel lines, or as a single continuously oscillating line made without lifting the pen from the paper; occasionally a second hatch pattern was overlaid on the first to make a cross-hatch pattern. The basic drawing elements – the lines, arrows, and geometric shapes – are called "primitives" (Figure 4-8). Primitives are used in diagrams in a variety of different domains, and are not limited to architectural design.

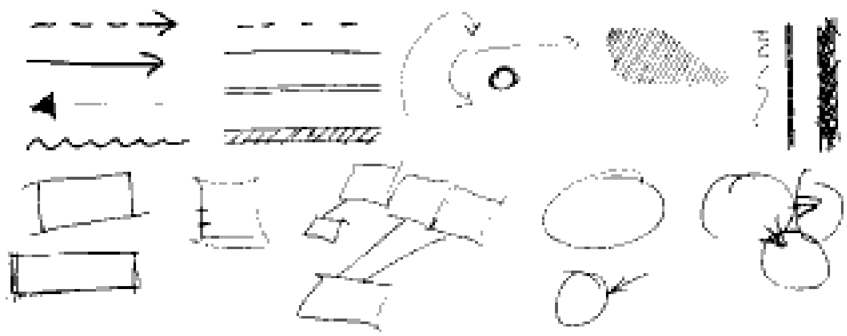


Figure 4-8. Primitives (drawing elements) used in diagrams included arrows, lines, hatches and simple geometric shapes

When primitives are combined, they can form symbols to represent architectural objects such as walls and windows or to illustrate natural phenomena such as sun and human figures (see Appendix B for a collection of the human figures). For example, a circle was composed with radial lines to indicate the sun; lines were composed to indicate walls and windows; and a circle with lines or a blob was drawn to indicate a person (see Figure 4-9).

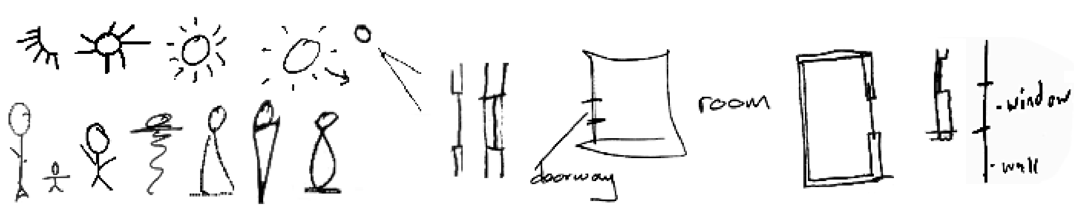


Figure 4-9. Primitives were composed to make more complex symbols.

Most diagrams drawn in response to the lighting issue in Task 1b (Problem 2. FIgure 4-10) included a representation of light from a sun passing through the building envelope. These representations were made using conventional symbols for the sun, light rays, windows and walls; some diagrams also included symbols for people and video screens.

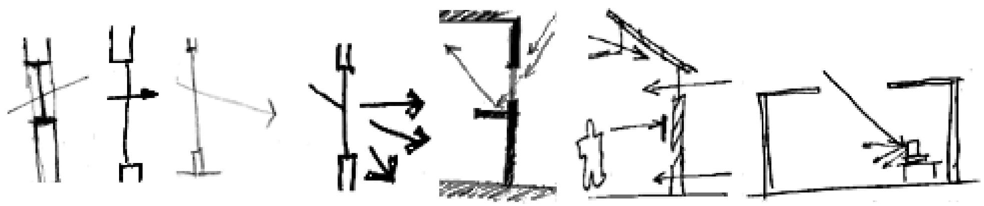
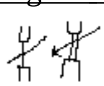
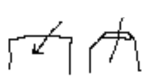
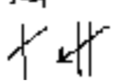


Figure 4-10. Diagrams illustrating lighting use light lines penetrating the building envelope (windows and roofs). Problem 2, Task 1.

Each symbol was drawn with small variations. Forty-one diagrams (66%) represented light rays as lines with arrows. Another fifteen diagrams (24%) represented light rays as simple lines (a total of 90% diagrams used lines). Only six diagrams did not use lines or arrows to indicate lighting. As shown in Table 4-7, representations of the building elements were highly conventional. For example, a window was shown as either a single or a double line drawn between two U-shaped incomplete rectangles, indicating the wall in which the window is lies.

Table 4-7. between group representations of light passing through building envelope.

diagram	lighting lines	A	B	C	D	all	% (of 47)
	penetrate window	6	12	6	7	31	66%
	penetrate roof	1	0	2	5	8	17%
	close to window	2	1	0	3	6	13%
	penetrate wall	0	0	0	2	2	4%

(3) Plan Diagrams Have More Text Labels Than Sections.

The third finding is that participants were more likely to include keywords from the given text as labels in plan diagrams than in sections. Of the fifteen plans drawn to illustrate Response 3 thirteen (87%) included labels, while, of the forty-two section diagrams, only seventeen (40%) included labels.

The use of labels varied significantly among the sub-tasks. Problem 3, Response 1 and 2 were about noise and acoustic privacy, produced a higher percentage of diagrams with labels. Problem 2 and Response 3, which involved lighting and sight lines, produced a higher percentage without labels.

The data from Task 1 also reveals that diagrams were labeled in different ways. Functional spaces were mostly labeled by placing text inside a shape (oval, blob, or rectangle), and only occasionally beside the shape with an arrow or line pointing to the shape (see Figure 4-11). In contrast, concept labels such as "supervise," "light," and labels of sectional elements such as "atrium" and "corridor," and labels of material, such as "walls," "frosted glass" were usually placed beside the shapes with or without a pointer (line or arrows). For example, in Problem 1, forty-seven (76%) participants wrote the word "map" as a label in their diagram; twenty-two (35%) wrote "staff," eighteen (29%) wrote "library" and six (10%) wrote "supervise." The text presented in Problem 1 is shown below, with these words that appeared in diagrams in uppercase:

> In a community **LIBRARY**, a separated special **MAP** room allows control over access to valuable material but if out of sight of normal staff areas may require an additional **STAFF** member to **SUPERVISE**.

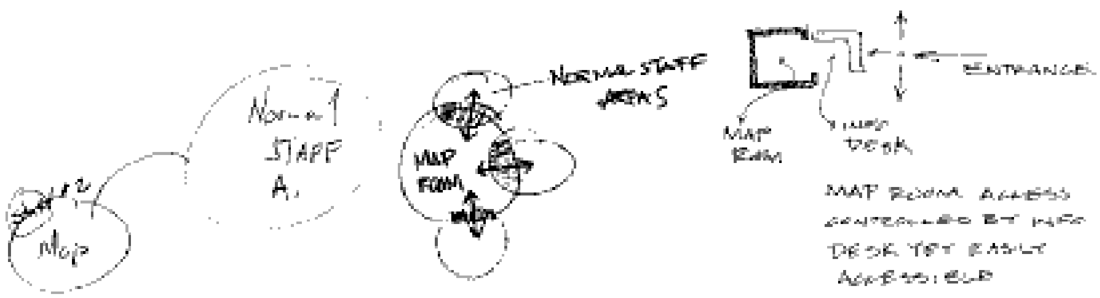


Figure 4-11. Diagrams use keyword "map" given in the text to label a space (Problem 1 of Task 1).

Response 1 of Task 1 described putting thick walls between courtrooms and public spaces. The two major space names were labeled on most diagrams (Figure 4-12). Spatial arrangements are illustrated with text inside ovals or rectangles. Thirty-five participants (56%) put down "C" or "Courtroom" and twenty-nine (47%) wrote "P" or "public." The text presented in Response 1 is shown below, with these words that appeared in diagrams in uppercase:

> Use thick walls between **COURTROOMS** and **PUBLIC** areas to reduce sound transmission.

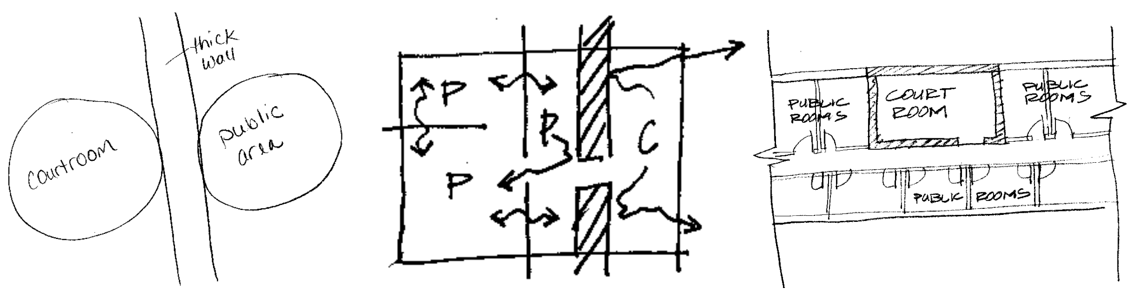


Figure 4-12. Diagrams use keyword "public" and "court" or "P" and "C" from given text to label space, Response 1, Task 1.

(4) Diagrams Included Labels and Symbols not Mentioned in The Text.

The fourth finding is that diagrams included elements – both text labels and symbols – that were not explicitly referred to in the text of the problem or response. For example, in response to Task 1b (Problem 2), which was about natural lighting and glare, a significant number of diagrams included representations of people, desks, chairs, and the building envelope, even though these elements were not mentioned in the text. Similarly, text labels in many of the diagrams also identified additional or extraneous elements. For example, in response to Task 1f (Response 3), which was about the use of frosted glass on corridor windows, text labels included "hall," "public," and "space." Although these labels are plausibly related to words in the given text, participants also included such diverse annotations as "President's office" "Prz Bath," and "small trees."

Responses to Problem 3 included the previously mentioned words "read," "children" and "library" as labels for spaces in the diagram. However, other words not mentioned in the text such as "play" and "kid" were also used as annotations. In Response 1, "Court" and "Public" (or sometimes just "C" and "P") occurred frequently in the diagrams, while three participants (5%) also included the unmentioned word "private." In Response 2, the previously mentioned words "atrium," "corridor" and "frosted glass" were included in addition to the unmentioned terms "space," "hall" and "public."

4.4.3. Task 2 - Consistent Interpretation of Diagrams among Designers

The six sub-tasks in Task 2 asked participants to write a story from a given diagram. Participants were provided guidelines to help them understand the text writing style. Though the wording and writing style of the participants varied, the sixty-two sets of stories were quite similar. Only one or two participants interpreted the diagrams are markedly different than the others. For example, when given diagrams consisting of bubbles with labels and arrow lines (about the relation of noise and accessibility to a music room and reading area), fifty-five participants (89%) introduced the concept of separation, writing that accessibility leads to adjacency of the two areas and noting that concerns about noise transmission should be addressed by a soundproof buffer.

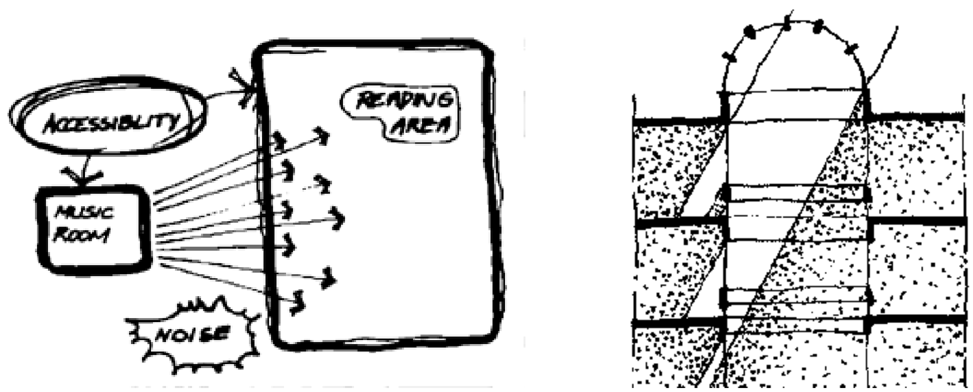


Figure 4-13. Sub-tasks for Task 2, create text from diagrams. (a) Music Room Issue. (b) Overheating caused by Overhead Glass.

The Task 2, data revealed two interesting features of how diagrams are read. First, symbolic conventions enabled participants to recognize and make stories. For example, in Problem 2 (Figure 4-13b), they interpreted the parallel filled lines as floors and the arc lines on top of the structure as a glass-enclosed atrium (thirty-eight participants, 61%). They saw the diagonal lines cutting through the buildings as sun light (fifty participants, 81%), and dotted hatching as spaces in shadow (forty-two participants, 68%). Second, participants used labels from the diagrams as keywords in the text descriptions they wrote but also words and concepts that were not explicitly given in the diagram. For example, in Problem 1 (Figure 4-13a), all participants included the words “music” or “noise” in their written text.

4.4.4. Task 3 - Designers can Easily Associate a Diagram with Its Pairing Text

Task 3 included two sets of diagram-text pairing exercises. Each set had three diagrams on the top row, and three texts on the bottom row. Participants were asked to draw lines connecting text descriptions with diagrams they described.

This task appeared to be the easiest to perform. Only two participants (out of 62) made inappropriate connections and their comments yield clues to why they failed. As the diagrams provided no indication of scale or labeling, elements like the division lines of a parking lot (see Figure 4-14 and Figure 4-3) could be misinterpreted as bookshelves in a library floor plan. Such a misinterpretation is especially plausible since he text about the library (Figure 4-3, second column text) appeared prior in the test sequence to the text about parking (Figure 4-3, third column text).

Figure 4-14. One participant failed to associate this diagram with its text because parking lot division lines look like bookshelves.

In the follow-up discussions, participants made interesting comments. For example, some suggested removing diagram labels to make the task more difficult. Some suggested that using symbols to illustrate a concept would make a diagram easier to read. For example, "no handicap access" could be illustrated with a restricted traffic sign showing a person and wheelchair inside the circle. Two participants commented that three-dimensional views or sections would be helpful for the Response 2 diagram (Figure 4-15) about "varying form and layout." A few participants circled a part of the diagram that could have had potential problem, and drew a line to write down the problem, while others underlined or circled words in the text and drew arrows out to comment further. Some participants interpreted the task as a design task, and made suggestions to solve the problem.

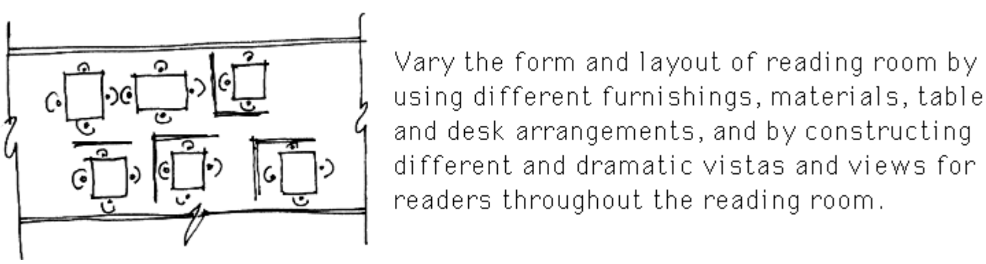


Figure 4-15. Participants suggested changing this diagram to a three dimensional view (Response 2 of Task 3)

4.4.5. Task 4 - Designers Agree with Archie Diagram-Text Pairs

In Task 4 participants were asked to comment on six diagram-text pairs from the Archie case library. They were asked to draw diagrams or comment on given pairings of diagrams and text description.

Table 4-8. Comments on Archie's text and diagram pair.

comment	P1	P2	P3	R1	R2	R3
fit	68%	90%	91%	95%	84%	81%
not fit	8%	3%	2%	0	0	0
no comment	24%	8%	6%	5%	16%	19%

*P1 = Problem 1, R1 = Response 1

The main finding in Task 4 is that participants felt that the text and diagrams were well paired, which suggests that people can understand and agree on other people's illustrations of text stories. For example, in Response 1, fifty-nine participants (95%) simply wrote "good fit," "OK," or "fit" to comment on the given pairing (Figure 4-16).

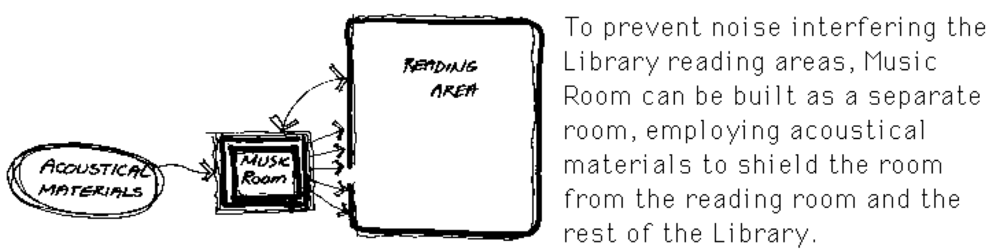


Figure 4-16. Diagram showing music room issues, given for Response 1 of Task 4.

An analysis of the Task 4 data reveals four features of diagram making. First, most participants read the diagrams clearly without much labeling, although several participants noted that labels help reading. Second, some participants suggested a sectional rather than a plan view to illustrate sight line relationships in Response 2 (sheltered waiting area remains visible). Third, participants had a tendency to read diagrams from left to right and to interpret diagram elements as real physical objects. For example, participants commented that the Response 1 diagram (Figure 4-16) looks like a mathematical formula A + B = C but still suggested that it would read better if the far left bubble labeled "acoustic material" were moved between "music rooms" and the "reading area." Some also suggested that the arrow lines representing noise should be kept within the "music room." Fourth, similar to the results of Task 1, concepts were illustrated in a conventional way, such as by using concentric arcs or lines with arrows to represent "noise," and by using lines with arrows to indicate either a circulation path, visual access, or even noise.

4.4.6. Post-Evaluation Questionnaire Results

The post-evaluation questionnaire asked participants to write and give examples of diagram elements. Only thirty-eight (61% of the total 62) participants answered this question. As shown in Table 4-9, thirty participants (79%) used lines and twenty-five indicated that wall lines are represented either with single lines or double lines with hatches between them. Twenty-four (63%) participants used ovals and bubbles, and twenty participants used rectangles. Nineteen participants (50%) wrote the word "bubble" or "bubble diagram" in their answers which indicates an understanding that diagrams as "bubble like."

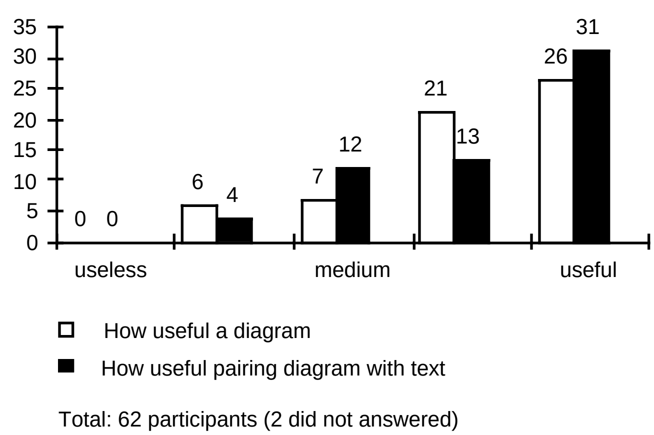
Table 4-9. Post-evaluation self-assessment of diagram elements used.

Type	arrow-line	line	bubble	rectangle	total answered
Participants	23 (61%)	30 (79%)	24 (63%)	20 (53%)*	38

* percentage are calculated based on the total of participants answered (38 = 100%)

Participants described using simple geometric shapes to form symbols for human figures, building elements (e.g., walls, windows, doors, desk and chairs), natural objects (e.g., sun, tree and mountain), and phenomena such as light and noise. They defined arrow lines to represent gradients, flow within a space, movement, circulation, leads-to, entrance, car path and motion. Lines are described as representing circulation, separation, and linkage, and for illustrating parts and hierarchies. Hatches indicate wall material and insulation. Horizontal lines are floors, while horizontal lines with hatches underneath means ground floor.

Table 4-10. Comment on how a diagram and a tool provide both diagram and text useful to convey ideas.



Participants agreed that diagrams are useful to convey ideas (Table 4-10). They described the use of diagram as "to understand and design," "to record and understand," "to clear ideas in mind," and "to illustrate or find a solution." Forty participants (65%) were certain that they could read other people's diagrams. Only five participants thought diagrams are only meaningful to those who draw them, and only two thought sketch and diagrams are "personal." One commented that not all relationships are diagrammable and "action" is particularly difficult to show. Some commented that a diagram is more important than text, because it gives more information; two also commented that diagram and that text should always be combined, and text can not stand by itself.

4.5. DISCUSSION – IMPLICATIONS FOR DIAGRAMS AND COMPUTING

The experiment results support the hypotheses that diagram are used in a consistent way to represent different issues in architecture. First, participating designers only used a small set of symbols in their drawings and arrange them in conventional and consistent ways. Figure 4-17 shows the lexicon of symbols they used to represent different design concerns. For example, a small set of symbols is used in conventional ways to represent sun, lighting, sight lines, noise, furniture, walls and windows, doors, and people. Second, designers exhibited different view preference (e.g., plans, sections, etc.) for illustrating different sorts of problems (e.g., spatial arrangement versus getting light into a building). Third, keywords from the stories are often used as labels in diagrams, and vice versa. Fourth, designers mostly agree with each others' diagrams; they interpret other designers' diagrams or texts as they were intended to be interpreted. These observations suggest that a computer-based diagramming tool would only need to provide limited pattern recognition facilities.

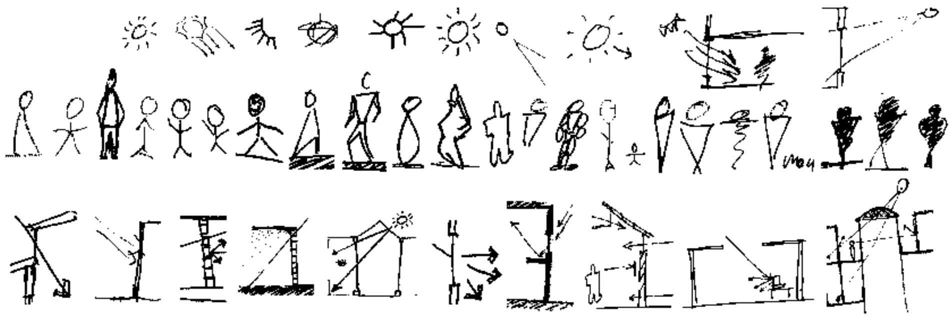


Figure 4-17. Designers used conventional symbols and configurations for architectural concepts in diagrams.

4.5.1. Designers' Shared Understanding of Diagrams

Designers seem to share an implicit understanding of the definition of diagram, even if it is unclear in their own minds. Some participants commented that they made sketches instead of diagrams, and some asked for a definition of "diagram." Some participants suggested that definitions of these terms should be included in the test material to ensure that they make the "right kind" of diagrams. However, most participants described the had little problem describing the elements in the diagrams they were given.

4.5.2. Future Experiments

The participants in this experiment were students with at least three semesters of design studio experience. Perhaps reflecting the range of participants' design studio experience, the task data shows a wide range of responses. However, perhaps due to their common background (all were students at the University of Colorado), the diagrams and texts had significant number of common features. A pre-test with two Georgia Tech Ph.D. students, who both did their undergraduate studies outside the United States, yielded similar results. If the common features are due to design education, then at least it is fairly universal. A future study could use a broader range of subject material and a more diverse population of participants, including design professionals, to explore the universality of diagram making and diagram reading. A similar test using material from different sources might reveal a different symbol universe. To more accurately and efficiently analyze the results, the next experiment might also be conducted with fewer participants and fewer questions, or be conducted in an automated way.[2]

These experiments also raise questions that could be explored in further experimental studies. First, the result suggest that diagrams are a simplified representations of complex relationships. What purpose, then, do diagrams serve in the design process? Second, the result suggest that architectural diagrams illustrate natural and artificial phenomena, building components, and human behavior. Are diagrams limited to these concepts? Third, what is the structure of a diagram? Is there a syntax for diagrams? Fourth, do diagrams reflect specific theoretical positions in design? For instance, do diagrams reflect particular educational or training background, or different stylistic approaches for generating and arranging forms? Fifth, to what extent are diagrams a matter of individual style and preference?

4.5.3. Future Work – Diagrams as an Index to Design Information Systems

This chapter began by reporting a study on the use of design diagrams in order to gauge the feasibility of using diagrams to access a design information system. The experiment found that diagrams are used in a conventional fashion, which supports the

[2] One might also automate the experiment, using a computer-based diagrammer to collect and analyze data (though some of the analysis process may be hard to automate).

feasibility of computer support for visual note taking and for indexing a database of design information. This study shows that designers can read other designer's diagrams and that they often use symbols to illustrate their ideas. In the experiment, many participants agreed that diagrams are useful and that existing Draw and CAD programs should provide the ability to build diagrams. This suggests that a computational environment that supports and recognizes hand drawn diagrams would be useful.

Chapter 3 reported a prototype system linking a diagramming tool called Archie's Napkin. That tool recognizes a small set of diagrams that serve as symbol indexes for cases. For example, a designer who is concerned about connecting two functional spaces might draw two circles or shapes connected by a line. The system uses the diagram to find stories about how corridors connect two rooms in the background while the designer remain engaged in design drawing.

To understand more about the role of drawing in design, one needs to look also at sketches and other types of drawing, not only at diagrams. Chapter 5 reports a design experiment to find out the relations between design intentions and design drawing. Different design intentions need different kind of computational support. Therefore, Chapter 6 reports an exploration on automatic activation of different design tools based on the design intentions detected from the design drawing.

CHAPTER V

DESIGN DRAWING EXPERIMENT – DESIGN CONCERNS & DRAWING CONVENTIONS

This chapter describes a protocol study designed to get at the relationship between design concerns and drawing conventions in the design process. In the study described in Chapter 4, several drawing conventions are identified with respect to specific design contexts and intentions (Do, 1995a). For example, attention to lighting is often indicated by arrows, attention to functional arrangements with bubble diagrams. However, this previous study only deals with diagrams drawn to illustrate given design concerns, not those drawn in the context of a real design task. This raises an obvious next question: Do designers use the same drawing conventions when thinking about design concerns in the context of design? One possibility is that designers perform different activities in the process of design than those captured in the previous experiment. Another is that they may be differently motivated but share the same drawing conventions. In order to answer the questions, a new experiment was conducted in which participants were asked to perform simple design tasks while attending to different design concerns.

The hypothesis is that designers use common drawing conventions when they design. Moreover, the conventions used will depend on the design concerns being considered. If this is true, the drawing conventions used in the previous chapter should be similar to the conventions used by designers in this experiment. The goals of the experiment are thus 1) to verify the findings of the previous study about drawing conventions in a design process, 2) to further investigate design context and intentions as they are reflected in drawing conventions, and 3) to establish a record of the experiment to demonstrate the feasibility of making drawing convention computable.

The next section describes the experimental setup, including the design program and tasks that were used to focus on different design concerns. Section 5.2 gives a description of the participating designers: their age, design experiences and their use of freehand drawing in design. Section 5.3 reports the four design sessions that were recorded. Section 5.4 analyzes the sessions and Section 5.5 discusses the findings. Finally, Section 5.6 summarizes the result and draws conclusions from the experiment.

5.1. THE EXPERIMENT – SETUPS, TASKS AND PROGRAM

Two undergraduate design students and two architectural instructors served as subjects for the experiment[1]. The test material given to the participants was a brief calling for the design of an architect’s office design. The brief involved a sequence of four tasks (five for the two student participants), where each task asked the participant to focus on a particular issue. The activities of the undergraduate students were recorded by hand on

[1] I asked only two undergraduate students and two instructors to participate the design drawing experiment because the goal was to verify if their drawing conventions would be similar to the previous study of sixty-two design students. I did not intend to look for a statistically significant result from this study. The hypothesis about drawing and design concerns would be considered valid as long as the conventions found from the new experiment are a subset of the previous findings.

letter size white paper. The activities of the instructors were recorded on videotape while an observer took notes on the designers' remarks and drawing actions. All drawings made by participants were captured on white paper and either yellow or white tracing paper.

The main question is whether one can tell what designers are thinking by looking at what they draw. Therefore, the experiment was designed around a design exercise with a real program that allowed participating designers to use freehand drawing while designing. The experiment was set up to obtain the designers' drawings made in response to each individual task. Video recording was introduced in the last two sessions with architectural instructors to gain more evidence from the verbal descriptions of design intentions.

The sequence of the experiment was as follows: 1) the task was given to the participants on a written instruction sheet accompanied by a brief oral explanation, 2) participants were given 5 minutes to complete a pre-experiment questionnaire that covered personal information about design experience and their drawing preference, 3) participants were given 10 minutes to perform the first task, and then 5 minutes each for the rest of the design tasks; and finally, 4) at the end of the design sessions, the participants were asked to comment on their drawing symbols and on the experiment itself by filling out post-experiment questionnaires. The combined tasks (including both the design problems and questionnaires) were designed to take 30 minutes to an hour total.

Office space design[2] was chosen as the focus of the design task in order to determine how symbols were associated with space arrangement, lighting, visual access and dimensioning. The design program was

---

[2] A program of an architect's office space design was chosen for two reasons: 1) participating designers would either have experience of their own office design or planning schemes so the exercise would be an easy task, 2) as Wineman describes in "Behavioral Issues in Office Design"(Wineman, 1986), the layout

> *"to design an office for an architecture firm in a 70 ft. by 25 ft. one-story warehouse, providing workspace for architects, CAD operators, contract draftsmen, a secretary and student interns. The office will be designed to have space for work groups, a meeting room, a small kitchenette, a bathroom, and a chief architect's private office, a general affairs section, storage space, printing and plotting area."*

After reading the design program, designers were asked to start with a new sheet of paper (or tracing paper) for each task and to focus on four different concerns in conceptual, schematic design: 1) spatial arrangement, 2) lighting, 3) visibility and privacy, and 4) fitting a special piece of furniture into the design. Undergraduate participants Roger and Noi were given five tasks while instructors Samuel and Mario were given four. Tasks 1 and 2 for the undergraduates were both concerned spatial layout and therefore were merged as Task 1 for the instructors. Tasks 3, 4, and 5 for the undergraduates then became Tasks 2, 3, and 4 separately for the instructors.

For the undergraduate students, Task 1 asked them to make a conceptual, schematic design for the program, and then Task 2 asked for the spatial zoning of the office, with the suggestion that they consider how to arrange the lobby, meeting area, different work groups and service area. For the design instructors, Task 1 asked them to pay particular attention to the spatial zoning of the office – that is, where to put the lobby, the chief architect's office, the meeting room, the different workgroup spaces, and so forth. Task 2

---

and design of office space influence the workers' behavior and that workspace design is more than just simple decoration. An office space design is an important design task. I do not intend to discuss the psychological aspects of the office space design in this thesis, however, the finding of design process might yield useful information for future investigation.

(Task 3 for undergraduates) asked them to focus on lighting for the meeting and working area. Task 3 (Task 4 for the undergraduates) asked them to consider visibility and privacy between different spaces. Finally, Task 4 (Task 5 for the undergraduates) was a special request for fitting a large meeting table into a conference room and a minimal square footage requirement for designers' work space.

Participants were told they had 10 minutes to complete the first task, and 5 minutes for each successive one. However, the time limit was not strictly enforced. The two undergraduate students followed the instructions carefully and observed the time constraints while the two instructors simply ignored the time constraints and spent about half an hour each to perform the first task. Therefore, the time spent by participants for the entire design experiment ranged from thirty minutes to one and a half hour. All of the participants told the observer that they enjoyed participating the design experiment. The two participating instructors even commented that they were ready to make detailed design and physical models after they completed the design experiment. Section 5.5 summarizes the post-experiment questionnaires..

5.2. PROFILES OF PARTICIPATING DESIGNERS

The pre-experiment questionnaire asked designers to fill in basic information – name, age, gender, and occupation. Participants were also asked to 1) briefly describe their architectural design experience, 2) wrote down how many years of education (college and graduate school) the had completed, and 3) report how often they used freehand drawing for architectural design on a scale from 0 to 4, (0 means never, and 4 means all the time), and finally, 4) report on how often they used freehand drawing to communicate with other designer, also on a scale of 0 to 4. (See Appendix C for consent and release form used in the experiment, and the pre-experiment questionnaire).

Table 5-1. Profile of participating designers

Name	Roger	Noi	Samuel	Mario
nickname[3]	Functional Designer	3D Sketcher	Philosopher	Research Architect
age	21	20	28	51
experiment date	09/16//96	09/17/96	02/12/97	06/18/97
design education	3 years	4 years	5 years	13 years
professional experiences	3 internships	1 internship	1.5 - 2.5 years in 3 architecture firms, & 1 year private business	3 years in architecture firm, & 15 years consulting business
use of freehand drawing in design (0 - 4) *	3.3 (right handed)	4.0 (left handed)	3-4 (right handed)	4.0 (right handed)
use freehand drawing to communicate	3.5	4.0	3-4	4.0
use computer for design (0 - 4) *	3.0	4.0	3-4	4.0
post-experiment self report on personalized symbols and short hands for designing	window, wall, door, entry, movement, link, height, dimension	tree, plant, human	circulation, light, visibility, thick walls, thin transparent screen	solid wall, chairs, people, dogs, plants, kitchen, windows, computers, roof

* (0 never - 4 all the time)

Based on the pre-experiment questionnaires, all of the participating designers have at least three years of design studio education in college with some experience working at architectural firms. They all used freehand drawing frequently in their design process and to communicate with other designers. Therefore, all four of the design sessions were considered valid and relevant to the research despite possible occasional outlying results.

[3] Caricature description, nickname are included here to associate with designers' first names (these are not their real names but bear same initials) through out this thesis to ensure easier differentiation between designers. The caricature in no way intended to contain any negative connotation.

Table 5-1 shows a summary of designers' profiles. A brief description of each participant is given below.

Roger was a graduating senior at the University of Colorado with a high grade point average in design-related courses and studio work. He had spent all his summer vacations in college working in architectural design firms. He was twenty-one when he participated in the design experiment. He was invited by different instructors to give talks to architectural programming classes and studios for the first year design students. He used freehand drawing all the time for designing and to communicate with others. He believed he produced good design by making sure his design work fulfilled "functional" aspects. Therefore, Roger is referred to as the "Functional Designer."

Noi was twenty years old, also a graduating senior when he participated in the design experiment. He came to the University of Colorado after high school education in Thailand, and a "career discovery course in architecture" at the University of Oregon. He spent a summer intern at a local architect's office. He reported an interest in lighting design and intended to look for lighting related jobs after he graduated from college. He used freehand drawing for "everything" that includes design and communication and he was proud that he had good sketching skills and the ability to draw "3D" sketches from any drawings. Hence Noi is referred to as the "3D Sketcher."

Samuel was teaching two courses at the College of Architecture and Planning at the University of Colorado when he participated in the experiment. He was twenty-eight with a Bachelor degree in architecture. He has just started a private business for a year as a free lance designer. Previously, he had worked for three different design firms. He also designed and worked on a 4800 sq. ft. house on a local creek. He used freehand drawing all the time, and loved to draw to communicate with other designers. He also made good pastel and color pencil renderings. He was a philosophy major and a professional rock climber before he chose to study architecture. He believed that everything that appeared in a design should be justified. Therefore, Samuel is nicknamed the "Philosopher."

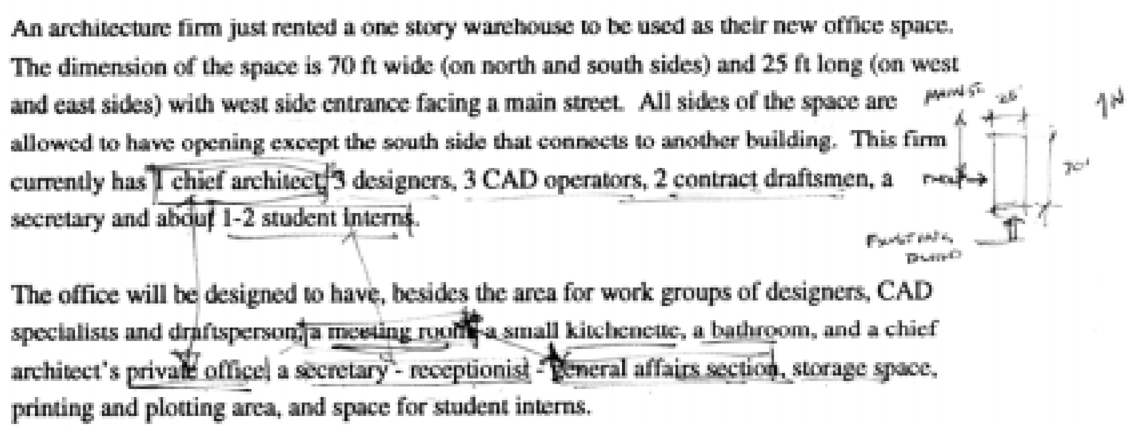
Mario was a visiting scholar from Mexico. He had professional degrees in architecture from universities in Mexico and the United States. He obtained his Ph.D. from University of California at Berkeley. He was fifty-one when he participated in the design experiment. He had three years experience in an architectural office in the U. S. and fifteen years experience in a private consulting business. He frequently used freehand drawing for architectural design and communication. He called himself a "Research Architect."

5.3. SUMMARIES OF DESIGN SESSIONS.

The following briefly describes what happened in each design session. A detailed analysis of the sessions and each participant's reasoning process is included in the next Section 5.4.

5.3.1. Roger, The Functional Designer

When the experiment started, Roger made notes when reading the program. He drew a small site plan (Figure 5-1 right) to indicate the dimensions [70' x 25'] of the site, made symbol to indicate direction North [arrow up and letter N], a line with double arrows to indicate the main streets, a single arrow line to annotate where the entrance would be, and a double-lined arrow for the adjacent existing building [on the south side of the site]. He then underlined and drew boxes to frame different job descriptions [personnel information: 1 architect, 3 designers, etc.], spatial and functional requirements [meeting room, bathroom, office, etc.], and drew arrows to link related spaces together. For example, he drew double-arrow lines from “chief architect” to private office, “student interns” to secretary and receptionist area, and “meeting room” to general affairs section (Figure 5-1).


An architecture firm just rented a one story warehouse to be used as their new office space. The dimension of the space is 70 ft wide (on north and south sides) and 25 ft long (on west and east sides) with west side entrance facing a main street. All sides of the space are allowed to have opening except the south side that connects to another building. This firm currently has 1 chief architect, 3 designers, 3 CAD operators, 2 contract draftsmen, a secretary and about 1-2 student interns.

The office will be designed to have, besides the area for work groups of designers, CAD specialists and draftsperson, a meeting room, a small kitchenette, a bathroom, and a chief architect's private office, a secretary - receptionist - general affairs section, storage space, printing and plotting area, and space for student interns.

Figure 5-1. Functional designer Roger’s process of understanding the design program: (a) part of the program was illustrated on the right side to represent site, and (b) personnel and space requirement in the program were underlined, framed by boxes, and linked by double arrowhead lines to indicate relationships.

When performing schematic design, Roger first drew a rectangle to represent the site, and then wrote down dimensions on the sides [70’ and 25’]. Then he drew bubble diagrams to represent different spaces, and wrote words to label the spaces, such as “SUPPORT”, “STORAGE”, “BATH” “OFFICE”, “MEETING R(oo)M”, “WAIT(ing area) , SEC(retary) and “KITCHEN”.

Then he overtraced the bubbles, added some arrows and straight lines to indicate where the wall, corridor and the room boundary would be. He drew double arrows to connect spaces (the same kind he used to link relevant space and personnel information in the design program) and talked about circulation concerns (Figure 5-2).

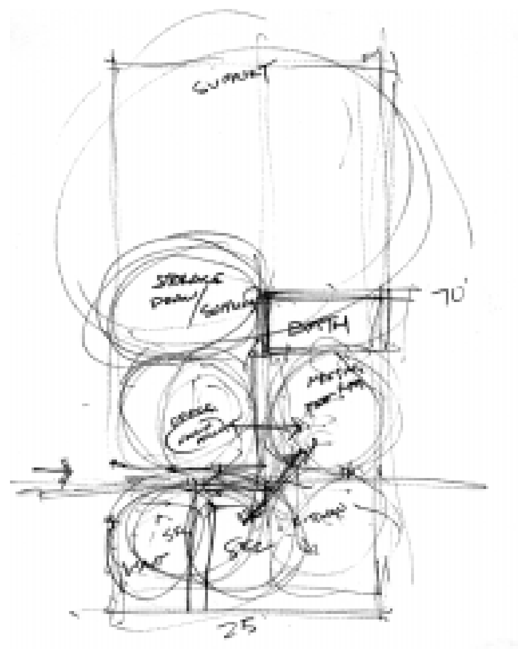


Figure 5-2. Functional designer Roger's bubble diagram showing conceptual design for spatial arrangement for Task 1.

For Task 2, Roger went on to explain the details of each room, and drew furniture to think about the space. First he designed the lobby and secretary area (Figure 5-3a) by drawing an arrow for entrance (he labeled it "ENTRY"), then a round coffee table and sofa, with some counter and desk space. He double lined the partitions showing where the walls and the door to the hallway would be (an arrow pointed up between the walls to indicate circulation path). Second, he drew the architect's office (Figure 5-3b), including a window, an entrance, chairs, a computer and two tables. He circled the computer in the corner to explain it to the observer. He indicated that the long side table on architect's right can serve as a small conference table when discussing with client (he wrote "CONF"). Then he went on to design a conference room, started with a long rectangle (to define the room boundary), and then filled the room with an oval-shaped table, nine chairs and a podium, with a blackboard behind it (Figure 5-3c left). He added a door (labeled “ENTRY” with an arrow), wrote down “HALL WAY” outside of the room (rectangle). He double lined the wall to the hall way and made window mullions on it. He then added some supporting facilities, such as a counter with a sink and a refrigerator [“for refreshment”] and maybe a cabinet [“for overhead projector equipment, etc.”]. He labeled “10” and “8” on the two long walls of the rectangle room to indicate different overhead ceiling heights (Figure 5-3c), then he drew a section to show that a skylight could come in from the taller ceiling height that introduced a window in a sectional view (Figure 5-3d).

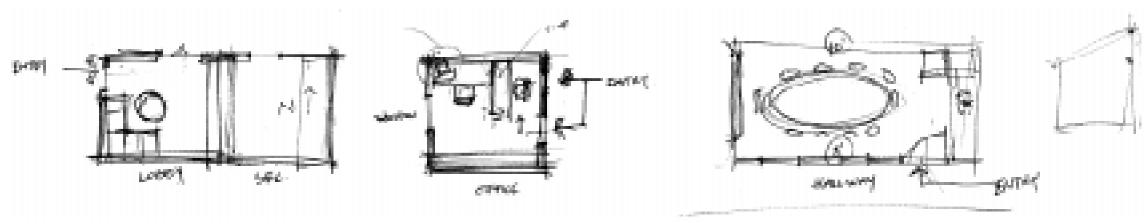


Figure 5-3. Roger’s drawing for Task 2: (a) a lobby and receptionist area showing a round table with couch, and secretary area with table and counter, both have opening to inside space. (b) a chief architect's office showing 1 table with a computer on it, a chair for the architect, a side table serves as a conference table with a chair for client. (c) conference room, showing an oval shape meeting table with surrounding chairs, wall with windows, a door to the hallway, supporting facility such as a counter with a sink, a refrigerator (on the top right corner), a podium and a blackboard (on the left), and then (d) a section showing different ceiling height (10' and 8')

The next task (Task 3) focused on lighting. Roger first sketched out the floor plan spatial layout with labels of space names, then he created a courtyard to indicate how every space can view the courtyard and receive sunlight (Figure 5-4a). Then he drew a sectional plan with a symbol of sun, a line with arrowhead to illustrate light ray coming into the building, a person sitting on chair working on a computer (Figure 5-4b). He explained that the light should not fall directly on the computer screen or it may cause glare.[4]

[4] This was an interesting and meaningful comment. What Roger’s concerns about glare on computer screens was a common problem reported from real building occupants. In Archie case library that we worked

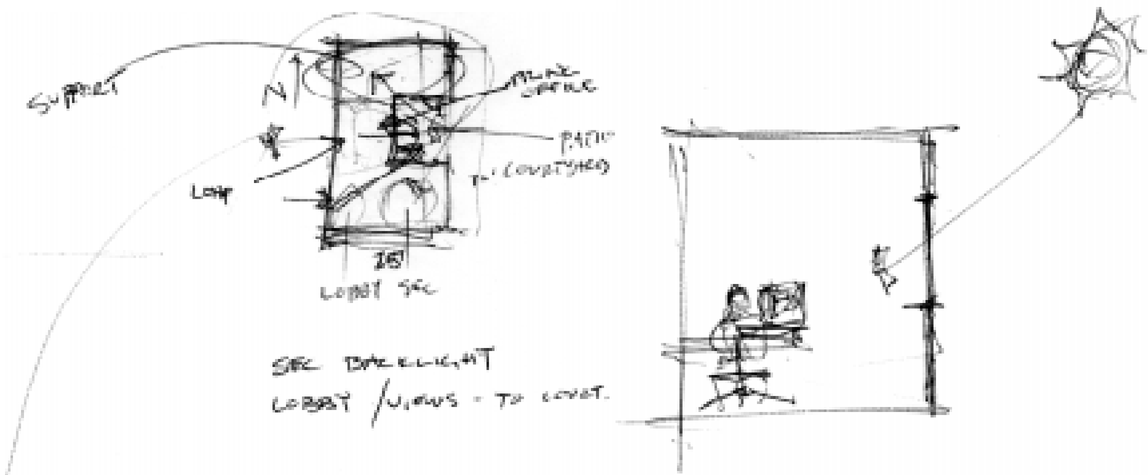


Figure 5-4. Roger’s drawing for Task 3, lighting: (a) the design of a courtyard place that allows lighting and visual access to all space surrounds it. (b) a sectional drawing depicting a sun with light ray comes in through the window to a room where a person sits and works with a computer.

For Task 4, which concerned visibility and privacy concerns, Roger’s first response was that he had already done it. He wrote down “SEE TASK 1” right under the task statement. He then decided to use another paper to blow up the plan view he made previously and make it more “room” like, instead of bubbles (Figure 5-5). After he labeled the rooms, he wrote his rationale next to the drawing: 1) BATH(room) accessibility between all areas, centralized with buffer, (he drew a solid line and labeled it “WALL”), 2) CONF(erence room) next to BATH(room) this avoids contact between clients and drafts(man), 3) PRINC(iple architect) next to CONF(erence room) to oversee what’s going on between firm and client, near LOBBY for VISUAL REF(erence) .

on before (described in Chapter 2), there are post occupancy evaluation (P.O.E.) cases about how lighting create glare on computer monitor screens. This drawing and explanation provided evidence that the designer was thinking about a particular design concern. The designer could benefit from the similar design cases if Archie case library is made available at this moment.

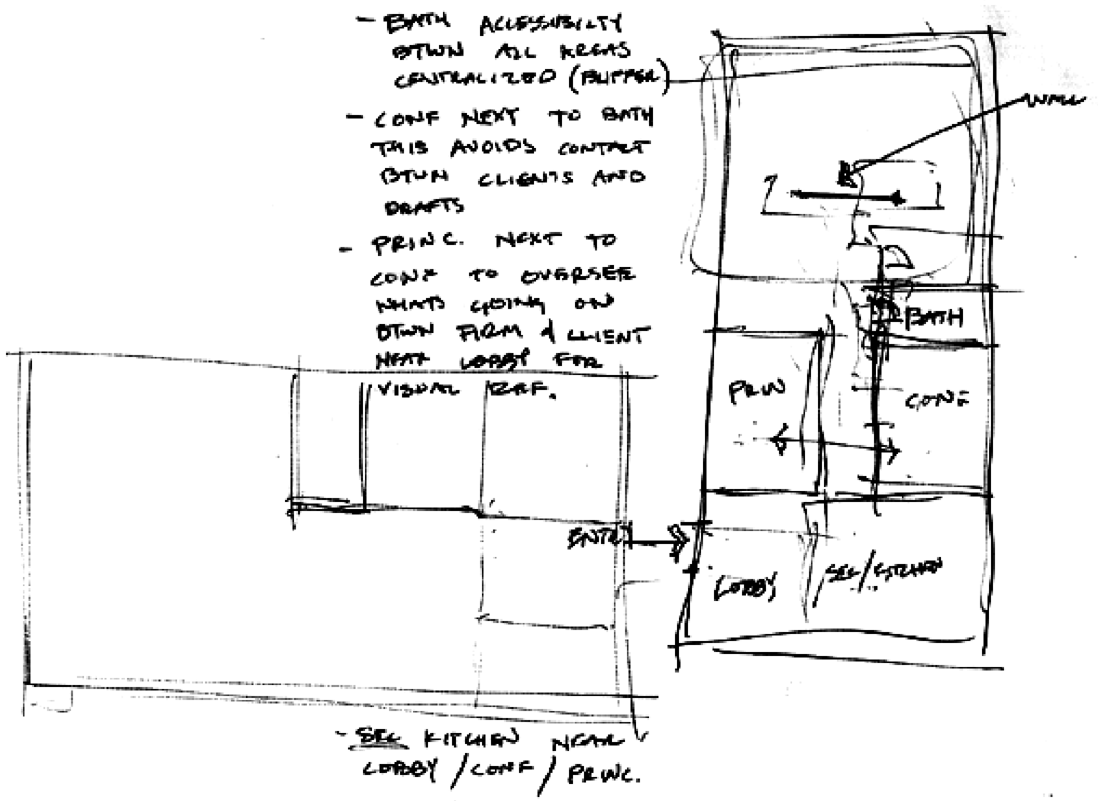


Figure 5-5. Roger’s drawing and notes explaining design rationale for the spatial arrangement.

The final task (Task 5) involved spatial requirements. Roger wrote down numbers for dimensions, reasoned about the size of the space and checked the requirements. For example, Task 5 called for a work space of at least 800 square feet, and the conference room had to accommodate a 4' by 10' table. Figure 5-6 shows one of Roger’s design drawings annotated with numbers for space calculation. The paragraphs below show his design reasoning process.

The designer first drew an outline of the site (25 x70) and then reasoned about the depth of the work space. He wrote down "800," then 4x8 while looking at the number 25 on the side of the site. After writing down 4x8, designer put down 32' along the side and drew a rectangle of that size. The designer murmured and calculated that 70-32=38 and deducted 1 for partitions, arriving at 37.

He then drew dimensional marks with 10' intervals along the length of the site (three 10's and a 7'). Then he checked to see if the table would fit into the conference room. First he wrote down the dimensions of the table (4x10), calculated and wrote down the answer 40, and doubled it (80) for buffer space. He drew a rectangle to represent the table. He hatched the table see how the space would work.

He then wrote down 200 to be the dimension for the conference room. He drew a rectangle that occupied two 10' sections and half of the width, and wrote down the dimension for the conference room (20x10) inside the newly made rectangle. He overtraced the rectangle to see if it would be large enough. Then he left some corridor space and drew a smaller rectangle on the left side, wrote down 10x15 and said "this should be big enough for another office space."

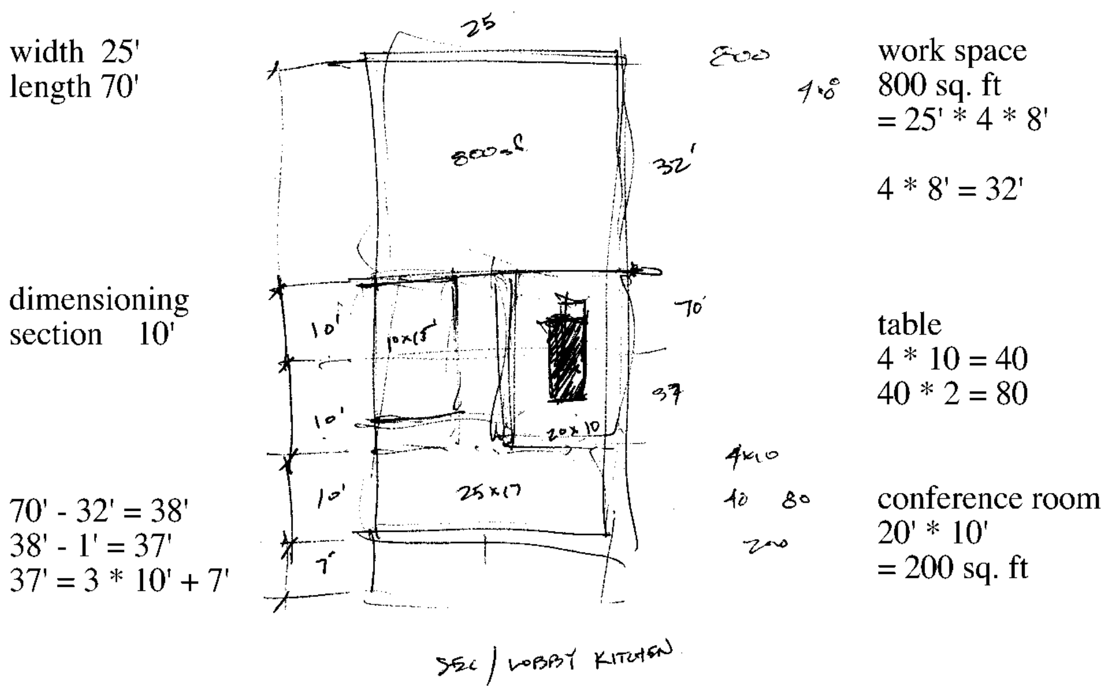
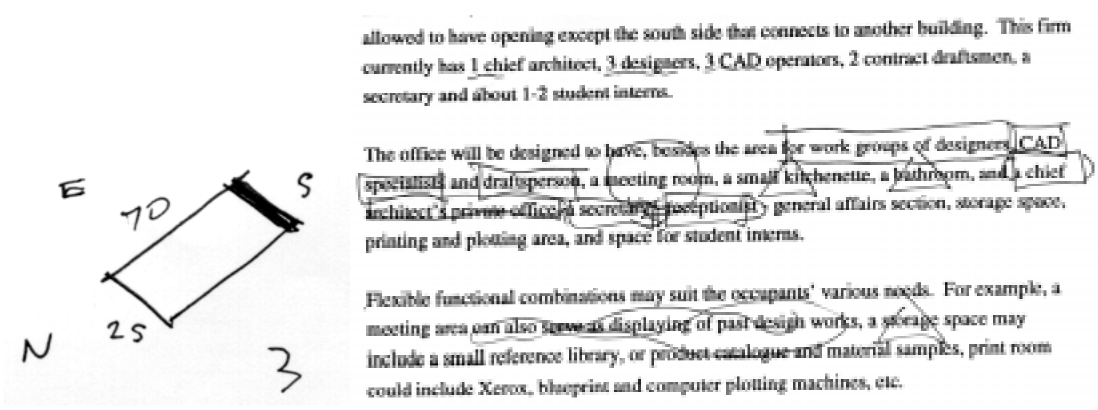


Figure 5-6. Process of calculating of space requirements and dimensioning: 1) mark the width and length of the site, 2) calculate to make room of 800 sq. ft. for work space, 3) calculate the left over space, divided by 10' intervals, 4) check dimension of table (4x10) to fit inside the conference room, calculate the space.

5.3.2. Noi, the 3D Sketcher

Like Roger, the first thing that Noi did after reading the first sentence of the program was draw a small rectangle to represent site, with dimensions [70, 25] and directions [N, S, W, E] labeled (Figure 5-7a). (However, Noi's direction North is down to the bottom left of the page, unlike all the other designers' convention of facing up to the top of the paper.) He hatched the south wall to indicate the wall was shared with adjacent building and could not have window openings. As he read the program, he underlined personnel information [1 chief architect, 3 designers, 3 CAD operators] just like Roger did. After reading the second and third paragraphs of the space requirement part of the design program, Noi started to use different shapes to frame space names – rectangles for designers, draftsperson and office, triangles for kitchenette, bathroom and storage, round shapes for meeting room, and round-cornered rectangles for secretary and receptionist (Figure 5-7b).



allowed to have opening except the south side that connects to another building. This firm currently has 1 chief architect, 3 designers, 3 CAD operators, 2 contract draftsmen, a secretary and about 1-2 student interns.

The office will be designed to have, besides the area for work groups of designers, CAD specialists and draftsperson, a meeting room, a small kitchenette, a bathroom, and a chief architect's private office, a secretary/receptionist, general affairs section, storage space, printing and plotting area, and space for student interns.

Flexible functional combinations may suit the occupants' various needs. For example, a meeting area can also serve as displaying of past design works, a storage space may include a small reference library, or product catalogue and material samples, print room could include Xerox, blueprint and computer plotting machines, etc.

Figure 5-7. 3D sketcher Noi's understanding of the design program: (a) thumb nail site plan illustrating dimensions, directions, and adjacent to existing building on the south wall. (b) underlines of personnel information, and use of different shapes to frame space names - rectangles for designers, draftsperson and office, triangles for kitchenette, bathroom and storage, round shapes for meeting room, and rounded cornered rectangles for secretary and receptionist.

Noi then used the blank space of the paper (bottom part) to sketch a long rectangle to represent the site. He then overtraced the 4 boundary lines, added vertical and horizontal lines to arrange some rooms on the top, drew a dashed line to cut through the plan to separate the top corner area (it was later labeled as “service”) from the rest of the plan. He continued to add more lines inside the rectangle, drew several small lines that were perpendicular to the left side of the site (east side) and then circled the bottom part of the site. Inside the circle he wrote down “WELCOME” and “DISPLAY” to label that area (Figure 5-8a). He then drew on the right another site plan (Figure 5-8b), double hatched the south wall (top), drew some partitions to divide the space unequally (smaller on the left, and larger on the right), unlike the previous one (Figure 5-8a). He drew window mullions on the left side of the walls, an added several vertical and horizontal lines to divide more space.

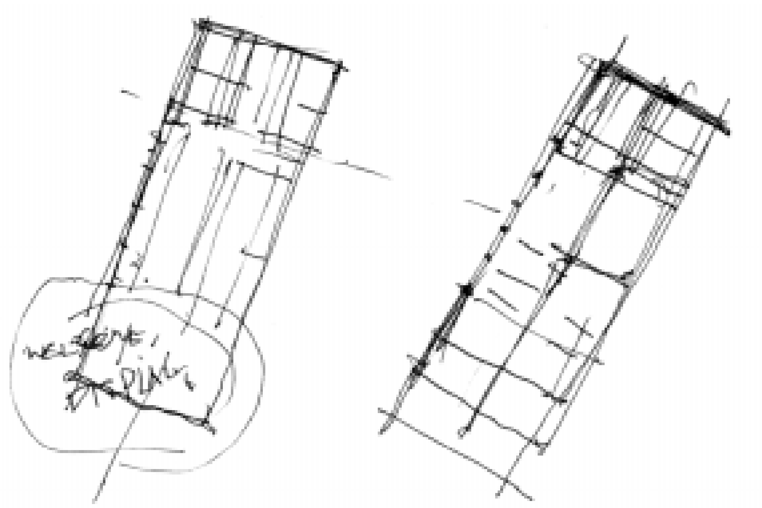


Figure 5-8. Noi’s spatial layout illustrated by using partitioning lines: (a) plan with walls, rooms, and corridor space. (b) plan with space partitioning lines focuses on openings, hatched line for solid wall on the South (top), dotted window mullions on the East (left).

At this time Noi said “the place is a warehouse that has a high ceiling.” Therefore, he reached for another piece of paper, explaining that he should draw a “3D” isometric view to see how the place feels. He quickly copied the floor plan again with all the partition lines on the new piece of paper. He explained as he drew: “a reception desk” [a rectangle] and “a desk” [another rectangle], and "this should be a hallway " [parallel lines, running top right to bottom left] and “by the hallway are the office tables” [short straight lines] (Figure 5-9a before extrusion into 3D drawing).

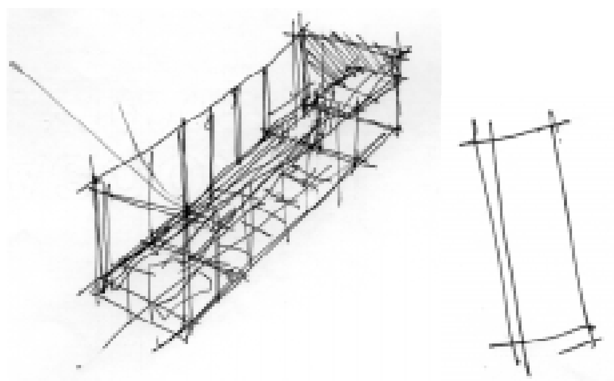


Figure 5-9. Noi's drawing of an isometric perspective that was derived from extruding floor plan. (a) the perspective view showing light ray coming in from the window. (b) a floor plan drawn to explain windows on the left side.

Then he began to make a “3D” drawing. He extruded the boundary up to draw an isometric perspective. On the left side he extruded two vertical lines from each of the window mullion dots on the plan. He then drew a small rectangle below this isometric to represent the site, double lined the left side (east side) and explained to the observer that east side should be good for windows that light could filter into the office in the morning. He went back to the isometric perspective drawing and drew two slanted lines (from the left toward the bottom right) that penetrated the window. He hatched[5] the back side of the

[5] Noi was a left-handed drawer, so his hatching pattern was running from top left to bottom right.

office (south side) and explained that the area would be for the kitchen, bathroom and storage purposes, and for the service area (Figure 5-9). This whole session was done in 6 minutes.

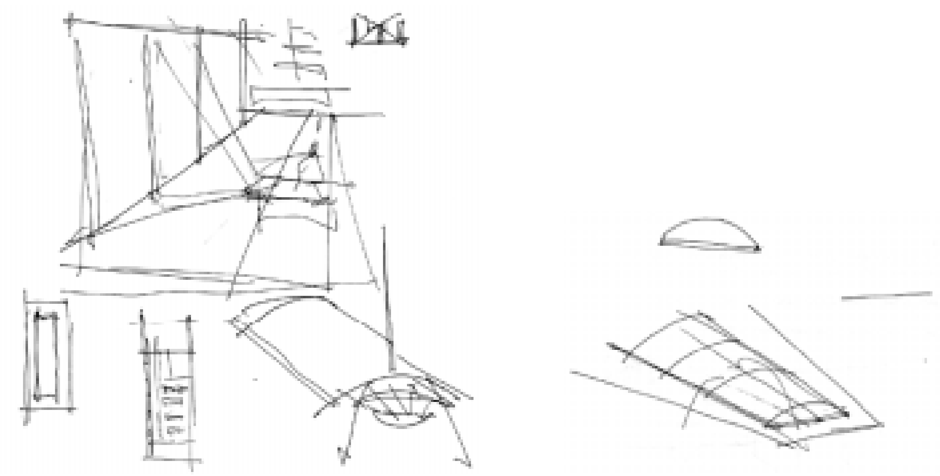
The next task (2) called for the zoning and spatial arrangement of the office. Noi first drew a plan (Figure 5-10a) with a basic site boundary and then hatched the top space, and wrote down “SERVICE” and “RES(troom)” while murmuring “service, kitchen, bath, restroom.” Then, on the small square he said and wrote “WORK(ing)” and then “OFF(ice)”. He drew a table underneath the work area, and dotted surrounding chairs. He then drew a line from the table and said “conference room” to explain to the observer. He wrote “RECEPTION” on the bottom space of the plan (Figure 5-10a).

Figure 5-10. Noi’s series of layout floor plans illustrated different focus (in drawing sequence a–e): (a) overall layout, (b)defining architect's office, (c) further consideration of path, hallway and partition, (d) window and kitchen counter, (e) reception area

Then, on the left of the plan (Figure 5-10a) Noi drew a similar plan (Figure 5-10b) to explore a partition for the architect's office with a window on the left side. He explained that normally he would trace over the previous plan (Figure 5-10a) on a tracing paper instead of drawing another floor plan on the side. On this plan (Figure 5-10b) partitioning lines divided a rectangular space on top. He made the left half of the bottom space into a divided linear space. He double lined the left wall to be a window, and added a small square to the right half of the space.

Then he drew another plan (Figure 5-10c) below plan (Figure 5-10a) to explore making office cubicles and CAD stations when murmuring “cubicles” and “CAD.” He pointed to the long lines on the left of the plan and explained it as a hallway. He then circled spirals down by the line and said "maybe put some partitions here". He doubled lined the top square space and said this should be “enclosed” for the “boss” and then drew a rectangle to represent a meeting table below the square. He then went back to the right of plan (Figure 5-10a) to draw another plan (Figure 5-10d) to explore putting window mullions (columns, small round circle dots), and design the kitchen counters. Then he drew another plan (Figure 5-10e) below plan (Figure 5-10d) to see if reception area could be further defined. This task (spatial arrangement, Figure 5-10) was completed in 5 minutes.

Noi performed the lighting task in three minutes. He first drew a person with a table in a sectional view (Figure 5-11a top), then drew two lines along the table to make vanishing lines in a perspective and two more lines to make a trapezoidal floor plan. He dotted window mullions on the vanishing line of the trapezoid first, then extruded them vertically up into “3D.” He drew several lines from the extruded windows onto the table and explained: “light” came in here from the window, and then drew a horizontal line saying there could be “shadow” as well.



(a) (b)

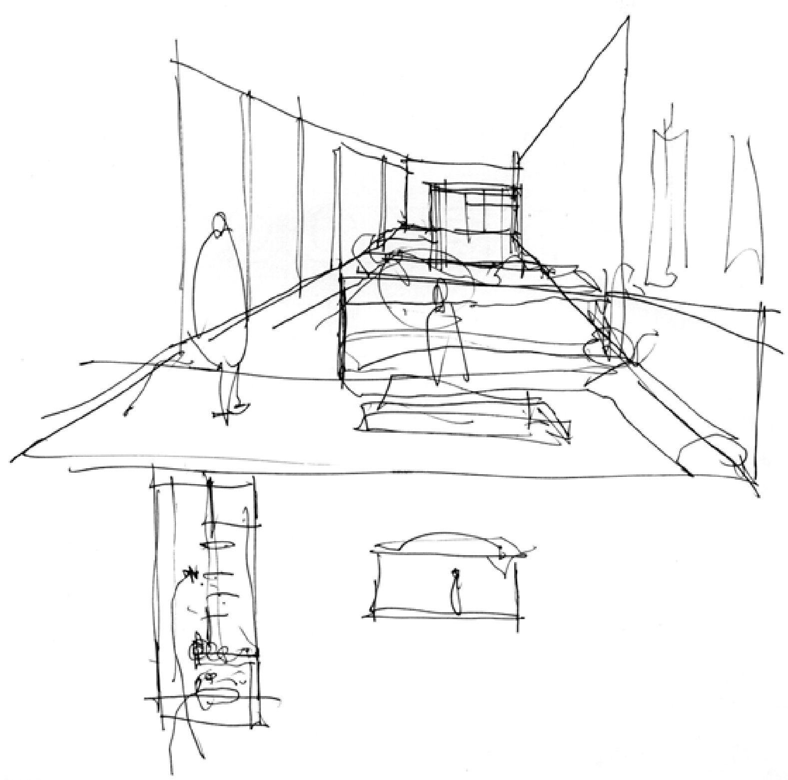
Figure 5-11. Noi’s drawings for lighting concerns: (a) top (left to right): sectional perspective, thumbnail section, bottom: ceiling plans, lighting fixture. (b) skylight perspective drawn from a dome section

He drew a small section (Figure 5-11b) to see how a person might appear in the space. Then he drew a plan (Figure 5-11b, second to the left) saying one could put some “hanging lamps” on the ceiling while drawing several parallel rectangles on the plan. He then drew a section of the lamp, a vertical hanging cord, connected to a down facing curve and a up facing bowl shape. He drew lines and arrows to explain this lighting fixture would have the curved layer to cast or reflect light. He drew another ceiling plan (Figure 5-11a bottom left) and said maybe we can do something on the ceiling too. He drew a rectangle inside the ceiling plan to illustrate a roof skylight window. Then he turned to the back of the paper. He drew a horizontal line and said, “ceiling” with “open sky(light)” and then drew a semi-circle on two horizontal lines to make a roof section. He drew the same shape again, and extruded it to make perspective view of the skylight roof (Figure 5-11b).

He explained that he was making a perspective drawing from a section while drawing several more arch curves, and drew two lines from the skylight downward, saying that the light can come in from the top.

Task 4 asked for attention to visibility and privacy concerns. Noi said he had already done it in the previous tasks. So he decided to quickly sketch in a perspective view and explained the space he designed: “on the left side of the office there are windows open to outside, and the square in the back would be for the boss’s office, then some cubicles for designers in front would be the working area.” Noi continued: “A person can sit here, this is reception area, let's put a couch for people to sit and wait here.” He drew a tree on a circular vase, a person sitting, and then a person standing. He said, “Since people would be waiting here, maybe there should be some drawing on the wall (he drew 2 rectangular shapes), and a table to put some building model.” (Figure 5-12a)

He drew the floor plan again (a straight rectangle, Figure 5-12b) on the bottom of the perspective and explained that the top corner square would be for the boss (chief architect), in front of it would be tables for the working area. He went on to drew some horizontal circular spirals and suggested putting some “plants” and “trees” here behind the receptions as a buffer to block the direct visual access to the working area behind. He then drew two arrowhead lines to indicate how people would come in to the reception area, and then to the work space. Finally he drew a section with a dome roof ceiling, double-lined ground and a person standing in the space (Figure 5-12c). The whole task (Task 4, Figure 5-12) took only two minutes.



a

b c

Figure 5-12. Noi's drawings for task 4: (a) Sectional perspective showing windows, chief architect's office in the back, reception area with couches, table, painting on walls and people. (b) plan, (c) section.

Noi quickly did the last task (5) in two minutes as well. He drew a rectangle and a square, extended the rectangle corners to four projecting lines to make a perspective view. He overtraced the square saying that it would be the boss's office, and drew several horizontal lines explaining that they were blinds on the window that can be opened to see the employers, or shut to ensure privacy. In front of the blinds he drew a rectangular shape, saying this would be the meeting table. He then hatched underneath the table to add depth, and then added legs, two circles as human heads to indicate people sitting, then a standing person, and then a rectangle to frame the person and illustrate hallway space. He drew several horizontal lines and L shapes (rotated 90 degrees clockwise) and said that these would be the working areas, and drew some lines to represent tables (Task 5, Figure 5-13 bottom). Then he concluded that he was done with the design.

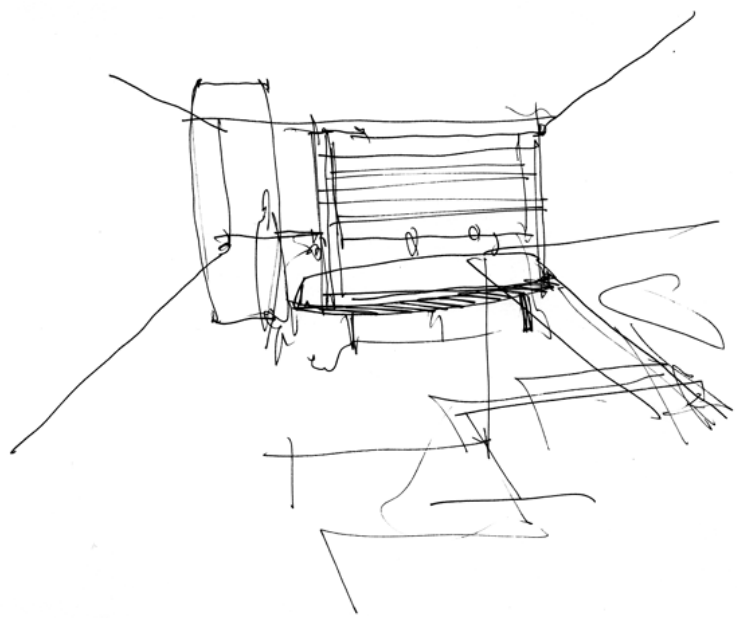


Figure 5-13. Noi’s sectional perspective drawing show a conference table in front of the chief architect's office

5.3.3. Samuel, the Philosopher

Like Roger and Noi, Samuel started by trying to understand the site and the program by underlining the space names in the program text. He then reasoned that the width and length was in a 1 to 3 proportion (Figure 5-14) so he drew a rectangle with the length approximately three times the size of the width. He reasoned that sun would come in from the east side in the morning, and he drew an arrow to indicate the sun light coming from bottom right (Southeast). He drew the site (6:22, and 7:51) with partitioning lines (like Noi) to be sure he had a sense of how big the place was and of the site constraints.

Then he drew bubble diagrams, like Roger, to partition the space to match the architectural program. Besides the diagrams, he wrote down the program on the top corner (09:24, "place for designers, CAD and draftsman"). He then explained his overall layout plan (10:20) by making circles and arrows for the main and side entrances, labeling them "public" and service spaces. He further divided public space to create a common waiting area with a "screen buffer" to the back (overtraced vertical lines). He divided private area to have a chief architect's office, and decided to move the side entrance from east side to the north side.

He then decided there could be a level change [overtraced vertical lines on the right] to the chief architect's office and that the north side wall should have lots of window openings. After the basic plan of layout was settled (12:00), Samuel continued to test the space by adding furniture like tables and partition walls, and more space bubbles (Figure 5-14, Task 1).

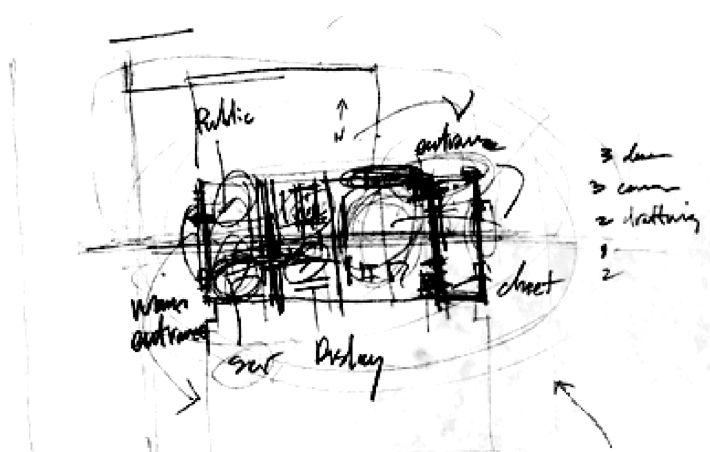


Figure 5-14. Samuel's drawing showing site, program notes (right), direction north (arrow N), sun light (arrow from bottom, south east), zoning (public, service, display, entrance), and level change to architect's office (bottom right rectangle with steps beside it).

Samuel explained (13:57) that he was drawing bubble diagrams (Figure 5-15) to get the look and feel of the layout. He circled different spaces to explain where the kitchen, work and public area would be. He drew a table [small rectangle] to split the circulation, and commented that there could be a guided tour to the gallery and office. He questioned whether the restroom should be in the middle of the room for visitors, or just stay in the back with the service area and a side entrance.

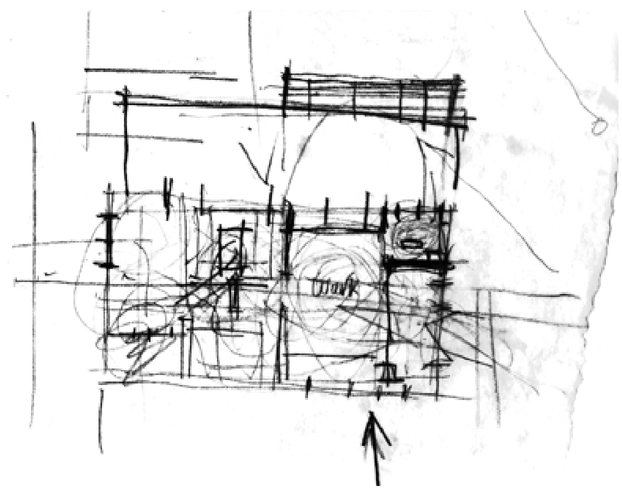


Figure 5-15. Samuel's bubble diagram for a spatial layout arrangement for Task 1.

He drew several arrows out from the architect's office, explaining how the architect can see what's going on in the firm. He then went on to explain how lighting can come in from a clerestory, maybe with high skylight windows (lines with division marks, below floor plan, bottom of Figure 5-15) on the south side (where adjacent to existing building and no windows allowed on the wall) to accept sunlight.

Samuel went on to the second task and found it asked for lighting concerns (18:33). He explained he already thought about it and would use this task to make some more detail design. He added dashed line for ceiling then drew a section of the building (Figure 5-16 bottom) with arrows to illustrate how the light would come in from the skylight window (Figure 5-16). He explained that the windows could have louvers on the curved roof to reflect and control the lighting angle, and could be augmented by artificial light.

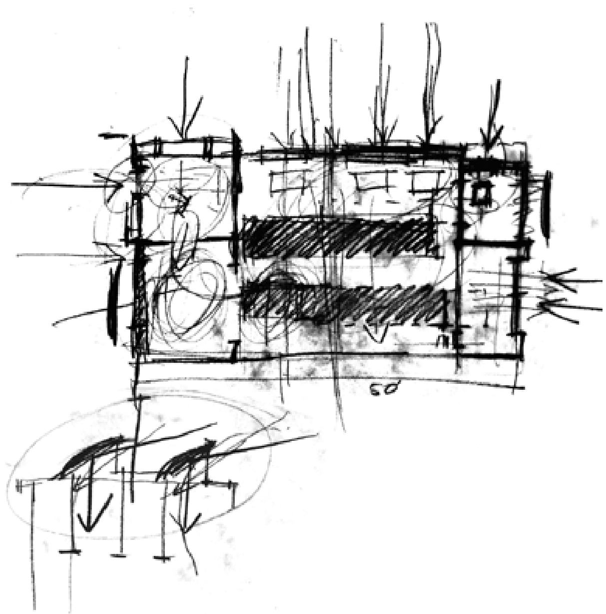
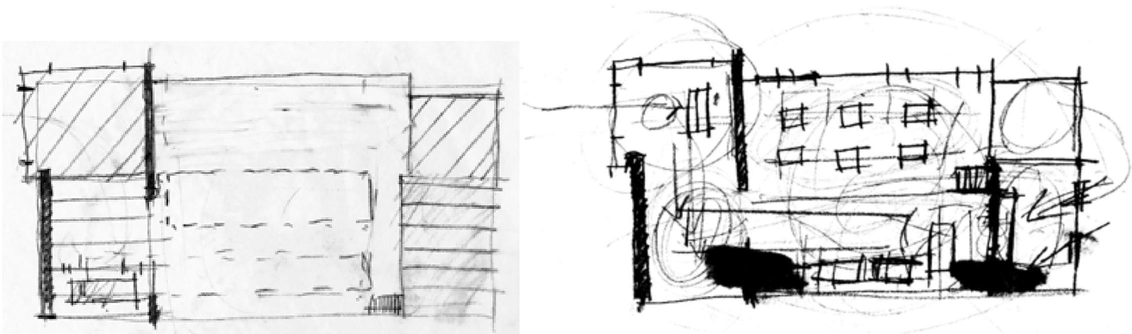


Figure 5-16. Samuel's floor plan showing north lighting (arrows on top), tables of work space (rectangles), lighting ceiling plan (hatched long rectangles) and a sectional drawing illustrating how lights come in and be reflected and controlled by the skylight windows and roof.

Task 3 asked the designers to focus on visibility. Samuel explained as he drew the floor plan again (35:07) the arrangement of solid walls, closed partitioning and clear screens. He hatched the kitchen and architect's office, added stairs (37:32) to think about how different areas would interact. He then drew dashed lines to illustrate a circulation path (Figure 5-17a). He drew arrows to indicate visitors' path. He folded the tracing paper to overlay the plan and then drew on top of it (Figure 5-17b). He erased the wall between the reception area and the office (erased mark, bottom left of Figure 5-17b). He added more partitions and furniture for the work area and the architect's office (bottom right, Figure 5-

17b), reasoning that a table would be needed to put drawings and models under the raised architect’s office (Figure 5-17b). He erased and moved the stairs (bottom left of architect’s office) up and left a black mark on the trace.



(a) (b)

Figure 5-17. (a) Plan of different space arrangement. (b) trace on top of the plan and added furniture (table and counter), steps to architect's office, lighting and circulation.

The final task asked for checking dimensions. Samuel first circled different spaces as he explained what he had done in the spatial layout (Figure 5-18). He added detail as he moved on: “let’s use wood floor for the reception gallery area”... it (receptionist and gallery area) can showcase the whole firm, allow nice controlled light to come in, and blocked visibility to the work space.”

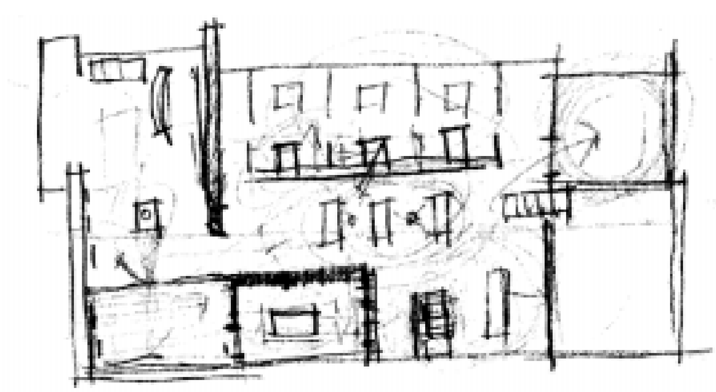


Figure 5-18. Samuel’s testing on the spatial arrangement - detail furniture and room partitioning.

Then he drew the meeting table (48:34) to indicate the conference room could be located next to the gallery area, while adding more solid walls to block noise and visibility. He went on to add to and explain his plan while totally ignoring checking the dimensions as requested. (However, he had considered the scale and dimensions carefully in the beginning of the session when he performed Task 1). Samuel was very articulate and explained as he designed. The total session took about fifty-three minutes.

5.3.4. Mario, the Research Architect

Mario started by trying to understand the site (converting the dimensions from feet to meters, Figure 5-19a), and the program, drawing the site several times to be sure he had a sense of its size (Figure 5-19b, Appendix F, and Figure 5-20). Then he partitioned the space, matching the smaller spaces to the architectural program (11:40; "we have one architect, two three...space for receptionist..."; Figure 5-19c) Unlike Roger, Noi, or Samuel, Mario did not underline the program requirements; instead he copied them down on the paper (Figure 5-19b) he was working on (Samuel also wrote down parts of the program on the tracing paper, see Figure 5-14, and Appendix E).

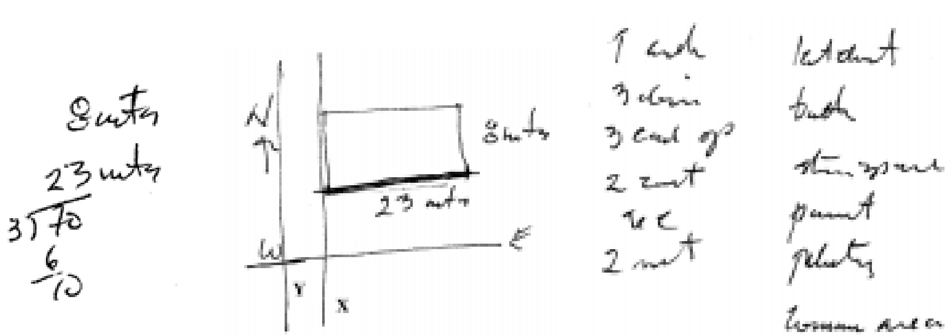


Figure 5-19. Research architect Mario’s drawings: (a) Conversion of 25 ft and 70 ft into meters (8 and 23). (b) site plan interpretation of the program, (c) written notes of personnel and space information of the design program on the tracing paper.

Mario then checked the sizes of spaces he made, using a scale to measure the dimensions, and drawing in tables, chairs, workstations to be sure that the space was adequate for the activities that were to take place inside. He went back and forth making spaces by adding partition walls and testing them by drawing furniture, etc.

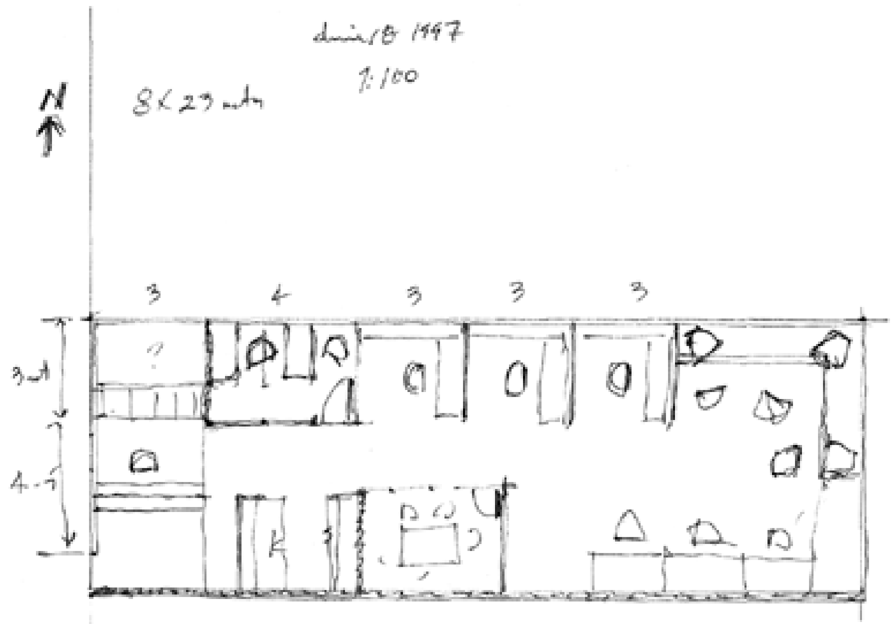


Figure 5-20. Mario's floor plan for Task 1 – he used detail furniture arrangement to test how the space would work.

At 17:42 Mario took a new trace layer (Figure 5-21) and began to draw the window mullions in the wall and structural columns. (He didn't need to do this, as he was not asked to solve any structural problems). He said "columns" (18:36) after drawing those repetitive small rectangles. These small rectangles can be identified as columns because (1) the little rectangle-squiggles were drawn on top of the boundary lines (which were supposed to be walls and windows) and (2) he drew them across the whole drawing, in a kind of grid or evenly spaced row, rather than inside a particular space (as he would if he were drawing a piece of furniture such as a chair or a workstation). He then came back to space partitioning (18:55) and drew a new wall, and then did space testing again by drawing furniture. He went back to the structure – using the structure (20:35) to set up some space definitions - and back to furniture again.

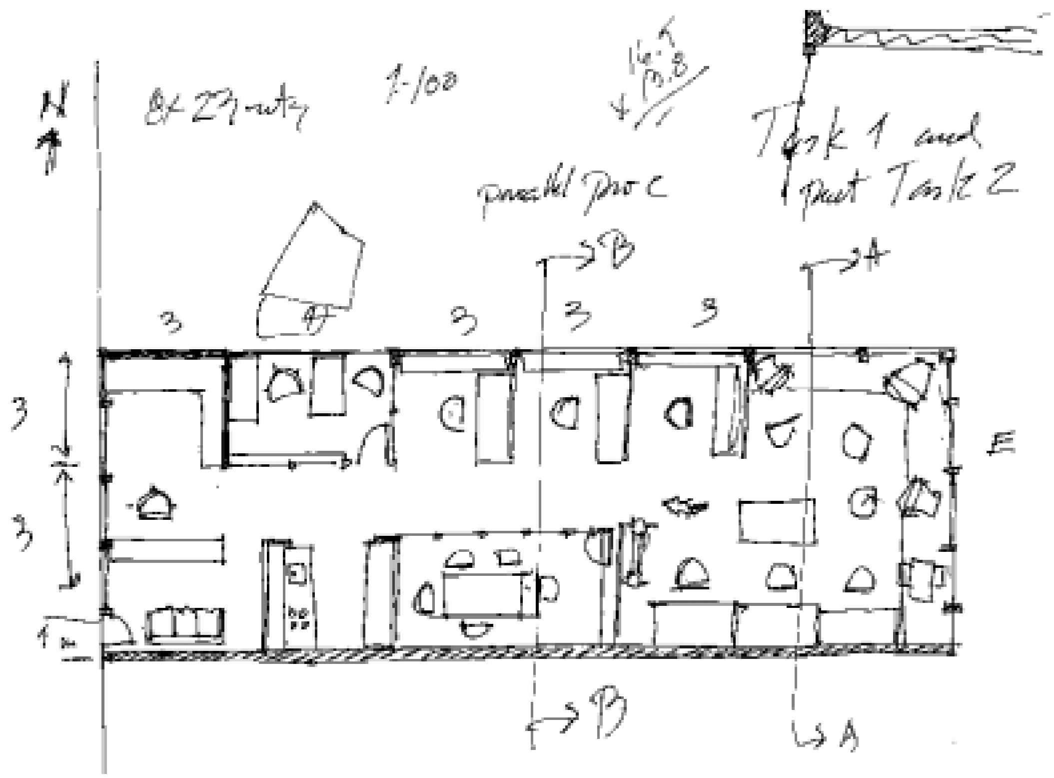


Figure 5-21. Mario's second floor plan has columns traced on top of last layer (Figure 5- 20, Task 1)

For Task 2, lighting (26:00), Mario looked at the problem statement but decided that he had already solved the problem. He was expecting Task 2 to be a different design problem with a different program. The observer asked Mario to explain about lighting, so Mario waved his hand over the design drawing (Figure 4-22a), talked about north, the window, etc. Finally he decided to work on lighting noting that some of the people would get "fried" (28:00). He started by cutting a section (A-A) across the plan (Figure 5-21 right) where the CAD workstations were. On the new trace (Figure 5-22b) he decided the height of a floor (3.5 m), and placed a row of windows. He then drew diagonal lines to indicate light rays from the windows onto the work area. It seemed that while illustrating lighting on a sectional view, he was still concerned and was checking that the dimensions of the spaces work for their intended use by drawing furniture, people and light. In other words, space planning was still going on in section even though he was asked to focus on lighting issues. He added (45:00) another sectional drawing (section B-B, cutting through the meeting room and designer cubicles, as referred to in the plan of Figure 5-22) showing people, table, chairs and hanging lamps (Figure 5-22c).

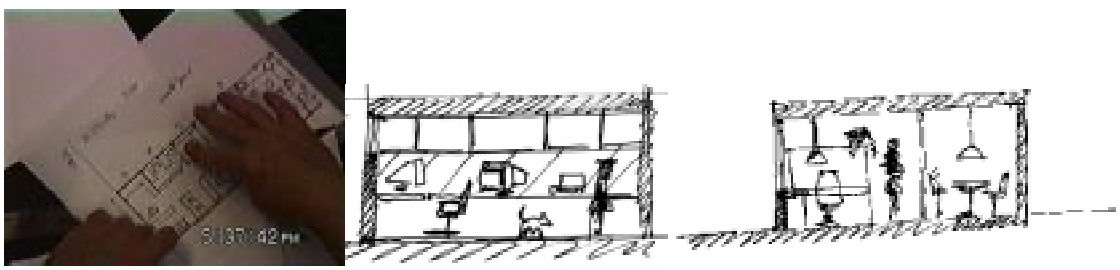


Figure 5-22 Mario’s responses to Task 2, lighting concerns: (a) Used gesture to explain about his design. (b) Section A-A depicting computer work stations. (C) Section B-B depicting a work space cubicle, the corridor space, and the meeting room.

Mario continued (45:00) to read the next task (3) on visibility and privacy. Once again, he seemed to think he already solved it and so instead he used the drawing to explain how his design deals with visibility and privacy. He was not doing any designing or drawing any new marks, but, in explaining the privacy and visibility issues he reviewed his idea and suddenly (46:28) began to rework his plan. At first it seemed that he was just redoing the space planning design (enlarging the chief architect's office, testing with furniture) but later (49:38) it became clear his rational for rearranging the spaces was to improve the visibility and privacy in the design (avoid visitors seeing the kitchen, etc.)

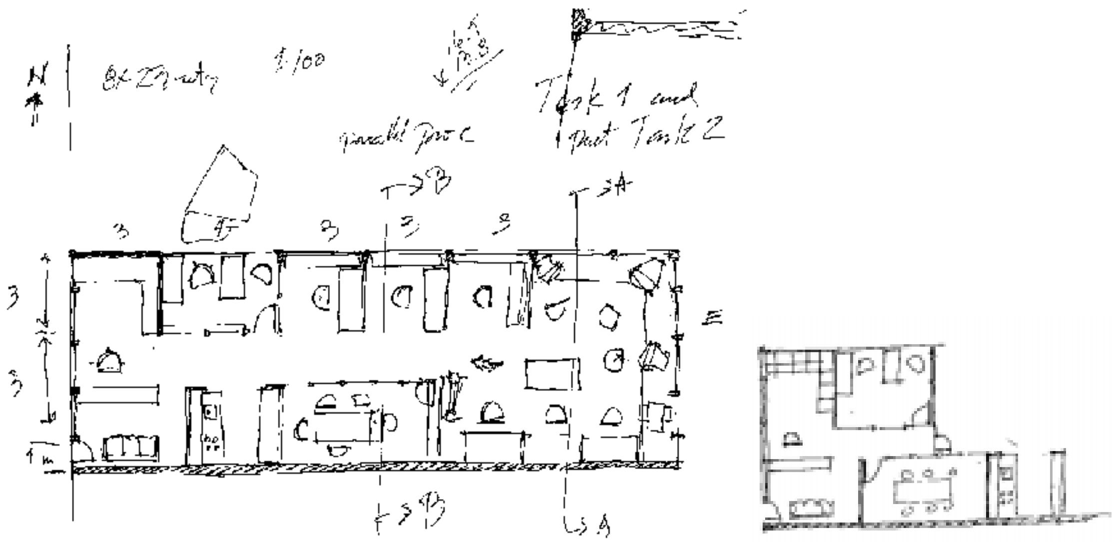
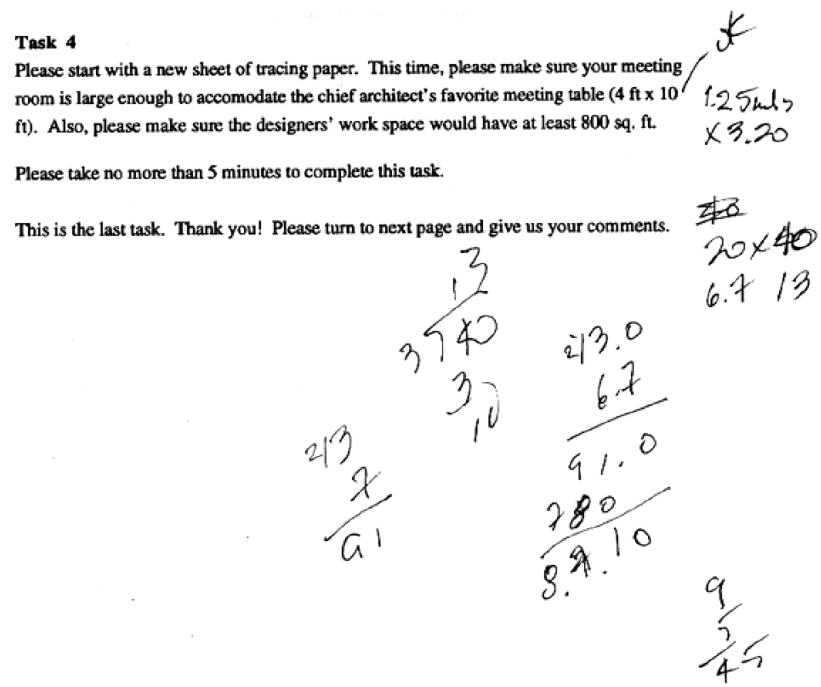


Figure 5-23. Mario's drawing for Task 3, visibility issues: (a) Original floor plan layout designed in the task 1 session. (b) Modification of switching the location of conference room and kitchenette (to avoid seeing dirty dishes) in the task 3 session.

Mario did not draw anything different when he was thinking about visibility and privacy concerns. He relied on his drawings from the previous tasks and used hand gestures to explain the movement and circulation among spaces. It seemed that in the beginning of this task session, his main concern was still the space-arrangement, but he tested his designs against visibility and privacy issues. When explaining and reviewing the plan to the observer, he saw a space arrangement that would improve visibility and privacy performance (hiding the kitchen in the back), and returned to space arrangement to implement it (by switching the location of the conference room and the kitchenette, Figure 5-23b).

Finally (51:00), Mario was asked to do some dimension checking (Task 4, minimum square footage requirement and fitting a large meeting table into conference room). He started by converting the dimensions in the problem statement from feet to meters as he did in the beginning of the design session. First he did the calculation in his head and wrote down numbers (4 ft x 10 ft became 1.25 meter x 3.20, and 800 sq. ft. became 20 x 40), and then made further calculations on paper, (Figure 5-24 shows whole calculation record of dividing 40 by 3, and multiplying converted dimensions (13 and 7, and 13 and 6.7)) to check out the square footage. He also used the scale to check the dimensions of the plan.


Task 4

Please start with a new sheet of tracing paper. This time, please make sure your meeting room is large enough to accomodate the chief architect's favorite meeting table (4 ft x 10 ft). Also, please make sure the designers' work space would have at least 800 sq. ft.

Please take no more than 5 minutes to complete this task.

This is the last task. Thank you! Please turn to next page and give us your comments.

Figure 5-24. Calculation and conversion about dimensions.

Mario spent about an hour to complete the whole design session and like Samuel, did not follow all the task statement requests.

5.4. DETAILED ANALYSIS OF THE DESIGN SESSIONS

5.4.1. Functional Considerations of Roger's Design Session – Diagram Conventions and Dimensional Reasoning

Results from Roger's design session came closest to supporting the original hypothesis. He followed the instructions of the tasks, strictly observed the time limits and explained, both verbally and in writing, his thinking, and used a set of well-defined drawing elements (or symbols) to indicate different concerns. He drew bubble diagrams when thinking about conceptual, schematic design [Task 1]. When dealing with spatial arrangement [Task 2], he drew furniture such as tables and chairs in the room to test how the space would work. He drew a sun symbol with light ray penetrating the windows into the building when working on the lighting [Task 3]. He drew lines and double arrow links to indicate relations or movement for visibility and privacy issues [Task 4]. He drew windows, walls and doors to define space in design. He underlined and framed space names with boxes when reading the program. He used dimensional symbols (width, ceiling height) and wrote down numbers when reasoning about dimensions [Task 5] (see Section 5.3.1 for Roger's dimensional reasoning design process). Finally he drew and made a small site plan with dimensions [25', 70'], direction North [arrow and N], and a street (double arrowhead line) and an adjacent building [a shaped arrow], and entrance [single line arrow] information after reading the program (Figure 5-25).

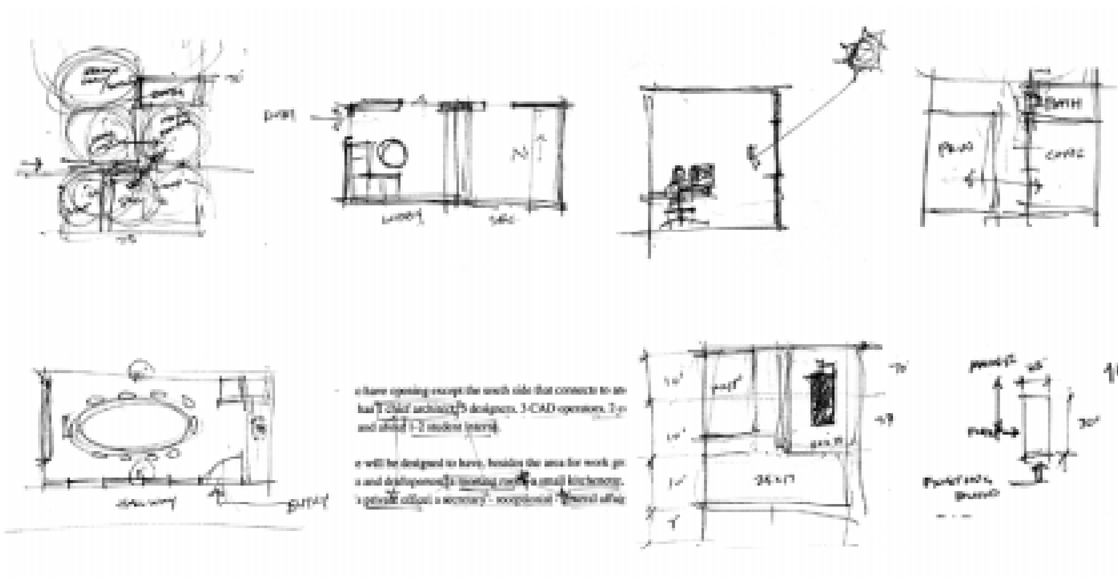


Figure 5-25. Function designer Roger's convention. Top:(1) bubble diagrams, (2) furniture, (3) lighting section, (4) double arrow line for relations. Bottom: (5) building elements: wall, window and door, (6) underlining and framing program, (7) dimensional reasoning, (8) shorthand of the site plan as understanding and reminding of the program

5.4.2. Noi's 3D Sketches: Lines for Partitioning, Lighting and Hatching

Noi also pretty much followed the time limits for performing each task. He explained his design process with occasional text labels, and he used a set of drawing conventions (Figure 5-26). However, Noi had a tendency to make perspective, isometric "3D" drawings out of his plan or sectional drawings. He explained it was very easy to make perspectives, "you basically just extend and extrude lines from the plan or sectional views." Toward the end of the design session, he simply ignored the task and went on to draw "3D" views. Despite his personal preference for making "3D" drawings, several conventions could still be identified. He made a small site plan with dimensions [70, 25] and directions [N, E, W, S], and hatched the south wall. He underlined personnel information, and used different shapes to frame different types of spaces in the program. instead of drawing bubble diagrams, Noi drew partition lines for spatial layout [Task 1]. Noi drew lines penetrating the building to illustrate lighting [Tasks 1 and 3] and changed the direction of arrows to illustrate reflective lights [Task 3, lamp section]. He used hatching and text labels to indicate space [Task 2] and drew simple furniture [tables] and human figures (Tasks 2, 3, 4, 5). He used double lines and small circle dots to indicate window openings, mullions and partitions [Task 1, 2, 3]. He used plan, section and “3D” interchangeably to support and illustrate his design [Tasks 1, 2, 4].

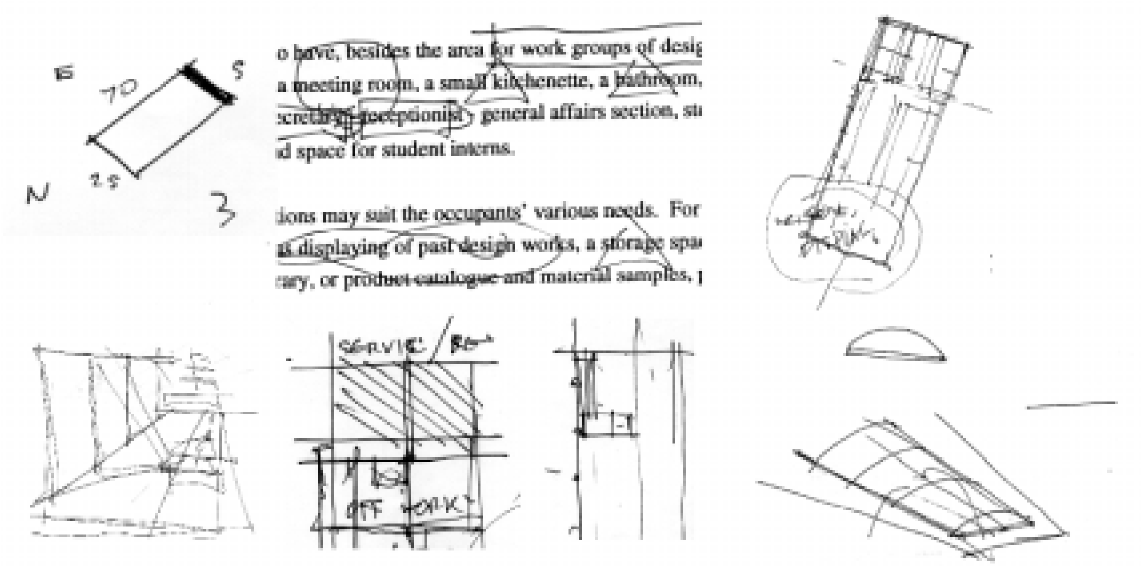


Figure 5-26. Noi's design conventions: (1) shorthand of the site plan, (2) underline and different shape framing program requirement descriptions, (3) spatial partitioning for layout, (4) lighting penetrate building envelope, (5) label and hatching of space, (6) dots and lines as windows and walls, (7) making 3D from plan or sections.

5.4.3. Philosopher Samuel's Design Process: Arrows for Lighting, Entrance and Visual Access

Like Noi, Samuel also ignored Task 4, which called for a square footage requirement for work space[6] and for fitting a large meeting table into the conference room. Samuel also ignored the time constraints and spent about fifteen to eighteen minutes for each task. He drew, talked and explained as he design. He used many hand gestures to explain his design reasoning, and he used a set of conventions in design drawings though he thought he used "very few" of them. Samuel chose to use yellow trace with pencil.

Samuel's design drawing for Task 1 seemed to be a hybrid of Roger's bubble diagram and Noi's spatial partitioning. He drew bubbles to represent different zones - public display areas and private work areas. He also drew many lines to define space -- "chief" (architect's office), "meeting" room, and "kitchen"– and called these lines walls, windows, or screens. At the scale of a space, he used slanted line hatching to indicate the area of the space, parallel horizontal lines to indicate wood flooring. At the furniture scale, he drew rectangles to represent tables, circles for chairs, a curved shape for the receptionist counter, a rectangle with wide divisions as a couch, a rectangle with narrow even divisions as stairs, and a circle inside a square to represent a model stand. He used arrow lines to illustrate North, sunlight, entrance, movement and views.

For Task 2, lighting issues, Samuel first drew dashed lines (19:34, "let me dot in where the light would come in from the top") on the floor plan to illustrate overhead ceiling

[6] Though at 19:15 as he stated for Task 2, Samuel mentioned "give half of the space to workers, Umm, about 50 ft."

plan, then he drew roof sections to illustrate how the light would come in from the lifted skylight windows. He went on to design and talked about lighting for each space, and added tables, walls and even an overhang roof for the side entrance (29:31). Later he talked about alternatives to direct sunlight, pointing out that the roof and window could provide more complex control of how the light can be reflected down to the work space.

Task 3 asked for attention on visibility and privacy. Samuel believed that he had already done it (35:14) and decided to use this time to "really define how they work". He drew the floor plan again with lots of explanation about how the wall and clean screens could help not only the definition of the space but also lighting and visibility control. He added a plotter and printer into the workspace and suggested a sound buffer. Seeing that some display was needed for printouts from the printer and plotter, he added a table. Realizing that after viewing the drawings, some would need to be stored, he then moved the stair to the architect's office from the bottom corner against the wall up to the center and assigned the space underneath the architect's office to be drawers for storing drawings. He mentioned the partition between the work area and architect's office might employ windows, even with mirrors on one side to ensure privacy and visual access. He drew many arrows to indicate visual access.

For the last task (Task 4), Samuel did not check the dimensions as requested from the problem statement, instead, he continued to talk about his design with more details, such as adding tables for "6 spaces for designers and draftsmen," tables for "design workshop," and possibly using a "wood floor" for reception and gallery area.

Briefly, Samuel's design session also confirmed the experiment hypothesis: 1) he underlined space requirement to understand the program, 2) he used circles, lines, and labels in a bubble diagram, 3) he drew sections with light rays to illustrate lighting, 4) he used arrows to indicate entrance, lighting and visual access, 5) he drew lines, hatching and shapes to represent windows, walls, and furniture to think about the design concept.

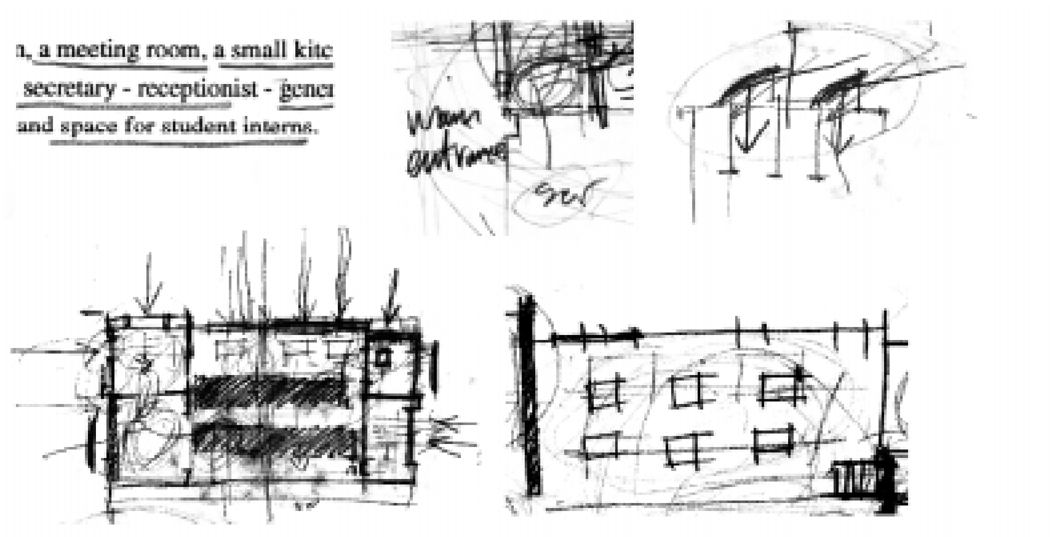


Figure 5-27. Designer Samuel's convention: (1) underlined program, (2) label with text and use arrow, lines and circle in bubble diagram, (3) roof section lighting, (4) arrows of lighting, entrance and visual access, with hatching, (5) architectural elements: stair, tables, windows and walls.

5.4.4. Research Architect Mario's Concerns – Scale, Dimensions and How Things Work

It seems that Mario's actions did not completely respond to the experiment design setup. The experiment failed in two ways. First, the experiment called for talk aloud protocols, but Mario said he preferred quietly doing design. Therefore occasionally the observer had to prompt Mario to verify his reading of the drawing. Secondly, the experiment design called for the participants to attend separately and distinctly to different tasks: space arrangement, lighting, etc. in a limited amount of time. The idea behind this was to ensure that one would know what the designers were thinking about and trying to accomplish, and therefore obtain more evidence of design intentions from the verbal transcript. Instead, Mario, an experienced designer, simply ignored the experiment instructions and spent half an hour designing for the first task and then used the time in later tasks (Tasks 2-4)to explain how his design addressed those issues while he continued checking and testing to see if his design worked.

Mario’s design session however was also successful: his design drawings, like others, revealed many drawing conventions. A match of the tasks with drawn figures of Mario’s drawings is similar to – though not exactly the same as – the other participating designers. Like Samuel, he used hand gestures to explain many design concepts, and did some thinking in his head without writing down or talking it out.

From reading the design transcript (see Appendix H) it seemed that Mario performed three things throughout the whole design session: (1) thinking about the site (understanding the dimensions and orientation by writing down numbers, using the scale measurement, and writing down N, E, W) (2) partitioning the space available to correspond to the program (arranging space by putting down partition lines for walls, windows and doors) (3) checking the partitioned space (by drawing in furniture such as tables and chairs, human figures, plants and dogs).

Mario’s furniture symbols were the most defined and easiest to recognize of the four designers. His design session supported the experiment hypothesis in that his drawing revealed several conventions : 1) he drew a small plan to represent design program, 2) he copied down the personnel and space requirements from the program on the tracing paper he was working on, 3) he drew similar kinds of objects in a sequence: table, table, table, then chair, chair, chair, and monitor, monitor, monitor, 4) symbols for door, walls and windows, 5) he drew cut sections to show lighting concerns, 6) he used furniture to test how the space works and to think about the particular design context, 7) he wrote down numbers to calculate square footage and to convert between different metrics, and 8) he used symbols to label dimension.

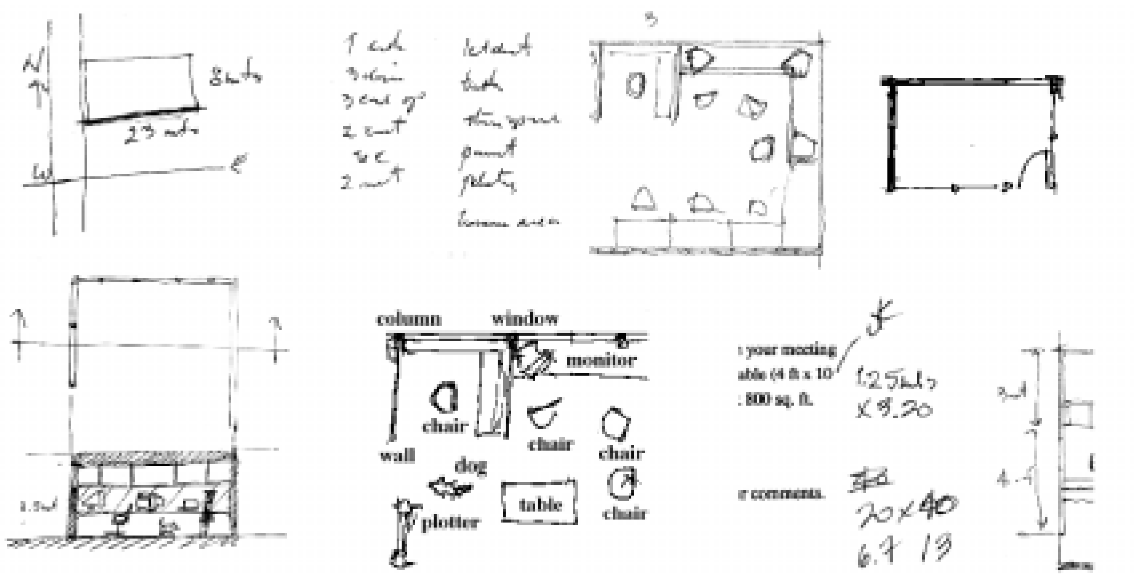


Figure 5-28. Mario's drawing conventions: (1) small site plan with direction north, and dimensions, hatched wall, (2) copy down design program, (3) drew same kinds of object in sequence, 3 monitors, 3 tables and 3 chairs, (4) door, wall and window, (5) cut section to show lighting and space, (6) use of furniture to test how the space works, (7) dimensional reasoning, conversion of number calculations, and (8) labeling the dimensions on side

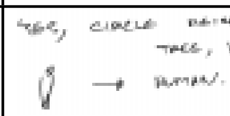
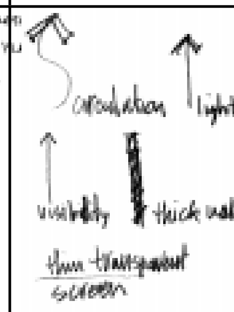
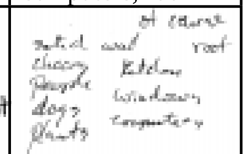
Mario’s design session reveals many drawing symbols: walls and partitions, doors, chairs, desks, computers, dogs, windows, light lines, numbers, north-arrows. Most of his drawing actions seemed to deal with space arrangement, and he continued to perform this task even when he was asked to deal with other issues. Therefore, this design session did not clearly show a direct mapping between drawing and design intentions as described in the tasks, though the drawings for Tasks 2-4 still bear resemblance with those of the other participating designers. On the other hand, it is interesting that Mario folded the first task – spatial arrangement – into all the rest of the tasks. As the other designers (Roger, Noi and Samuel) declared, they had thought about lighting and privacy issues and dimensions when they were performing the first task. Perhaps the participating designers were trained to think about and attack spatial arrangement concerns by weaving other issues in. It made it harder to try to take apart the design task into small subtasks.

5.5. POST-EXPERIMENT QUESTIONNAIRE

All participating designers filled out a post-experiment questionnaire after they performed the design experiments. Participating designers were asked to express their opinions about the design tasks: were they clearly defined and easy to perform? All four designers reported that the design session was fun and easy to perform. The designers acknowledged that they used personalized symbols and short hands when making design drawings. They used drawing conventions such as windows, walls, door, circulation, dimension, lighting and human figures. Table 5-2 shows a summary of what they reported.

Table 5-2. Response of the experiment from participating designers

Name	Roger	Noi	Samuel	Mario
How clearly were the tasks defined *(0 unclearly- 4 clearly)	task 1 4.0 task 2 4.0 task 3 4.0 task 4 4.0 task 5 4.0	task 1 2.5 task 2 3.0 task 3 2.0 task 4 3.0 task 5 1.5	task 1 3.9 task 2 4.0 task 3 2.5 task 4 3.3	task 1 4.0 task 2 4.0 task 3 4.0 task 4 4.0
How difficult to perform the tasks • (0 difficult - 4 easy)	task 1 1.0 task 2 2.0 task 3 1.0 task 4 1.5 task 5 4.0	task 1 3.5 task 2 2.5 task 3 1.5 task 4 1.5 task 5 1.5	task 1 3-4 task 2 3-4 task 3 2.5-3.5 task 4 2-3	task 1 4.0 task 2 4.0 task 3 4.0 task 4 4.0

How often do you use a computers for architecturally related work? # (0 never - 4 all the time)	3.0	4.0	3-4	4.0
Do you think you have your personalized symbols and short hands for designing?	yes	yes	if I do, they are very few	of course
What are they?	window, wall, door, entry, movement, link, height, dimension	tree, plant, human	circulation, light, visibility, thick walls, thin transparent screen	solid wall, chairs, people, dogs, plants, kitchen, windows, computers, roof
symbols and short hands				
Would you like a computer program to have freehand sketching abilities? What should it do?	Yes, I want to see what I draw in the pad I'm drawing on. Save old sketches for reference and come back to them and do overlays and sketches.	Yes, it should has(ve) all ability to sketch, but also has ability to undo or overlay, and it should act like a real pencil.	The ability to hide itself. The idea being when I sketch - I want to SKETCH, that's it, I don't want to fumble with a clumsy interface or be distracted from my sketch.	I don't know
Suggestions and comments for the experiment?	Think about changing program. An Arch.(itect's) office might be to(o) easy for arch.(itects) who have already worked in one and know what is good and bad.	It's very fun. Task can be more organized. Thank you.	You need to make it so that you forget you are using the program. (continue from last question)	None, ok. None - Place the camera in a more comfortable place.

* (0 difficult - 4 easy) • (0 difficult - 4 easy) # (0 never - 4 all the time)

The end of the questionnaires asked designers to openly comment about what kind of computer support for freehand drawing they would like to have and provide suggestions for future experimentation. The consensus was that designers like to draw freehand sketches and if a program were to support this ability, it would need to be as close as to the touch and feel of a real pencil and paper interface. Roger, the functional designer, suggested that a sketching program should have the ability to keep the old drawings for future reference, and allow designers to retrieve, overtrace and modify them.

5.6. EXPERIMENT RESULTS: DESIGN CONTEXT AND DRAWING CONVENTIONS

The experiment examined drawing symbols and configurations used in the design process and how they differ between different design tasks.

The previous study with sixty-two designers (Chapter 4, also (Do, 1995b)) reported that designers share and can understand other designers' drawing conventions in diagramming architectural concepts. The goal of this chapter was to verify whether these conventions also appear when designing. In the four design sessions, several drawing conventions that correspond to different design concerns were identified, for example, bubble diagrams and line partitioning for spatial arrangements, a sun symbol and light rays for Natural lighting concerns, and the use of numbers to calculate and reason about sizes and dimensions. These findings confirm the previous results. Below is a description of the five major findings from the current experiment.

5.6.1. Designers Share Drawing Conventions: Symbols and Their Configurations

The conclusion from the experiment sessions and the post-experiment questionnaires is that designers use drawing symbols in their design in a consistent fashion. Participating designers chose primitives from a limited universe of geometric shapes and symbols in their drawings, and composed them in highly conventional ways. They predominantly used lines, ovals and blobs, rectangles, and hatching. Primitives such as ovals and rectangles were drawn with varying size and aspect ratios (Figure 5-29, top)

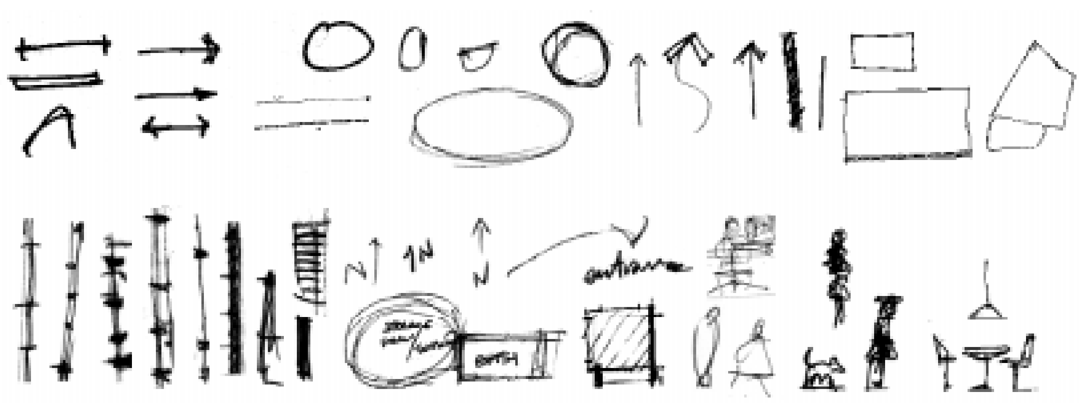


Figure 5-29. Top: Primitives (drawing elements) used in drawing included arrows, lines, hatches and simple geometric shapes. Bottom: Symbols (architectural objects) such as walls, windows and stairs are formed by combination of primitives.

These primitives are combined together as symbols to represent architectural objects. For example, a symbol for direction North was composed with an arrow and a letter N; lines were composed to indicate walls and windows; and circle with lines or a blob was drawn to indicate a person (Figure 5-29).[7]

[7] This finding is the same as from the last chapter (4).

5.6.2. Designers Drew Furniture in Space and to Put Themselves in Context

The results of the experiment suggest that designers draw simple shapes representing furniture in order to put themselves in the right context to think about design. Furniture symbols are arranged in a configuration to represent how the space will look and feels. For example, the top row of Figure 5-30 shows variations of a conference table. Three designers (Roger, Noi and Mario) drew chairs (small rectangles or dots) surrounding the table, and one (Samuel) drew a larger rectangle surrounding the table to see (test) if the space was big enough. Two designers further drew a door and windows along the wall, service counters and a white board for the room. The bottom row of Figure 5-30 shows different furniture placed in space for a lobby and office space.

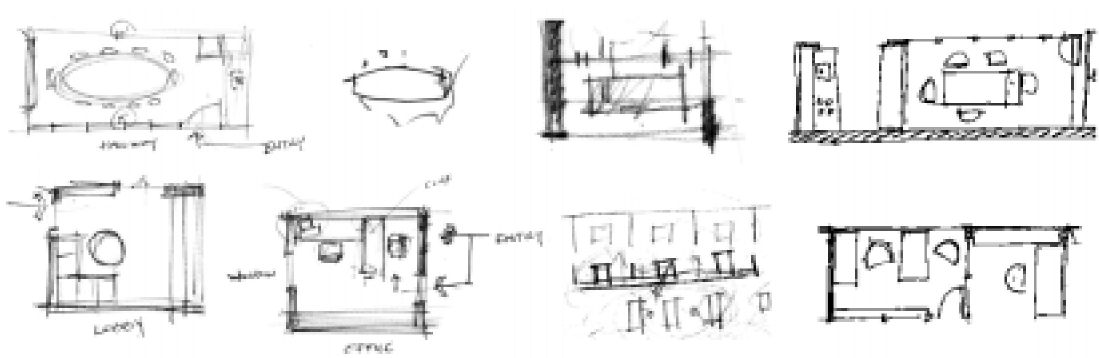


Figure 5-30. Symbols give clues for context. Top row: different designer's symbols for the conference space (Roger, Noi, Samuel, Mario). Bottom row: lobby, office (Roger) and cubicles (Samuel, Mario).

5.6.3. Lighting Concerns are Portrayed in Sectional View with Light Rays

Another finding of the experiment is that different design concerns tend to be correlated with either sectional or plan representations. The experiment asked designers to focus on separate issues such as zoning, lighting, visibility and dimensioning. As in our previous study of diagram making, participating designers seem to share a preference for using plan or section to illustrate certain architectural concepts. For example, they chose a plan view to illustrate relationships between different spaces and zones for layout design and sectional views to illustrate lighting conditions. Figure 5-31 shows lighting included in a sectional view, with light from the sun passing through the building envelope. These representations were made using conventional symbols for the sun, light rays, windows and walls; some drawings also included symbols for persons and computers.

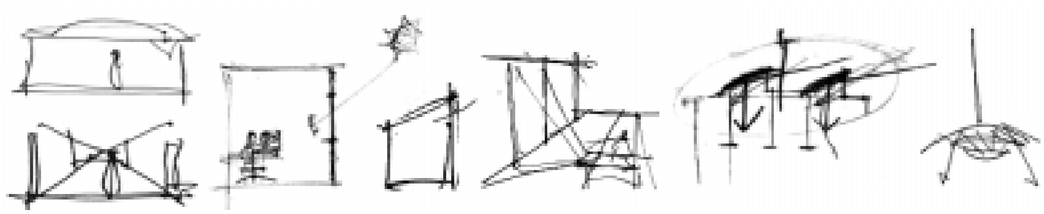


Figure 5-31. Drawings illustrate lighting by using light lines that penetrate building envelope from windows and roof. The changed direction arrows indicate reflecting light from a roof skylight (second to right) and in an interior lighting fixture design (right).

5.6.4. Attention and Focus Can be Identified through Labels and Overtracing

The design drawings from the experiment exhibited that plan drawings have more text labels than sections. The data also revealed that participants frequently included key words from the design concepts as labels in their drawings. Labels for functional spaces were mostly written inside a containing shape (oval, blob, or rectangle), or occasionally put beside the shape with an arrow or line from the label pointing to the shape it identified (see Figure 5-32). The video protocols of the empirical studies showed that designers constantly engaged in overtracing, repeatedly outlining a particular shape or area of the drawing. This overtracing, or redrawing, serves as an act of selection, that draws attention to an element, refines a shape, adds detail to the drawing, or explains something. Designers also used hatching to distinguish a particular space from others.

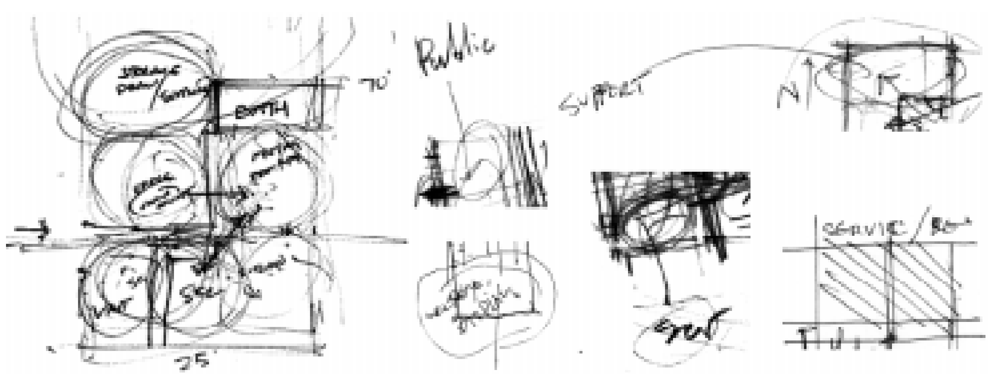


Figure 5-32. [a] a bubble diagram for layout and zoning, [b, c, d, e] designers used overtracing (circle) to select a space, they put labels inside the shapes, or drew lines out to indicate space use, [f] hatching is another way to emphasize a certain area and distinguish among other spaces

5.6.5. Dimensional Reasoning through Figure Calculations

When thinking about allocating objects or spaces within required dimensions, designers wrote down numbers beside the drawing to reason about scale and calculate sizes. For example, three designers (Roger, Noi and Mario) wrote down numbers to label dimensions of the site when reading the design program. One designer (Samuel) reasoned the length and width proportion (3:1) of the site before he drew the floor plan. One designer (Roger), when performing the final task about dimensional requirement, wrote down a sequence of numbers to label dimensions, to calculate the square footage requirement, and to reason about what the width and length of a space should be (Section 5.3.1). One designer (Mario) wrote down numbers to convert feet to meters and to calculate area dimensions.

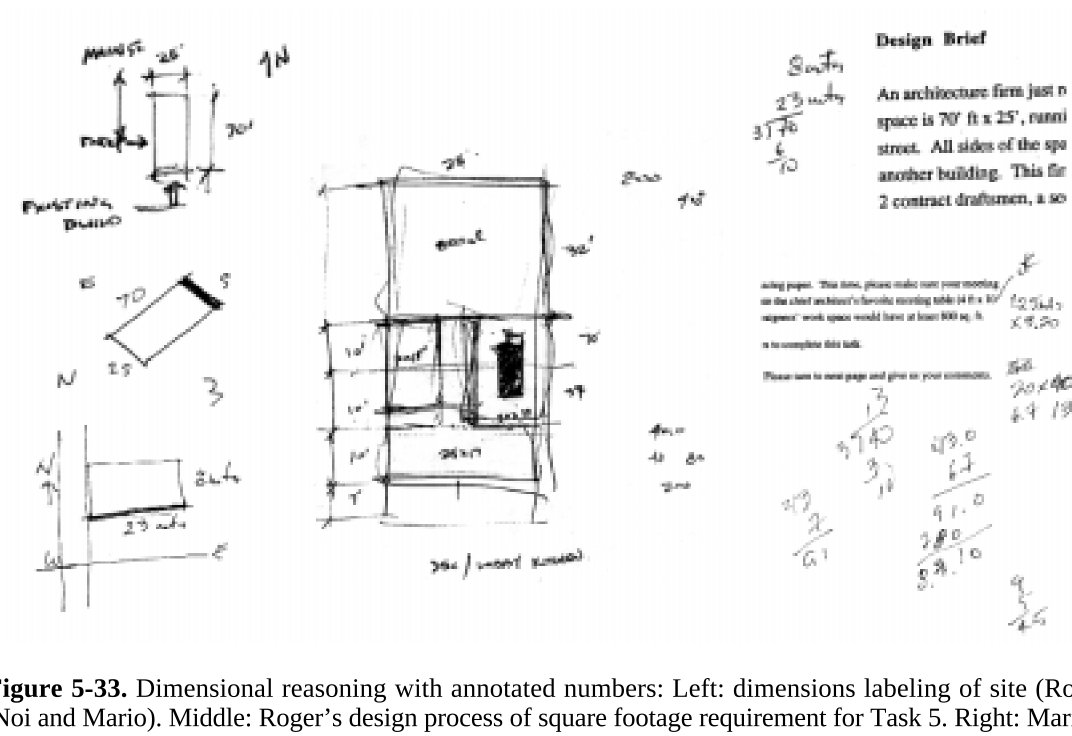


Figure 5-33. Dimensional reasoning with annotated numbers: Left: dimensions labeling of site (Roger, Noi and Mario). Middle: Roger's design process of square footage requirement for Task 5. Right: Mario's calculation and conversion of feet to meter and square footage. top: 70ft x 25ft became 23m x 8m, bottom: 800 sq. ft. was converted to 20ft x 40ft to 6.7m x 13m

5.7. SUMMARY AND CONCLUSIONS

A new empirical study[8] on design drawing is described in this chapter. The study used an architectural office as the design program and asked participating designers to focus on different concerns for each task. This experiment was set up to verify if designers use the same drawing conventions when they think about design concerns and when they design. The answer is yes. The results of the studies showed that the four participating designers shared drawing conventions not just among themselves but also with those who participated in the design diagramming experiment described in Chapter 4.

The design drawing experiment revealed several patterns: 1) Designers share drawing conventions. They consistently use symbols to represent walls, windows, furniture and human figure. 2) Designers combine symbols in specific configuration to indicate design contexts. For example, conference rooms are portrayed as chairs surrounding a long table, and the direction North is indicated by letter N and an arrow. 3) Designer have different drawing preferences for different design concerns. For example, visual access concerns are portrayed in plan view with arrows representing view lines, lighting issues are illustrated in sectional view using light rays that penetrate the building. 4) Designers write down design concepts or space names as labels in their drawings. They overtrace their drawings to draw attention to specific shapes or areas. 5) Designers write down numbers to reason about scale and calculate about dimensions.

There were also some individual differences among the designers: 1) Roger, the functional designer drew "bubble diagrams" for spatial arrangement. He was articulate and

[8] The experiment setups and results are partially reported in (Do & Gross, 1997) and (Do, 1997).

had developed a personal set of symbols for direction, furniture and dimensioning. He wrote down numbers to label dimension and to reason about square footage. 2) Noi, the 3D sketcher was obsessed about turning all his plan and sectional drawings into "3D" drawings. He used lines as "spatial partitions" to arrange space. 3) Samuel, the philosopher talked a lot about what he was doing when he designed. His verbal protocols were very informative. However, his drawing symbols tended to be few and simple. He drew rectangles to represent tables, counters, printers and rooms. He used arrows to represent many concepts such as sun light, entry, view, and circulation. His spatial layout plan was a hybrid of "spatial partitioning" and "bubble diagram." 4) Mario, the research architect made the whole design session to be a testing task by drawing furniture elements. He used a set of symbols for different kinds of furniture, and structure elements such as columns and walls. He wrote down numbers to calculate and convert feet to meters. He copied down program requirements on the tracing paper unlike other designers who underlined or drew shapes to frame space requirement on the design program.

The four design sessions from the empirical studies in this chapter showed evidence that different concerns and context can be identified through drawing conventions. In other words, an intelligent computer program to support design should be able to identity design concerns by recognizing designers' conventional or personalized drawing symbols and configurations. Different design concerns should be supported by different knowledge-based design tools. Therefore, part of the results from this chapter are used for the Right-Tool-Right-Time prototype system described in the following chapter (Chapter 6).

CHAPTER VI

A PROTOTYPE SYSTEM FOR INFERRING DESIGN INTENTION

This chapter describes the implementation of Right-Tool-Right-Time, a freehand sketching system that infers design intentions from symbols and configurations that appear in a designer's drawing. It describes the software architecture of the system, the contexts and intentions of drawing symbols and configurations, and the connections to various design tools.

The system is based on several assumptions. First, in order to be useful, knowledge-based design tools must be available at the right time. Second, different design activities need different kinds of supporting tools. Third, drawing conventions such as symbols and diagrams can serve as clues to activate the right design tools at the right time. The prototype of the Right-Tool-Right-Time system described in this chapter was built to explore how an intention-inferring drawing environment might help designers use knowledge-based design tools more effectively. The Right-Tool-Right-Time system detects drawing symbols and configurations, uses them to infer the designer's intent and task context, and then provides appropriate design tools to support decision making for that task. For example, when thinking about natural lighting, a designer might draw a configuration consisting of a symbol for the sun and an arrow that representing a light ray in section. The presence of these symbol configurations indicates that the designer's current concern is natural lighting. In a specific drawing context (e.g., sectional view) the design concern is called an 'intention' (e.g., natural light). The question then is: can a computer recognize these symbols and configurations and hence their associated intentions? If a computer can infer design intentions from drawing conventions, then these drawing conventions can serve as a trigger to access appropriate design tools targeting the task at hand.

The concept of design intention employed in RTRT starts with the observation that designers constantly engage in different design tasks when designing. Intention is associated with the current task. Many design intentions can exist in (and be inferred from) a single drawing type (or 'context'). For example, 'lighting concerns,' 'ceiling height,' and 'dimensioning' intentions can exist in a drawing context called 'section.' Similarly, 'furniture arrangement,' 'spatial layout' and 'visual analysis' intentions can exist in a context called 'plan.'

The RTRT system described here demonstrates that it is possible for a computer program to recognize drawing symbols and, based on those symbols, activate different design tools. RTRT is a working system built on top of the Electronic Cocktail Napkin environment described in Chapter 3. RTRT was implemented in Macintosh Common Lisp (MCL). The system building efforts and the studies of design drawings were carried out at the same time. Both the system building and the empirical work reinforced each other: the empirical work provided the basis of the system-building and the system-building provided questions and hypotheses for further investigation. Therefore, RTRT is not an ideal system based strictly on empirical studies to support design drawing intentions. Rather, it was built to explore the possibility of recognizing design drawings and intentions with the current implementation.

The chapter is split into five sections: Section 6.1 first explains the RTRT system architecture; Section 6.2, the recognition of context; Section 6.3, the recognition of intention; Section 6.4, the inter-application communication with various design tools; Section 6.5, how the scenario would work, and finally section 6.6, the strengths and weakness of the current implementation.

6.1. SYSTEM OVERVIEW

6.1.1. System Architecture

The Right-Tool-Right-Time system (as shown in Figure 6-1) basically activates different assistants based on inferences made from a design drawing, as the designer engages in a design process.

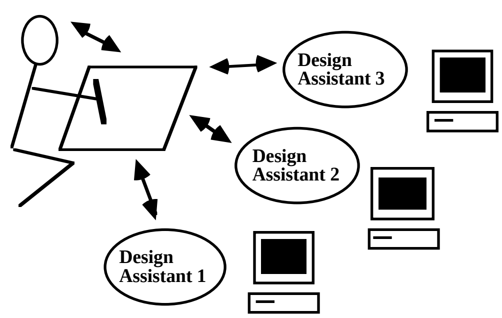


Figure 6-1. The Right-Tool-Right-Time is conceived as a drawing environment with an inference engine that monitors the design drawing and activates different tools.

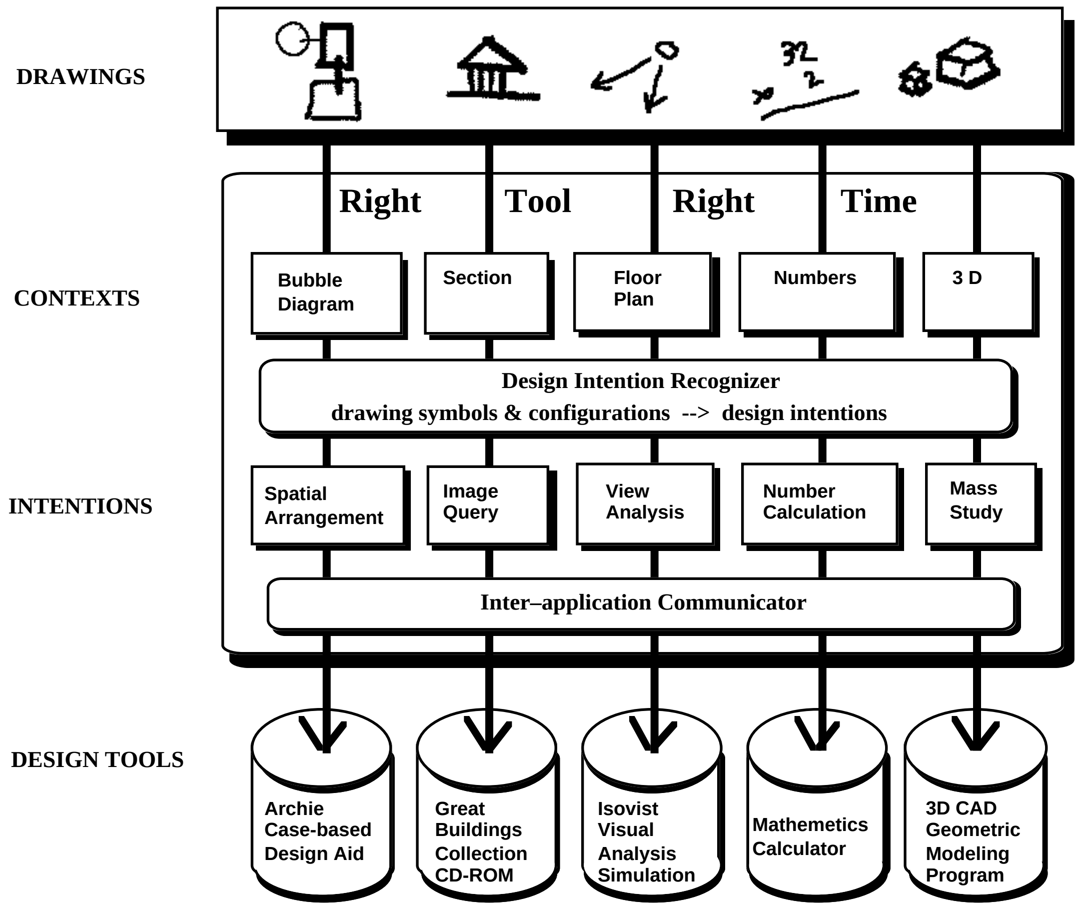


Figure 6-2. The conceptual model of a drawing environment. Top: different drawing symbols and configurations (bubble diagram, facade, view lines, calculation and 3D solid). Middle, the Right-Tool-Right-Time system that 1) identifies drawing contexts (contexts are illustrated inside boxes), 2) design intention recognizer (rounded box) infers design intentions (from symbols and configurations, intentions are illustrated inside boxes), 3) inter-application communicator (rounded box) activates different tools based on detected intentions through drawing symbols and configurations. Bottom: different design tools – Archie case library, the Great Buildings Collection, Isovist viewshed analysis program, a calculator, and a 3D modeling program.

Figure 6-2 shows a conceptual diagram of the Right-Tool-Right-Time control structure.[1] Briefly, the process flow is 1) drawings, 2) contexts, 3) intentions, and 4) design tools. The first row of the diagram shows an example array of different drawing

[1] Although Figure 6-2 is a conceptual diagram, the five design tools identified – Archie, Great Buildings Collection, Isovist, Calculator and 3D modeling program – are all available in the RTRT environment. Here we only show five columns, but RTRT can support more columns once the drawing symbols, contexts and intentions are clearly defined.

fragments – symbols and configurations. From left to right, these fragments are: a bubble diagram of a spatial arrangement, a diagram of a building facade, a viewshed analysis drawing of a person with view lines in a plan view, a numeric calculation, and a 3D solid mass study.

The large middle box of the diagram represents the Right-Tool-Right-Time system. The first recognition effort of RTRT identifies the context of the given drawings (e.g. bubble diagram, section, floor plan, etc.). Then a design intention recognizer[2] (rounded box) attempts to infer design intentions (represented here as boxes) from given drawing symbols and configurations within a recognized context. The intention recognizer matches design intentions and drawing configurations using a translation table. From left to right, these design intentions are: spatial arrangement, image query, view analysis, number calculation, and mass study. The last item in the RTRT system is an inter-application communicator. The inter-application communicator takes intentions and issues commands to different design assistants. The design assistants,[3] or knowledge-based design systems, are portrayed here as cylinders on the bottom row of the diagram. From left to right, these systems are: the Archie case library, the Great Buildings Collection, the Isovist visual analysis program, a calculator, and a three-dimensional modeling program. The inter-application communicator sends commands using a standard event protocol[4] to other programs on the same platform. For example, a command could query a database or

---

[2] The intention recognizer can generate one or more intentions from each drawing. The examples given here about the relationships between drawing and intention are all one–to–one mappings. There could be one–to–many mappings as well.

[3] RTRT issues three kinds of tool activation commands: 1) It brings up a tool such as a calculator or a ruler. 2) It generates specific queries from the drawings to tools like Archie or the Great Buildings Collection. 3) It translates freehand sketching into different programs such as Isovist visual simulation and a 3D modeling program. (See Section 6.4 for details)

[4] Macintosh's Apple Event Handler supports Right-Tool-Right-Time's implementation of inter-application communication. Inter-application communication is further described in Section 6.4. Cross platform communication can be obtained using TCP/IP and is an interesting direction for future work.

activate a visual simulation program to support the task at hand. The vertical lines in the diagram illustrate the flow of information from the design drawing at the top, to design intentions parsed by Right-Tool-Right-Time, and finally to different design tools at the bottom.

6.1.2. Recognition Phases

This section gives an overview of the process of recognizing a design intention. The four phases of recognition, as illustrated in Figure 6-3, are: 1) symbol recognition, 2) configuration recognition, 3) context identification, and 4) intention inference. These phases constitute the control mechanism used by Right-Tool-Right-Time to activate different design tools.

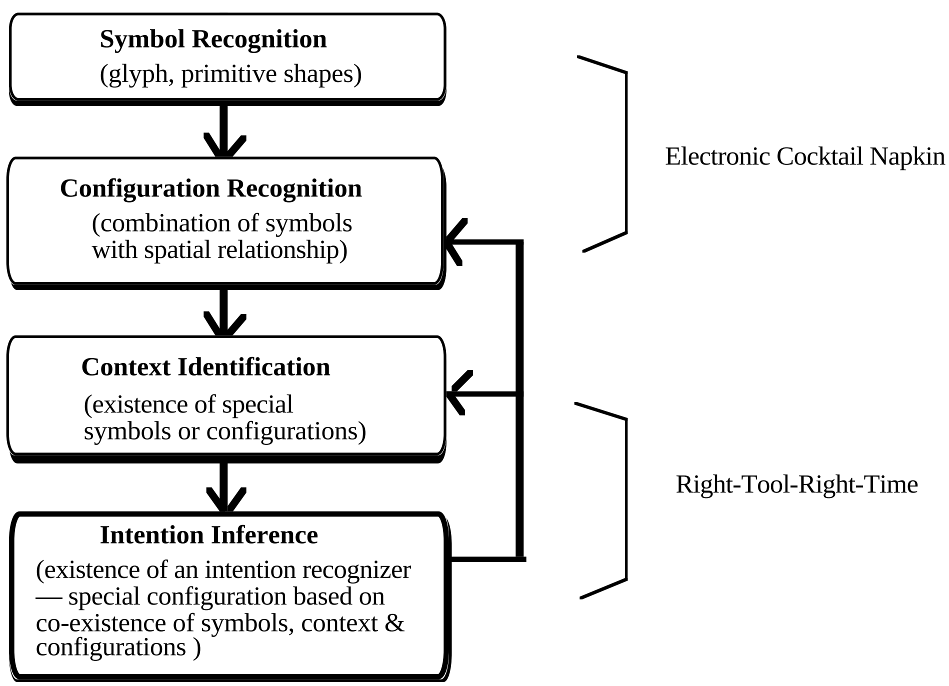


Figure 6-3. Phases of the intention recognition

The first two phases, symbol and configuration recognition, are handled by the Cocktail Napkin. Right-Tool-Right-Time employs these functions and adds two new phases, ‘context’ and ‘intention’ recognition. The following subsection briefly reviews the two phases handled by the Napkin program. Section 6.2 then provides a more detailed description of context and intention recognition.

6.1.3. Recognition of Symbols and Configurations

The Napkin recognizes all drawn primitive shapes (symbols) as they are drawn. First, it compares the features of each drawn shape to previously trained symbol templates (as described in Section 3.3.1) and finds the closest match. For example, Napkin recognizes drawing symbols[5] such as circles, arrows and boxes, which may consist of one stroke or multiple strokes (see Figure 6-4).

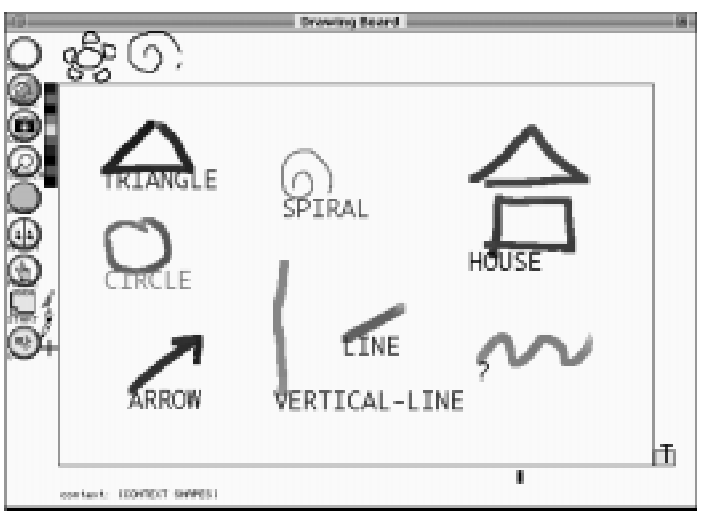


Figure 6-4. Designers’ drawing symbols recognized by the Electronic Cocktail Napkin Program.

[5] The terms ‘primitive shapes,’ ‘symbols’ and ‘glyphs’ are used interchangeably to represent designer’s drawing marks. Glyph is the technical name defined in the Napkin program for these objects. I chose to use the word ‘symbol’ throughout this dissertation. In any case, these drawing marks are simple geometric ‘primitive shapes.’

As shown below, the process for handling symbol recognition consists of two directives: 1) compare new symbol with defined symbol set in the list of the currently known context, 2) if a symbol is unique to a context, make that context the current context.

```
SYMBOL RECOGNITION

    for each *context* in *current-context-chain*
    try matching new symbol in drawing
          against symbols defined in context
    if match is unique to context, set *current-context*
```

Next, Napkin's configuration recognizer considers the spatial relationships between all drawn elements (section 3.3.2). It identifies any configuration in the drawing that matches a previously defined one, groups its components into a compound symbol and displays the name of the identified configuration. For example, a 'window' is identified when a group of symbols in a specific configuration (a vertical line directly connected to two long rectangles, one above and one below) is found (see Figure 6-5). Likewise, a wiggly line drawn directly below a horizontal line is identified as a symbol for the 'ground' (Figure 6-6)

```
CONFIGURATION RECOGNITION

    for each *context *in *current-context-chain*
    map all symbols using mappings of context
    using spatial relations of context
          search for configurations defined in context
    if matching configuration is unique to context
          set *current-context*
    display *current-context*
```

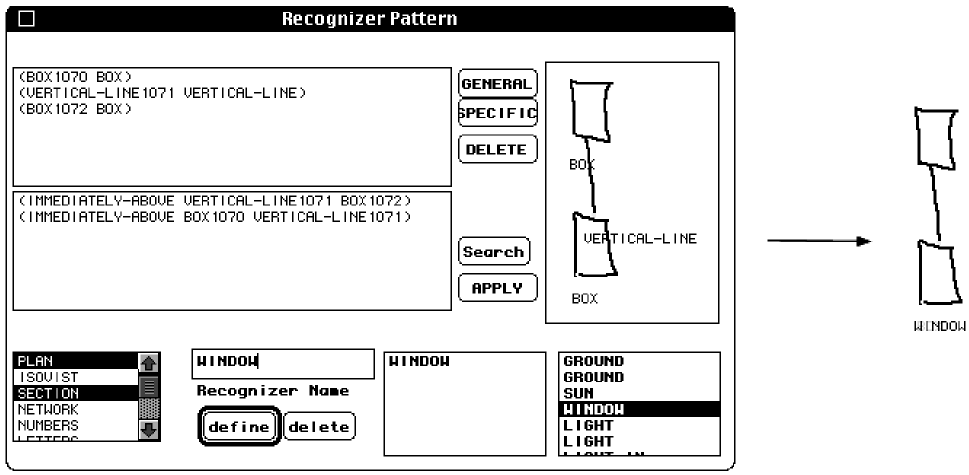


Figure 6-5. A configuration called ‘window’ is identified by recognizing a vertical line directly connected to two long rectangles, one above and one below. (Note that the window symbol can exist in either a plan or a section context, the ambiguity remaining until further symbols are recognized.)

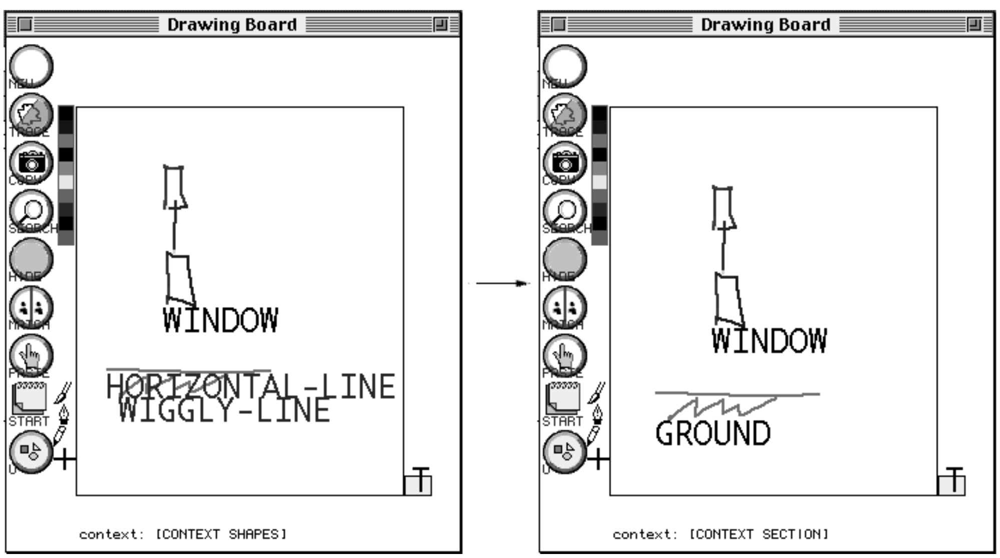
6.2. CONTEXT RECOGNITION

At this stage, Right-Tool-Right-Time takes over to further identify context and intentions. After the Napkin recognizes symbols and configurations, RTRT attempts to identify the context of a drawing. The context recognition function will set the variable *context* to be the identified context and add it to the list of the current contexts (a variable named *current-context-chain*). It then executes any necessary mappings and attempts to recognize all configurations in the current context.

```
SET CURRENT CONTEXT

    set context global variable (*all-contexts*)
    set *context *to head of *current-context-chain* list
    map all symbols in drawing using mappings of context
    recognize all configurations defined in context
```

The configuration ‘ground’ described above is unique to the drawing context called ‘section,’ and so after finding a ‘ground,’ RTRT sets the current context to ‘section’ (see message at the bottom window in Figure 6-6b). Identifying the drawing context affects the further recognition of symbols and configurations. The specific context might involve new interpretations and mappings. For example, the section context maps the ‘box’ and ‘vertical-line’ symbols in ‘window’ to symbols for ‘wall’ and ‘glass.’ The mapping slot in the ‘section’ context describes the interpretation of the shape ‘box’ as the object ‘wall.’



a **b**

Figure 6-6. A ‘ground’ symbol is unique to a ‘section’ context. Therefore, after recognizing the ‘ground’ context (horizontal line directly above wiggly line), the system sets the current context to be ‘section.’

At this stage, the system checks the current context (‘section’) to see if any spatial relationships between the detected configuration and other symbols can constitute another higher-level configuration. This is a recursive and iterative process. If any spatial relations between this configuration (window) and other elements of the drawing (e.g., sun and light ray) indicate a match (light into window, as shown in Figure 6-10), a new configuration will be identified.

The next section explains further how the computer system recognizes and represents a drawing context.

6.2.1. Representation of a Drawing Context

A ‘context’ is a named data structure[6] that stores a list of recognizable graphic symbols, configurations, spatial relations and mappings specific to that particular drawing context. Table 6-1 shows the components of a ‘plan’ context.

Table 6-1. Elements of a Context ‘PLAN’ in four categories: 1) glyph templates, 2) configuration recognizers, 3) spatial-relations, and 4) mappings.

[context plan]	filename: $plan
symbols (glyph)	([glyph room] [glyph wall] [glyph door] [glyph column])
configurations	(#<recognizer double_door> #<recognizer dining_table> #<recognizer desk>)
spatial-relations	((concentric contains overlap)(immediately-above above higher- than) (immediately-right-of right-of) same-size adjacent connect line-connects (lines-connect lines-tee lines intersect line-a- side line-b-side line-neither-side) lines-parallel)
mappings	(((plan wall) (lines line)) ((plan vertical-wall) (lines vertical-line)) ((plan horizontal-wall) (lines horizontal-line)) ((plan room) (shapes box)) ((plan column) (shapes circle)))

[6] A context named ‘plan’ has symbols and configurations such as wall and a double door. The specifications and records of these components for the context ‘plan’ is stored in a file '$PLAN' for future retrieval.

A context has a set of symbol templates (second row of Table 6-1), which store the features needed for recognition such as number of strokes, corner, size etc. Only symbols unique to a context are stored in that particular context. For example, the ‘plan’ context stores symbols found only in plan drawings (e.g., wall, door, column, etc.). Symbols that are shared by more than one specific context are clustered in higher-level context templates. For example, graphic symbols such as circles and arrows, which are common to most drawings, are stored in the general ‘shapes’ context.

A context also has a set of definitions for configurations. For example, a ‘plan’ context (third row, Table 6-1) has definitions for configurations such as a dining table set and a double door.

A set of spatial relations (fourth row, Table 6-1) that are relevant to the context are also listed (e.g., concentric, above, connect). The list of relevant spatial relations limit the generation of uninteresting relations in a given context. For example, in the ‘bubble diagram’ context, the only relevant spatial relationships might be ‘connection’ and ‘containment’ while in the ‘numbers’ context only adjacency may be relevant.

Finally, a set of mappings (last row in Table 6-1) indicate how certain symbol templates in this context are mapped to symbols in other contexts. For example, in the ‘plan’ context a ‘line’ (from the general lines context) is interpreted as ‘wall’ and a ‘box’ is interpreted as a ‘room.’

6.2.2. Initial Context and Current Context

The initial drawing context for Right-Tool-Right-Time is the most general ‘shapes’ context, which includes only common geometric shapes such as circle, box, triangle, and spiral. Three other contexts are available at the basic level. The ‘lines’ context includes many types of lines such as solid line, dashed line, dotted line, vertical and horizontal line, and wiggly line. The ‘numbers’ context includes the number symbols from 0 to 9. The ‘letters’ context includes the twenty-six English letters. Shapes, lines, numbers and letters constitute the basic elements of all diagrams. Other contexts defined in RTRT are ‘plan,’ ‘bubble_diagram,’ ‘section,’ and ‘3D.’ These represent drawing types (or view preferences).

The system maintains a list of all known contexts and a ‘current context chain.’ Recognition and definition takes place in the ‘current context.’ The ‘current context chain’ is an ordered list of the contexts that are searched when a symbol or configuration cannot be recognized in the current context. It specifies a sequence of other contexts that may contribute to recognition. When the current context is set, the system moves the context to the beginning of the current context chain. The chain therefore contains a list of contexts previously recognized, the most recent contexts listed first.

The drawing context is set when RTRT recognizes a symbol that is unique and specific to a context. When the designer starts drawing, the initial context is ‘shapes.’ When the designer stops drawing for two seconds, the program activates the background recognizers. The recognizers look first in the current context for templates to identify the symbols in the drawing. If no match is found, the system searches symbol templates in the other contexts. For each symbol in the drawing, a list of the possible matches with the symbol templates is returned. If the most likely match is in a context other than the current one, then that context is returned as a candidate for 'current context.' If a symbol is unique to this found context, then RTRT sets the current context.

```
if unique symbol
      set *current-context*
      then set *current-context-chain*
if unique configuration
      set *current-context*
      then set *current-context-chain*
```

The drawing context can also be identified when RTRT recognizes a unique, context-specific configuration. For example, in the 'plan' context, a 'rectangle' surrounded by four 'circles' on four sides is defined and recognized as a 'dining_table_set' (see Figure 3-8). Recognizing 'dining_table_set' would set the current context to be 'plan' because this configuration only exists in a floor 'plan' context. Figure 6-7 shows another example in which a 'ground' configuration (horizontal line above wiggly line) would identify a 'section' context.

Figure 6-7 A context specific configuration called 'ground' is identified by a horizontal line directly drawn above a wiggly line. As a result of recognition of this unique configuration, the current context is then set to be 'section.'

In summary, the existence of a unique symbol or a unique configuration indicates a context. Once the program knows a drawing’s context, it can better interpret the symbol elements and configurations.

6.2.3. Mapping Named Symbols for Different Contexts

After a context is recognized, RTRT determines if any further actions need to be taken. As soon as the designer draws a glyph or makes a configuration that the system can identify as belonging to a specific context, then the program adjusts its representation of the current context and calls a mapping function to reinterpret symbol names in the new context. For example, if the ‘plan’ context indicates that ‘line’ in the ‘lines’ context must be mapped to ‘wall’ then all recognized ‘lines’ will be re-interpreted as ‘walls.’ After detecting a ‘door’ symbol in a drawing that has many lines, the context will be set to ‘plan’ and the lines mapped to ‘walls.’

The mapping of symbol templates between different contexts is provided because a symbol can have different meanings in different contexts. For example, a ‘circle’ in the ‘shapes’ context maps to the letter ‘O’ in the ‘letters’ context, to the number zero in ‘numbers’ context, and to a ‘column’ in the ‘plan’ context.

A mapping is a list of the names of 1) new context, 2) new symbol, 3) source context, and 4) source symbol. Mappings are stored in a slot in a context table (Table 6-1, last row). Example code[7] below shows how to map a simple symbol with different interpretations in different contexts:

```
(make-mapping ‘letters ‘o ‘shapes ‘circle)
(make-mapping ‘numbers ‘0 ‘shapes ‘circle)
(make-mapping ‘plan ‘column ‘shapes ‘circle)

(make-mapping ‘plan ‘room ‘shapes ‘box)

(make-mapping ‘plan ‘wall ‘lines ‘line)
(make-mapping’ plan ‘horizontal-wall ‘lines ‘horizontal-line)
(make-mapping ‘plan ‘vertical-wall ‘lines ‘vertical-line)
```

Mapping enables a symbol to be interpreted differently in different contexts, without duplicating the template or repeating the recognition effort. A line is first identified as a line in the general ‘lines’ context. If the current context is ‘plan,’ it will then be interpreted as a ‘wall,’ or if the current context is ‘bubble diagram’ it will be interpreted as a ‘connection.’

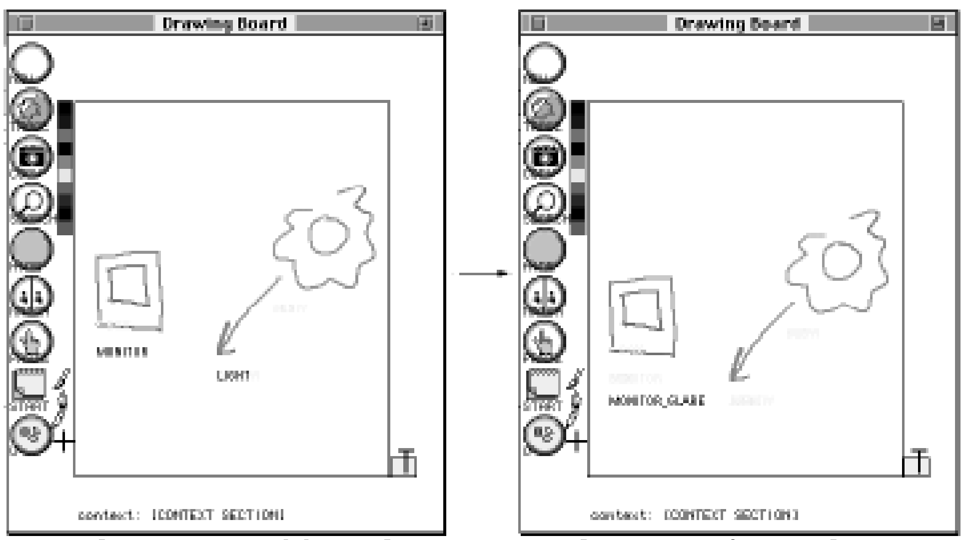
6.3. INTENTION RECOGNITION

The last recognition phase is the inference of intention. An ‘intention’ is represented and recognized as an ‘intention configuration.’ The special feature that makes an intention configuration different from a regular configuration is that its name is stored in the ‘intention’ name slot in the ‘intention-application-mapping’ table. The table is later used to decide which inter-application communication protocols to send to which design tools.

[7] Of course, to make mapping abilities available for designers, a simple interface for ‘make-mapping’ should be built for next generation of Right-Tool-Right-Time. Here the code is provided to illustrate how a ‘make-mapping’ function can be carried out.

6.3.1. Recognition of an Intention Configuration

An ‘intention configuration’ is recognized by the existence and interrelationships of one or more special configurations in the current context. For example, as shown in Figure 6-8, an intention about ‘computer monitor glare’ is named as an intention configuration called ‘monitor_glare’ and identified through the co-existence in a drawing of 1) a sectional drawing context, 2) lighting symbols (e.g., sun and light ray), and 3) a computer monitor symbol.



[context: sectional drawing] [intention configuration]

Figure 6-8. An intention configuration called ‘monitor_glare’ can be defined by co-existence of a lighting symbol configuration ‘light’ (sun and arrow) and a configuration called ‘monitor’ (concentric boxes) in the sectional drawing context.

Many intention configurations are identified by the co-existence of drawing symbols and configurations. For example, the presence of table and chair symbols in a drawing indicates an intention of furniture arrangement, even though these furniture elements need not be arranged in any particular way.

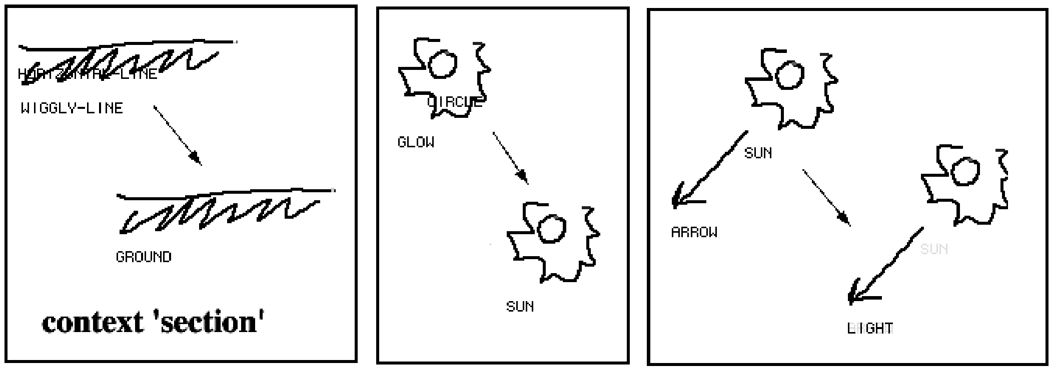
Strictly speaking, an 'intention configuration' is a configuration of other configurations. Intention configurations are higher-level configurations, compounds of the regular configurations. For example, as stated above, an intention configuration called 'monitor_glare' is a configuration of 'monitor' and 'light,' which are themselves configurations of lower level symbols. A monitor symbol is composed of two concentric boxes and light is composed of a circle for the sun and an arrow for a light ray.

Recognizing an intention configuration is an incremental process. First RTRT identifies the current context. Then it runs the configuration recognition function, which identifies any previously defined symbol configurations. Then for every configuration recognized, the system checks the intention-application-mapping table to see if the name of the configuration matches a name of an intention. If so, this configuration is labeled as an 'intention configuration.'

```
INTENTION CONFIGURATION RECOGNITION

    at *current-context*
    run configuration recognition
    for every configuration in the *current-context*
    see if any configuration name
          appears in *intention-application-mapping*
    flag this configuration as *intention*
```

For example, Figure 6-9 shows how the system recognizes intention, from the lowest level of symbols, to configurations, to an intention configuration. First, two 'boxes,' one drawn directly above a 'vertical-line' and one drawn below it, are recognized as a configuration called window. This 'window' configuration is unique to the 'section' context and so the context is set to 'section.' Second, a 'circle' symbol inside a 'glow' light symbol is identified as a configuration called 'sun.' In the context 'section,' a 'sun' with an 'arrow' is identified as an intention configuration called 'light.'



[context: sectional drawing] [intention configuration]

Figure 6-9. Incremental definition and recognition of drawing symbols to configuration, to design intention. Left: a ‘horizontal-line’ above a ‘wiggly-line’ is recognized as a ‘ground’ configuration, this unique configuration sets the context to be ‘section’; Middle: a ‘circle’ inside a ‘glow’ is recognized as a ‘sun’ symbol; Right: ‘sun’ with an ‘arrow’ in the ‘section’ context is recognized as ‘light’ intention.

An intention configuration is a compound of symbols and configurations in a specific context. For example, Figure 6-10 shows that a ‘window’ and a ‘light’ found together in the ‘section’ context are recognized as an ‘intention configuration’ called ‘light_in,’ which indicates an intention about getting light into a building.

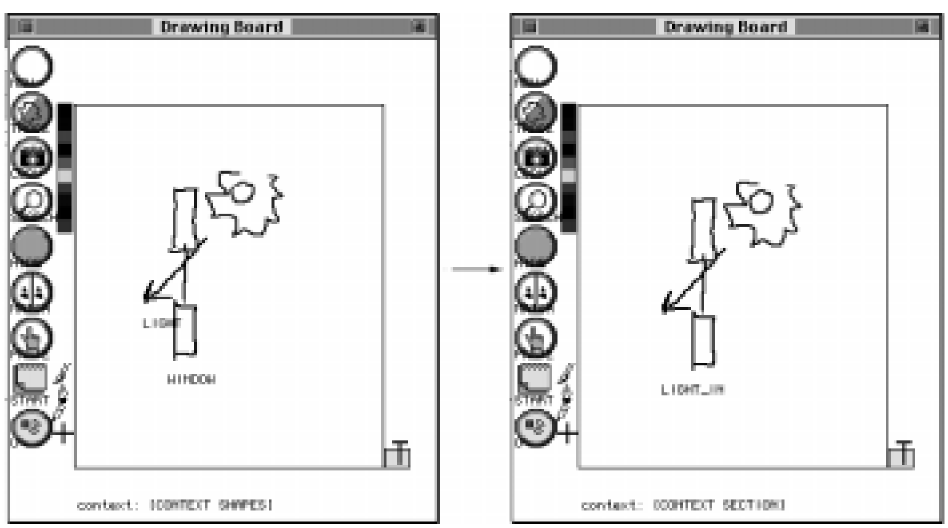


Figure 6-10. An ‘intention configuration’ can be defined by combination of a ‘light’ configuration and a ‘window’ configuration.

6.3.2. Context and Intention

Context and intention are two important concepts. Therefore RTRT uses two different classes of objects to serve their different purposes.

Just identifying a context as ‘section’ will not determine an intention, but it will narrow down the possible intentions to a smaller list (e.g., lighting, ceiling height, building appearance, etc.), which is a subset of the entire intention table. With the existence of lighting symbols in the section context, the intention can be further identified as lighting. By checking only the symbols listed in the current context, Right-Tool-Right-Time limits the search for possible intentions.

Occasionally, a drawing may contain symbols that are used in two different contexts indicating different intentions. For example, a ‘sun’ symbol with light ‘rays’ usually appears in the ‘section’ context and indicates a concern about ‘natural lighting.’ However, on some occasions, the ‘sun’ might appear in a ‘plan’ view with concerns about ‘heating’ or as a ‘metaphor’ for other concerns. Therefore, once a context is recognized (e.g., ‘section’), it will help identify a configuration marking a specific intention (natural lighting), even though it is defined in two contexts (e.g., ‘sun’ in ‘plan’ and ‘section’).

Recognition of an intention configuration proceeds in order of the current context chain as the regular recognizers do. Like these regular recognizers, some intention configuration are ambiguous, and may represent different meanings in different contexts. For example, a configuration ‘number’ could exist in either a ‘plan’ or a ‘section’ context. Knowing the context would not help identify the intention of the ‘number.’ The intention could be recognized either as ‘dimensioning’ or ‘calculation’ (of area) depending on its neighbors. In these cases, recognizing an intention requires checking neighboring symbols (which may be part a configuration) to decide the context. For example, as shown in Figure 6-11, a number with neighboring arrows indicates dimensioning, while numbers together with arithmetic marks (e.g. +, -, =, etc.) indicate a calculation intention (regardless of context).

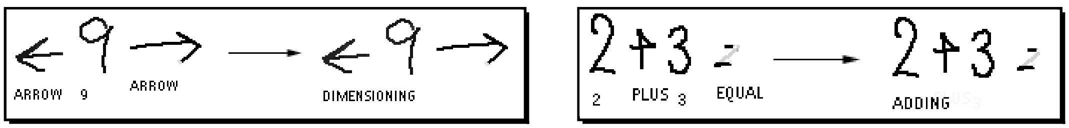


Figure 6-11. A number with neighboring arrows indicates dimensioning, while arithmetic marks together with numbers indicate a calculation intention.

6.4. WHAT THE COMPUTER DOES ONCE IT DETERMINES INTENTION

6.4.1. Identifying the Commands to send to Design Tools

Once an intention configuration is identified, Right-Tool-Right-Time looks up the name of the associated design tool from an intention-application-mapping table. The table is made for those unique symbols and configurations that identify intentions supported by available design tools. Each entry in the intention-application-mapping table (Table 6-2) compiles a list of symbol and configuration names, context names, and related application names.

For example, a day-lighting intention can be identified by finding a lighting intention configuration in the drawing. The configuration ‘light’ is a combination of a ‘sun’ and a ‘light ray’ (arrow) symbol penetrating a ‘window’ configuration. A window configuration consists of a vertical line drawn between two wall sections (rectangular boxes).

Table 6-2. A context, intention, drawing symbols and application table.

intention recognizer	context	symbol and configuration	design tool (application)	command
zoning	bubble diagram	numbers, arrows, space+	Archie Library	activate-archie
orientation	plan	North‡	Archie Library	activate-archie
thermal energy	plan	sun, lighting*	Energy Analysis	activate-energy
furniture layout	plan	dining table	Archie Library	activate-archie
visual perception	plan	walls, person in plan view, sight lines	Isovist Visual Analysis	activate-isovist
spatial layout	plan	door	Archie Library	activate-archie
artificial lighting	section	lamp#	Archie Library	activate-archie
lighting	section	sun, lighting*	Archie Library	activate-archie
monitor glare	section	lighting, computer	Archie Library	activate-archie
ceiling height	section	lamp, table, ground, person	Archie Library	activate-archie
house	section	shapes, house†	Great Buildings Collection	activate-gbc
massing study	3D	cube¤	3D Model Program	activate-3d
multiplying	numbers	multiply, numbers	Scientific Calculator	activate-calculator
adding	numbers	plus, sum	Scientific Calculator	activate-calculator
dimensioning	numbers	numbers, arrows, dimension marks	Ruler	activate-ruler
words	letters	letters, words	Netscape Web Browser	activate-netscape
unknown	anything	anything	Netscape Web Browser	activate-netscape

* ‘lighting’ configuration contains light rays (arrows) that penetrate building envelope (walls & window)
† ‘house’ configuration contains roof (triangle), columns (a row of lines), and a base (a line or a rectangle)
‡ ‘north’ configuration contains a letter N, and an arrow
+ ‘space’ configuration contains a ‘label’ inside a shape, where ‘label’ is a row of letters
¤ a configuration and a symbol can have same name, a cube can be a configuration of lines and shapes
‘lamp’ configuration contains a wire (vertical line) above a lamp cover (triangle)

After identifying an intention by checking the intention-application-mapping translation table, the system finds (from the table) a specific command to send. For example, if RTRT finds that the corresponding design tool (application) name for the 'light' intention is 'Archie,' then it sends a specific command to Archie.[8]

Currently the Right-Tool-Right-Time system supports three types of interactions with the tools: 1) simple activation, 2) query to a search engine, and 3) translation of drawings to different formats. Table 6-3 shows the types of interactions and the commands that belong to each category.

Table 6-3. Interaction types of RTRT to various tools: 1) activation only, 2) keyword query to a search engine, and 3) translation of drawings to different format.

Type	Interaction Type	Name of Tool	Command
1	Activation Only	Calculator	activate–calculator
1	Activation Only	Ruler	activate–ruler
2	Keyword Query*	Archie	activate–archie
2	Keyword Query	Great Buildings Collection	activate–gbc
2	Keyword Query	Netscape Web Browser	activate–netscape
3	Translation	Isovist Visual Simulation	activate–isovist
3	Translation	3D Model	activate–3d

* The names of the recognized intention configurations are used for keyword query to search engines.

The first type of interaction with a design tool is simple activation. For example, when identifying a calculation intention (adding or multiplying), RTRT activates a calculator for the designer to check her numbers. When a 'dimensioning' configuration is

[8] I have arbitrarily chosen Archie to be the tool for this particular intention to illustrate the feasibility of a one-to-one intention to tool mapping. However, it is not difficult to implement a one-to-many or many-to-many mapping in this RTRT scheme. It will 1) require adding new rows of intention and tool names and 2) adding an interface to display multiple tools that have been activated at the same time.

identified, the system looks up ‘dimension’ in the intention-application-mapping table and finds its associated tool to be a ruler and issues an ‘activate-ruler’ command.

The second type of interaction is sending a keyword query to a database search engine such as the Archie case library or the Netscape Web Browser. For example, if the intention configurations recognized from the drawing are ‘light,’ ‘table’ and ‘lamp,’ the ‘activate-netscape’ command is issued, which sends a request for the URL of search engine (Excite’s search page) and fills in the query form with all the identified intention names.

There are several reasons for activating a Web Browser. First, the desired information, as specified by an intention, might, in some cases, only exist on the Web. Second, the Web might be useful as a backup tool, when Right-Tool-Right-Time cannot identify design intentions from the drawing. In such cases, the drawing might still have recognizable symbols and configurations, whose names can be used as keywords in a search engine query aimed at finding relevant information.

The third type of interaction involves translation of freehand drawings to other types of drawings, which can then be analyzed with a specific design tool. For example, lines can be translated into walls and a simulation then run with the Isovist visual analysis program.

A command is a request for action. The command ‘activate-archie,’ for example, issues a specific communication protocol to Archie. The communication protocol is processed using AppleScript[9] and Apple Event Handler. Each command has a uniquely defined function.[10] For example, the 'activate-archie' command takes the name of the intention configuration 'light' and uses it to form a query to Archie to find a set of cases that were previously indexed with the keyword 'light.' The command 'activate-netscape' uses keywords identified from the drawings to form a query to existing Web search engines. Figure 6-12 shows the definition of the function 'activate-netscape.'

```
(defun activate-netscape ()
    (message-dialog (format nil "Activating Netscape Browser..."))
    (netscape-go-to-url
          (format nil "http://search.excite.com/search.gw?search=~A"
                (read-from-string
                      (make-query-string-from-list *intentions*)))))
```

Figure 6-12. A sample code showing the sub-functions of the command 'activate-netscape.' It first brings up a message to inform the user that the system is activating a browser, and then issue a command called 'netscape-go-to-url' to form a query to the browser with the string passed in from the recognized 'intention.'

This function 'activate-netscape' does two things: first, it brings up a message dialog to inform the designer that the Netscape Browser is being activated, and second, it forms a keyword search request to a Web search engine (e.g., Excite) by formatting the names taken from the list of recognized intention configurations (*intentions*). For example, if the list of the intention configurations is [table, light, lamp], then 'activate-netscape' will send Netscape a query "http://search.excite.com/search.gw?search=table+light+lamp." The Web search engine will then return web sites that are indexed by these keywords.

[9] AppleScript is a scripting language standard that was developed to facilitate communication between different applications. The Apple Event Handler identifies elements of an event, its ID number, and a pointer to a handler function and start the procedure of defined functions.

[10] For this prototype system, I have written functions for each different command to specify what kind of protocol to send to communicate with each tool. The future plan includes an interface that allows the user to specify the tools and the interactions they desire by showing examples.

Similar to ‘activate-netscape,’ the command ‘activate-gbc’ sends the identified intention names as a query to the search window of the Great Buildings Collection. The Great Buildings Collection has all its multi-media information organized in HyperCard. Therefore, ‘activate-gbc’ uses the special scripting language of HyperCard (HyperTalk) to form a query and retrieve any information that contains the identified intention names.

Other types of interactions have similar communication patterns. For example, when a designer draws arrows (sight lines) originating from a circle (view point) in a ‘plan’ context that already includes many wall lines, the command ‘activate-isovist’ translates the position of the view point and the locations of the walls in the floorplan and then displays them in the Isovist visual simulation program. Designers can then use the Isovist program to calculate and plot different visual fields while moving the view point around in the floorplan. Similarly, when the designer draws cube symbols in a ‘3D’ context, RTRT will issue the command ‘activate-3d’ to translate the sketchy cubes into a three dimensional model and activate a special program that allows the model to be rotated and viewed from different angles. The last kind of interaction is the easiest. The command ‘activate-calculator’ sends a command to bring up the application calculator. If the application is not running, it will start up that application; if the application is already running, it will activate the application and bring it to the front window.

6.4.2. Design Tools Currently Supported by Right-Tool-Right-Time

Several design tools are connected to the system to demonstrate the concept of an integrated design drawing environment that activates the right tools at the right time. The tools currently employed in the Right-Tool-Right-Time system are Archie, the Great Building Collection CD-ROM, the Isovist visual analysis program, the Netscape Navigator Web browser, a simple 3D modeling program for cubic forms implemented by the author, and a calculator.

Several other points about Right-Tool-Right-Time are worth mentioning. First, a ruler or scale and a grid that can be and snapped to will be included in future versions. Second, rectification of drawings from sketchy lines to cleaned drafting lines is currently supported through Napkin's 'rectify' function. Third, the number of the design tools that Right-Tool-Right-Time currently communicates with does not indicate a limitation on the number of tools that can be incorporated. The Right-Tool-Right-Time project is a prototype that demonstrates that it is possible to link more than one tool to the design drawing environment and to activate them automatically or semi-automatically.

The Isovist program[11] (Do and Gross 1997) aims to help designers understand the visual field of a person in a built environment. An Isovist (Benedikt 1979) is the area in a floor plan visible from a given view point, bounded by partition walls and other physical environments. As a person walks through the floor plan, the area of the Isovist field and its

[11] I have selected this program from among a set of visual and spatial analysis programs I have written. The various implementation approaches of visual analysis are described in (Do & Gross, 1997). The investigation of Isovist was done partly to support a larger set of analysis tools for the College of Architecture at Georgia Tech (Peponis & Wineman, 1995; Wineman & Hodges, 1995).

perimeter change accordingly. The program provides instant feedback and calculation of the visible area from a given position of a person. The program also allows designers to add, move, and remove partitioning walls in the floor plan.

Besides the Isovist visual simulation, other design tools associated with RTRT are: the Great Buildings Collection CD-ROM, Archie case library, a 3D modeling program and a calculator[12]. The Great Buildings Collection CD-ROM is a multi-media presentation of more than eight hundred famous buildings (including photographs, plan, section, animation, 3D model, etc.). Archie is a case library of Post Occupancy Evaluation stories about public buildings such as courthouses and public libraries (see Chapter 3). A simple 3D modeling[13] program was implemented and connected to Right-Tool-Right-Time so that when a designer sketches solid cubic shapes, these shapes can be translated into a 3D model program and viewed from a continuous range of viewpoints. A calculator is included in the suite of design support tools because many examples of numeric calculations appeared in the design drawings that were recorded and studied (see in Chapter 5).

The RTRT prototype system was implemented to make knowledge-based design tools more accessible to designers in early design. However, the eventual usefulness will depend partially on the content of the associated tools and the way information is indexed and organized. A future step might be to focus on what subsets of information would be most useful and how to access them. (For example, future work might identify sites or information resources on the Web, rather than simply doing an open Excite search.)

[12] SciCalc 2.0, Beta software by Steve Costenoble www.hofstra.edu.
[13] The future plan includes connections to commercial CAD software.

6.5. HOW DOES THE SCENARIO WORK?

Chapter 1 introduced a use scenario for how the Right-Tool-Right-Time system would work for a designer. The examples shown there were all real screen shots of the working prototype system. As stated in section 6.4.1 above, RTRT currently supports three types of tool activation. They are: 1) activation only, 2) formation of a query to an existing search engine, and 3) translation and interpretation of a drawing into a particular format for a particular tool. The following briefly describes the three type of interactions.[14]

First kind of the interaction between the RTRT to design decision supporting tool is merely activation. For example, when RTRT recognized numbers and arithmetic marks (e.g., + as a mark for 'plus') in a 'numbers' context, it will infer from the configuration of [Number Left-of Plus Left-of Number Left-of Equal] is an intention of "adding" figures together.

The process of recognizing an intention is done through recognition of low level symbols (numbers 2 and 3, and arithmetic marks + and =) and spatial relations (i.e., immediately left of). The system checks the intention-application-mapping table (as shown in Table 6-2) and decides to issue the "activate-calculator" command.[15] The command sends an activation request to the program called "calculator" and brings it forth for the

[14] This section elaborates section 6.4.2 with more details about functions and processes. It illustrates how the process of the scenario in Chapter 1 would work, using slightly different examples.
[15] Then, of course, this type of interaction can be taken further to the third type. For example, when RTRT recognized the intention is "adding numbers 2 and 3" it could establish a request of inputting these numbers into the calculator (if the calculator allows this request) and could also even pass the answers back to the designers.

designer (Figure 6-13). This type of interaction is the simplest one; it only involves the processing of recognized symbols and configurations in a design context.

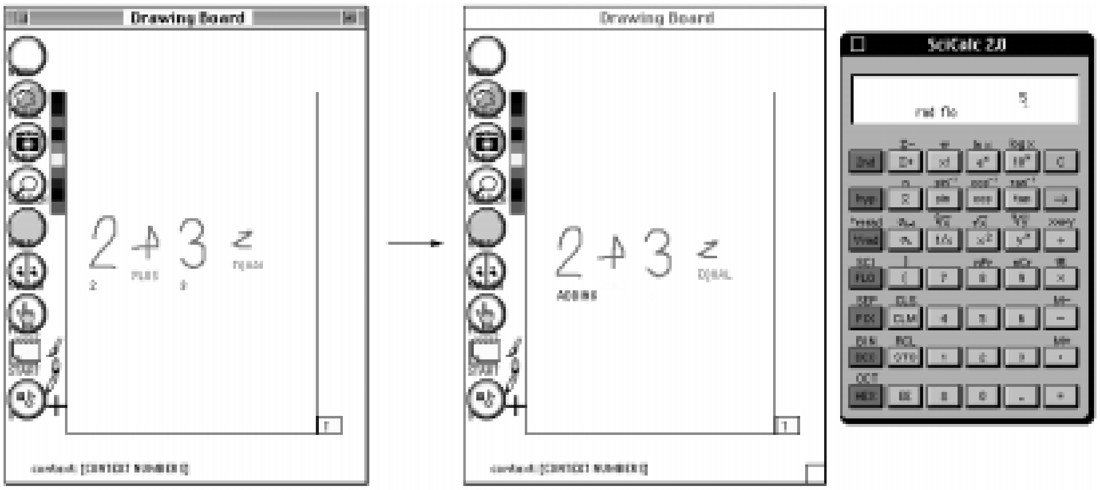


Figure 6-13. A configuration of numbers and arithmetic marks (left) is recognized as an "adding" intention (middle) and RTRT activates a calculator (right).

The second type of interaction goes beyond simple activation by sending the recognized intention as a keyword query to a search engine associated with a library or information base (e.g., Archie case library, the Great Buildings Collection CD-ROM or the Web). For example, a configuration consisting of a 'computer' symbol, a 'sun' symbol and 'light ray' arrow constitute a "monitor_glare" intention. In this case, the lower-level symbols such as the boxes, circles and arrows are first recognized. Then higher-level configurations are identified: a box inside a box is a 'monitor,' a monitor directly on top of another box is a 'computer,' and a sun symbol with an arrow indicates 'lighting.' After these configurations are identified, RTRT further searches for intention configurations and finds that 'computer' and 'lighting' together identify a 'monitor_glare' intention. It then checks the intention-application-mapping table and finds the command protocol to be ‘activate-netscape.’[16]

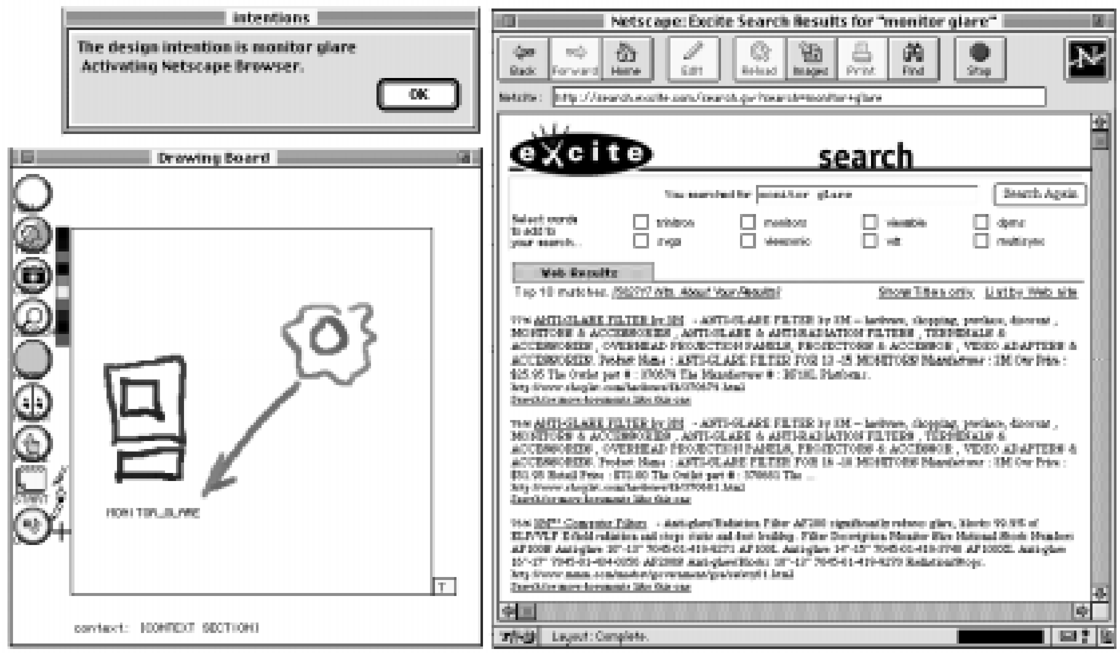


Figure 6-14. A ‘monitor_glare’ intention is recognized (bottom left) and RTRT prompts a message (top left) and activates a Web Browser (right), requests a keyword search and the search returns a list of relevant information.

This command does two things as, illustrated in the sample codes shown in Figure 6-12. First, it informs the designer that the system is activating a Netscape Web Browser. Then it issues a command to the browser to go to Excite’s URL (Uniform Resource Locator, a Web address). This command reads the strings from the current identified ‘intentions’ and formats them as keywords to pass to the search engine URL. For example, the intention ‘monitor_glare’ would be broken into ‘monitor’ and ‘glare’ and a request would be issued to the Netscape browser to go to Excite’s search engine with the query

[16] The name of the tool could be Archie, or any other knowledge-based design tool that is available. Here we only use Netscape Browser with Excite Web search as an example. Other Web search engines are also available. (RTRT can make different search requests to different Web Browsers.)

“http://search.excite.com/search.gw?search=monitor+glare.” The Excite search engine then returns a list of results that match the keywords (Figure 6-14).[17]

The third type of interaction from RTRT to design tools is the most complex of the three. It involves identification of the inferred intentions (based on symbol configurations, the intention-application-mapping table and the current context), activation of the tool and also interpretation and translation of the designer’s drawing into the appropriate data formats. The translation of drawings to the tool-specific format enables designers to use the tool without having to follow complicated inputting procedures. For example, when a “view” (visual perception) intention is recognized, RTRT translates all the lines in the ‘floorplan’ context as ‘wall lines’ and passes them to the simulation program called “Isovist,” which then allows the designer to freely move the view point and receive immediate feedback (plots and calculations of the visible area).

Here is how the process goes. First, RTRT recognizes a context-specific intention configuration called “view” that consists of two arrows. This recognition sets the drawing context as ‘plan’ and then identifies the Isovist visual analysis program as the tool to be activated. The system translates the current drawing into wall lines (from lines) and a view point (from the arrows and the circle symbol) and sends them to the Isovist program (Figure 6-15).

---

[17] Interestingly, the search results suggest how to solve a monitor glare problem to the extent that they include references to companies that make filter screens. If the system instead asked Archie to do a search, it would return many stories from the case library offering architectural solutions.

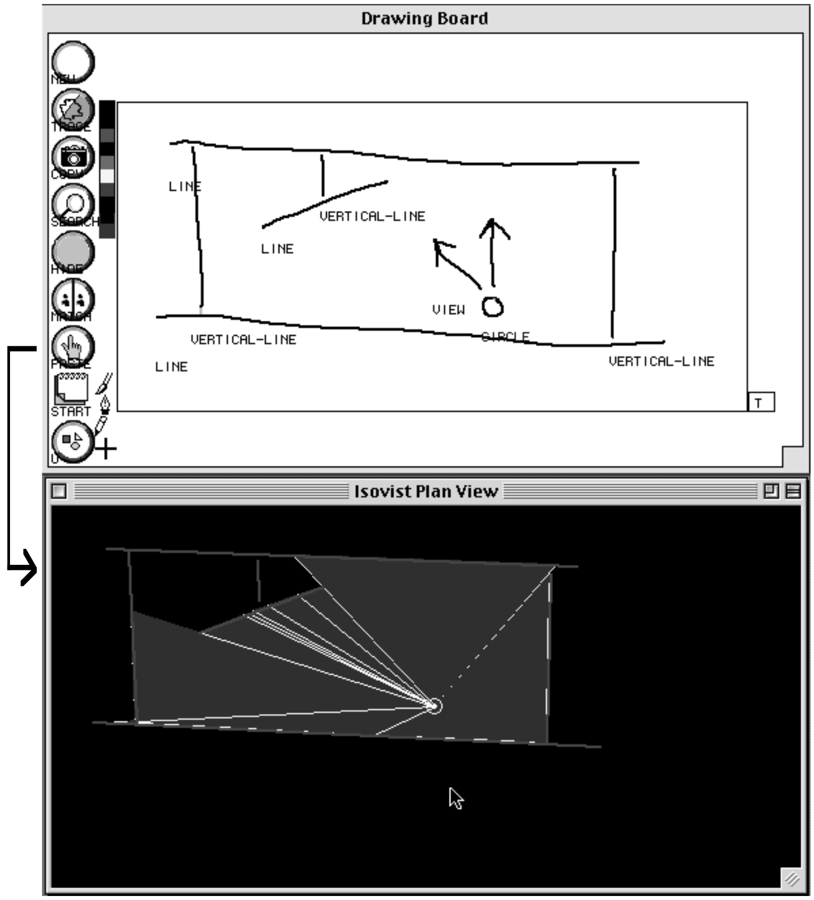


Figure 6-15. A ‘view’ intention is recognized in a ‘plan’ context (top) and the drawing is translated into wall lines (from lines) and a view point (arrows and circle) and sent to the Isovist visual simulation program (bottom).

6.6. DISCUSSION

The purpose of the software implementation is to demonstrate the practical implications[18] of the empirical research: if designers can understand each others’ drawings, it should be possible to program the computer to understand them as well. The

[18] However, the computer implementation is not a direct consequence of the experiments. The programming and the empirical studies took place approximately at the same time.

implementation of Right-Tool-Right-Time serves as a vehicle for formulating and exploring research questions about drawing conventions and design intentions.

Implementing a computer program makes clearer the ways one might perceive and respond to design intentions. For example, how can a machine (as well as a human designer) infer[19] which design tool is most relevant based on the symbols and configurations found in a drawing? How can one decide between one tool and another? These questions can only be investigated through empirical studies.

Building on an existing program (the Napkin) has advantages and disadvantages. It saves time and energy to build on top of an existing program. The hands-on experience makes it easier to see what would and would not work and is more engaging than designing a system from scratch on paper. On the other hand, using an existing program as a platform also imposes limitations and may cause confusion. Right-Tool-Right-Time employs Napkin's 'context' and 'higher-level recognizers' in its intention recognition scheme. There has been a great deal of discussion about what a recognized 'context' is and what exactly 'context' means in a real design drawing session.

The confusion is caused partly by the naming of concepts and objects. Ideally, if a computer system is to support a real world scenario, naming should correspond to real use. However, because an existing program is used, and because the investigations into drawing and design concerns were done at the same time as (and in some cases even later

[19] For example, when is it relevant to provide cases from a slide library and when to retrieve a much broader set of tools? Shall we provide more tools than just one? Here I describe my prototype implementation of linking specific intentions to specific design tools to illustrate that the linking between design intentions and tools can be done. I am aware that the extension of many tools is needed and my prototype represents a preference (or bias) and has room for debate. One direction of the future work as discussed in Chapter 7 would be a general interface that can be customized and trained by the designers for what tools to be activated and what actions to be taken.

than) the system implementation, a gap has appeared between what has been implemented and what the experiments suggest is needed. For example, the scenario in this chapter uses ‘context’ as a way to identify ‘drawing types.’ Therefore, a ‘context’ is the general description of a condition of a drawing. Often the ‘context’ corresponds to the different ‘views’ of the drawing such as ‘bubble diagram,’ ‘floor plan,’ and ‘section.’ These contexts can also be thought of as conventional drawing preferences. A ‘context’ is a certain kind of ‘drawing’ where design events take place. A ‘drawing’ contains symbols and configurations. For example, a ‘bubble diagram’ has different shapes of bubbles, text labels in the bubbles, and lines between the bubbles to indicate connection and adjacency concerns. A ‘section’ has symbols for the ground, wall, window, roof, ceiling, and so on.

The definition of ‘intention’ can be clarified as well. Right-Tool-Right-Time uses Napkin’s higher configuration functions to compose an ‘intention configuration,’ which is a configuration that is specifically connected to a certain design task. An intention indicates a course of action that the designer is, or may be taking. Strictly speaking, an intention configuration is not exactly an ‘intention’ per se, as viewed in the real world (it’s a drawing), but it gives clues as to what the designer might be thinking about.

The next chapter further describes and discusses the implications of building a computer program like Right-Tool-Right-Time and how one might build and use it differently.

CHAPTER VII

CONCLUSION, DISCUSSION AND FUTURE WORK

This chapter summarizes the dissertation, discusses the findings, and outlines future research directions. Inferring intention from drawing has proven to be an interesting and complex problem. Despite the progress of this dissertation many interesting research extensions deserve consideration.

7.1. SUMMARY OF THE DISSERTATION

This dissertation covered three areas: 1) a review of related work, 2) two empirical studies of drawing, and 3) a computer implementation of activating design tools based on recognition of drawing. The main goal was to find out what might be in a designer's drawing that a computer program could recognize and support. This has not been adequately established in the literature to the level of specificity necessary for programming a design aiding tool, and hence therefore, the first focus of the research was the empirical studies. Experiments were designed and conducted to study the relation between drawing conventions and the intentions and concerns of the designer. The second focus of the dissertation was to show that it is possible to program a computer to recognize and interpret these drawing conventions. The Right-Tool-Right-Time prototype program demonstrated how a freehand sketching system could infer intention and support the automated activation of different design tools based on designers' drawing acts. The following briefly summarizes how each chapter addressed these issues.

The first part of the dissertation (Chapters 2 and 3) reviewed related work on drawing importance, empirical studies, computational sketching systems and antecedent projects.

Chapter 2 started by reviewing the role of freehand sketching and diagramming in design. Section 2.1 included a literature survey and case studies of design drawings. Many authors report interviews and observations of famous architects and conclude that designers need to draw to think about design and therefore drawing is important for design thinking. Designers' self report anecdotes also confirm this observation. The second part of Chapter 2 described studies of drawing carried out by cognitive scientists. From the view of cognitive science, drawing is the external representation of concepts that is made to help the processing of information and facilitate problem solving. Section 2.2 described protocol studies as a means to examine drawing in design, and summarizes the results of several design protocol studies. These studies generally conclude that 1) there are different types of information in the drawing such as shapes, things, views, lights and circulation, and 2) there is a relationship between design reasoning and design drawing. However, none of the studies identify the different types of drawing conventions or their relation to design tasks. Therefore, empirical studies were needed to find out the relation of the drawing conventions to different design concerns. This set the background for the empirical studies reported in Chapters 4 and 5. The third part of Chapter 2 reviewed a series of computer sketching programs. It described the focus and important features of these sketching systems. It also argued that computer programs must enable unstructured, freehand drawing to support design activities.

Chapter 3 further described the parent projects for the Right-Tool-Right-Time prototype. It first reviewed design stage models by different researchers and proposed a three phase model comprising: 1) organization, 2) ideation, and 3) fabrication. The design phase model suggests that the different phases of a design process involves different activities and therefore call for different kinds of supporting tools. Then it reviewed the computer-aided design research efforts that are the background to this dissertation, specifically, a case library called Archie, a freehand sketching program called the Electronic Cocktail Napkin, a linking scheme between these two systems, and a Drawing Analogies module that extends the tools connected to the Napkin diagrams.

Archie is a case-based design aid for architectural design. The case library has over three hundred Post Occupancy Evaluation stories of public buildings such as libraries and courthouses. The Cocktail Napkin program is a computer-based design environment built to support hand drawn sketching and diagramming. It emphasizes recognition and interpretation of drawing, management of trace layers, and graphical constraint maintenance.[1] Based on a usability study of Archie that indicated that it needs a drawing interface, a scheme for indexing Archie's case library with diagrams drawn on the Napkin program was conceived and implemented. After connecting Napkin to Archie, the Drawing Analogies module was developed to connect to more design tools, such as architectural slide libraries, and visual reference databases. This then inspired the idea of Right-Tool-Right-Time – a drawing program would not just link to different tools, but also activate

[1] The Right-Tool-Right-Time prototype described in this dissertation is built on top of the Napkin program which handles its lower level recognition.

these tools based on the task at hand. But then what is the task at hand? Can one identify design intention from a design drawing? Therefore, empirical studies on drawing and intent were needed.

The empirical studies (Chapters 4 and 5) were designed and conducted to find out what designers draw that a computer program could understand. The experiments included a questionnaire, a diagramming exercise of design concerns, and design protocol analysis of verbal and video transcripts.

Chapters 4 described empirical studies on diagramming. A series of empirical studies focusing on design problem description and diagramming (Do, 1995a; Do, 1995b). This experiment had designers diagram design concerns (problems and responses) represented case stories from Archie's case library. The experiment took the form of a sequence of very specific tasks (draw from text, write from diagrams, pairing diagram and text, and commenting diagram text pairs) rather than recorded protocol because the data set was collected from a large pool of participants (62) simultaneously. The experiment identified several interesting findings and these were later pursued in more depth with a design experiment (Chapter 5). The findings were that: 1) designers share a universe of symbols for design drawing, and they arrange these symbols in conventional and consistent ways, 2) designers use different view preferences (i.e., plans or sections) for different concepts to illustrate different design concerns (e.g., spatial arrangement versus getting light into a building), 3) keywords from the stories are often used as labels in diagrams, and labels from the diagrams are used as keywords to write stories, and finally, 4) designers can understand and agree with each others' drawings.

Considering that this diagram making empirical study (Chapter 4) was not performed in the context of a real design session, a design experiment was then conducted. Chapter 5 described an empirical study to investigate the relations between drawing and design intentions in a design process. It was designed to further verify if the set of graphic symbols designers use in the design process is a subset of the previous findings. The questions were: Do designers draw different symbols and configurations when dealing with different design concerns? Are these symbols and configurations used consistently?

This experiment specifically asked designers to focus on different concerns as they designed. Participating designers were given a program (brief) to design an architect’s office and their design process was videotaped while an observer took notes. The tasks for the experiment were developed from the previous Archie diagramming experiment results, which are 1) spatial arrangement, 2) lighting, 3) visibility and privacy, and 4) fitting a special piece of furniture into the design. This empirical study (on designing an architect’s office) verified that, within the sample of designers and task chosen for this experiment, designers use different drawing symbols and configurations for different design concerns in a design process. The drawing conventions found in this experiment are a subset of those identified in the previous study. Briefly, the findings were 1) designers share drawing conventions for depicting objects such as windows, walls and human figure, 2) designers combine symbols in specific configurations to indicate design concerns, such as an arrow and a letter N to indicate the direction North, 3) designers have different drawing preferences for different design concerns, for example, lighting issues are usually illustrated in sectional view, 4) designers write down words to indicate design concepts or space names as labels in their drawings, and 5) designers write down numbers to reason about scale and to calculate dimensions.

The last part of the dissertation (Chapter 6) explored the idea and implementation of a prototype system called Right-Tool-Right-Time (RTRT), an integrated freehand sketching environment that attempts to deliver appropriate design tools for the task at hand.

The design computing part of the thesis asked: 1) How can a computer support the drawing tasks identified in the literature and empirical studies described above? 2) What system architecture will allow development of a flexible, extensible drawing environment that can provide automatic or semi-automatic access to a wide range of information resources? 3) How can individual symbols and combinations of symbols be recognized by the computer? 4) How can design studies help computers recognize design tasks, 5) How and when should the various design aids be activated? Chapter 6 described how RTRT recognizes context, intentions and what it does when it recognizes an intention. It attempted to explore how the results from the empirical studies can apply to a computer program even though the program implementation did not implement all the results of those studies. The Right-Tool-Right-Time is a control mechanism. It involves two phases – an initial parsing phase and a secondary processing phase. The initial parsing is done through Cocktail Napkin's basic functionality: 1) the low level recognition of hand drawn glyphs, 2) the higher level recognition of configurations through the analysis of spatial relations among diagram elements. The secondary processing involves: 1) identification of context, 2) identification of intention using a translation table that maps drawing configurations to intentions, and finally 3) the activation of different design tools by issuing commands to other applications. Compared to previous sketching tools, the Right-Tool-Right-Time program; 1) allows users to continue to work in a "sketchy" mode without rectifying objects; 2) makes high-level inferences about designers' intentions based on complex systems of symbols, as well as individual symbols; 3) provides access to a wide and extensible range of information tools.

The current Right-Tool-Right-Time system locates and presents useful information in three ways. First, the system extracts design intentions from drawing and issues them as keywords to query a database such as Archie, the Great Buildings Collection and the World Wide Web through a Netscape Browser. For example, if the intention is identified as ‘lighting’ and the tool is to be ‘Archie’ the system forms a query of ‘lighting’ as a keyword to search the Archie case library. If the tool is identified as Netscape, it sends a query for the keywords identified from the drawings to existing Web search engines. Second, the system imports the designer’s freehand sketch into different programs such as an Isovist visual analysis simulation or a three-dimensional modeler. Third, the system can merely bring up a tool based on the identification of intention. For example, when identifying a calculation intention (adding or multiplying), RTRT brings up a scientific calculator for the designer to check her numbers.

In sum, this dissertation aims to identify whether certain drawing conventions are frequently associated with certain design intentions and whether they can be identified by a computer program. The empirical studies were conducted to identify design drawing conventions, their associated tasks and potentially useful design tools. The computer implementation incorporated findings from the empirical studies to support automated tool activation, and provided a context and intention detection mechanism. It was argued that successful design-aiding tools must be grounded in a detailed understanding of design. Therefore this dissertation focuses more on the empirical studies of drawing, supplementing the experiments with a prototype computer system to illustrate that the results from the empirical studies can be used to support building a computer system. The Right-Tool-Right-Time system is implemented on top of the Napkin program, providing activation links to selected design aids to demonstrate the computability of design drawing.

The purpose of the implementation is not to build a system that supports all design activities, but to demonstrate how different design aids might be activated based on inferences from drawing symbols and configurations. As a prototype system, only a restricted set of tools and a one-to-one connection of intentions to tools was implemented.

7.2. DISCUSSIONS

7.2.1. Contribution of this Dissertation and the Lessons Learned

This dissertation is composed of empirical studies and a prototype system implementation. The experiments show that designers share drawing conventions when they think about design concerns. The prototype computer system, Right-Tool-Right-Time, illustrates an architecture for accessing design tools through design drawing.

The work employs the findings from the empirical studies (symbol conventions used in different contexts and for different intentions) to implement the Right-Tool-Right-Time prototype in a freehand sketching environment. This work shows that it is possible for a human or a computer to make intelligent guesses about the designer's intent by observing the drawing marks the designer is making. Right-Tool-Right-Time demonstrates that an intention inferring system can be built based on recognizing design drawing configurations. Furthermore, the Right-Tool-Right-Time system could encourage designers to use different design tools by providing access to different knowledge-based design systems and a search engine to the Web.[2]

[2] An obvious future project is to conduct an evaluation of how useful the Right-Tool-Right-Time system is at the current stage and what additional functionality should be included.

The prototype aims to provide information access during the conceptual design phases when designers engage in freehand sketching. Besides serving as the aim of this research, the prototype also brings up several interesting concerns. Some implications, are particularly significant. For example, supporting the freehand sketching may invite serendipity due to ambiguity. On the other hand, the power of providing information to designers in the early design phases could also encourage designers to commit to decisions too soon. Providing designers access to information early in the process could change the way that people design.

A major contribution of this dissertation lies with the studies on design drawing. The work of this dissertation goes beyond previous protocol studies that link drawing and verbal self-reports of design thinking by identifying specific drawing conventions associated with particular intentions. In other words, in the domain of architectural design, the graphical marks that designers make are conventional and correspond to the specific tasks that they engage in as they solve a problem. This demonstrates that specific design concerns and tasks within domains have specific vocabularies. In working on a problem that requires various and distinct subtasks, the differences in drawing marks offer a way to infer the designer's intention. For example, in working on an architectural design problem, a designer draws certain symbols and configurations when considering spatial arrangement of functions, others when considering lighting and visual access, and still others when examining dimensional fit.

This dissertation also reveals drawing type preferences as they are associated with design tasks (e.g., plan view for functional layouts and section for lighting concerns). It shows the importance of drawing type and context in interpreting symbols and configurations to infer design intention. It takes a crack at how these might be embedded computationally. Because the drawing marks are conventional and not idiosyncratic, they can be recognized and interpreted by a program and used to govern and guide human-computer interaction. In a computer system to support architectural design, for example, identifying the designer's working intention enables the system to offer task-relevant advice, whether prompting the designer actively or making the advice available on demand.

This dissertation makes contribution not only to the field of architecture, but also to the understanding of interactive design-aid systems in other domains; lessons learned can apply to other design domains as well. The prototype system Right-Tool-Right-Time focuses on human-computer interaction, and has implications for computer-aided tools for design. The focus of this research was on architectural design and the drawing conventions that architects use when thinking about various architectural design tasks. The examples given in this research came from the domain of architectural design, and they show that the early conceptual design stages involve many different tasks and domains. It seems highly likely that other design disciplines (e.g., mechanical or electrical engineering) also have drawing conventions that are associated with specific design tasks and intentions. If this is the case, the RTRT approach could be used to access a suite of tools in those domains as well. It is highly possible that for other design domains similar results will obtain (e.g., drawing conventions will be linked to intentions).

From a computational standpoint, the novelty of this research is that it goes beyond other approaches (e.g., SILK (Landay, 1996)) in that 1) it is end-user programmable, and 2) it takes into account context and intention. It is believed that by giving designers the ability to program the recognition of symbols and configurations will increase the usability of the tools. Most systems taken the opposite approach by providing only one way to interact with the program and can not be customized by the users. For example, hand writing recognition programs (e.g., Graffiti) ask user to learn to write with particular stroke sequence and curvature to input data into a Personal Digitizing Assistant (i.e., Palm Pilot and Newton). It is also believed that context is critical in the interpretation of drawings. For example, arrows can mean many different things; it could denote a light ray in a section or an entry in a floor plan. Another interesting point about this dissertation is that it deals with recognition of drawing at a level of granularity similar to where idioms and phrases lie in natural language processing.[3] Right-Tool-Right-Time takes into account not only the recognition of distinct freehand drawing symbols but also configurations of symbols.

The RTRT approach could provide designers useful information. However, it could also change the way that people design with computers. Some may object to an overly eager assistant that constantly brings forth tools and information even when it is not wanted. At the level of implementation, clearly, an “eagerness” setting is needed, to allow the designer to control the level and frequency of the computer’s interruptions. More generally, this issue raised a larger question: Should computers be “transparent pencils” supporting the ways that designers currently work? If RTRT changes the way people design, because they have access to more information earlier, is this good or bad?

---

[3] Here are some examples that illustrate the importance of recognizing phrases and idioms in machine translation of natural language. For example, SYSTRAN system translates ‘hydraulic ram’ into ‘water goat’ (p. 169 (Simons, 1984)). Boden described a machine translation from English to Russian and back to English will translate the proverb of “The spirit is willing but the flesh is weak” into “The whisky is fine but the steak is not so good” (p. 166 (Boden, 1987)).

7.2.2. Limitations of the Current Work

This dissertation covers three major parts. The first part of the dissertation is a literature review of drawing, cognitive studies and sketch systems, and a description of pilot projects for RTRT. The second part comprises empirical studies of diagramming design intentions and a design drawing experiment. Finally, the third part describes implementation of a prototype system. The three parts make a complete story – starting with background, focusing on empirical studies and ending with a prototype system. Each of the three areas of focus could be refined, discussed with a greater degree of depth and completeness.

The literature study of the importance of drawing in design, and architect's self reports of the use of drawings in their design process could have been a source of identifying drawing conventions that famous architects use. Examining drawings from all the books and classifying the types of drawing symbols and configurations could be an interesting project of its own. To compile a collection of published architects' drawing conventions could be a way to further validate the findings drawn from the empirical studies chapters 4 and 5. However, because of time and copyright concerns in presenting the work, this approach was not taken.

The diagramming experiment reported in Chapter 4 used Archie's stories as test material and therefore the resulting drawing conventions all heavily emphasized the functional aspects of design concerns. This is appropriate because in the conceptual design stage architects are often concerned with use. However, the creative part of the early design process includes tasks such as finding analogies and borrowing forms, which were not addressed[4] in the empirical studies. The design experiment chose an architect's office as the design brief because all participating designers would have knowledge about this subject. It is believed that the general functional issues with different building types would remain the same[5]. While the overlap between the symbols found in the literature and those found in the empirical studies is encouraging, however, this research does not provide the evidence and cannot claim that the drawing symbols and configurations found are applicable to all other building types, or to all tasks.

Another limitation has to do with analyzing and collecting the design drawing data. The diagramming experiment collected a large data set - 62 participants, 4 tasks with 6 questions each, a total of more than one thousand answers. All experimental data was recorded on paper, and no audio or video information was collected. When occasionally ambiguous readings of the data arose (only three cases in the whole data set), it was hard to find the original participant to verify reviewer's reading[6].

The design experiment reported in Chapter 5 took a different approach. It engaged video and verbal protocols and collected drawings designers made. However, the sequence of drawing and the process of overtracing cannot be easily extracted from the final static drawing. It would be more informative if one could display the sequence in which the drawing was made. A stop frame from the video is less clear and of a lower resolution than a paper drawing. A digital drawing device that could record and play back the drawing sequence would therefore be useful. Although the studies of the dynamic nature of drawing was beyond the scope of this research, it may be valuable to automate the experiment

[4] Nevertheless, a prototype computer implementation called Drawing Analogies (see Appendix A) attempts to address the creative process of form borrowing in design.
[5] One can recognize that the drawing conventions found from the diagramming experiment with library and courthouse stories bear similarities with the architectural office design experiment with close examination.
[6] In this case, however, I managed to track down two participants to explain their reading.

apparatus by using a computer-based drawing environment to record, collect and analyze data.

Several drawing symbols and configurations are identified as associated with specific design intentions from the case studies and experiments reported in the previous chapters. Chapter 6 describes the prototype implementation. The Right-Tool-Right-Time employs the findings regarding symbols and configuration conventions as identification clues for intentions. The design tasks used to identify drawing conventions with design intentions were: lighting and visual analysis, furniture layout and spatial configuration. Therefore, the prototype of the Right-Tool-Right-Time was implemented only to support these "design intentions." However, RTRT has the potential of activating additional supporting tools as long as the design intentions and the drawing conventions can be identified. Right-Tool-Right-Time is not an ideal system, because it only covers part of the findings from previous chapters. However, it illustrates the potential for further exploration.

The purpose of the Right-Tool-Right-Time system is to allow the designer to remain in design sketching activities while relevant and useful information is found. However, the system, if too eager to jump up and make suggestions, could potentially obstruct the design flow. Therefore, the obvious important next concern will be to develop an interface that allows designer to continue without interruption and that can be customized to fit an individual designer's style and needs. One important task is to consider how useful tools can be presented less obtrusively in a design environment by simply flagging or highlighting them with icons in the background, or by allowing different working modes for different phases of design process.

Currently Right-Tool-Right-Time only supports a one-to-one relation of intention to tool activation. In real design activities, there are many important facets of design that will need more than one tool per intention. These issues are beyond the scope of this research. However, one-to-many or many-to-many relations of intention to tool could certainly be implemented by allowing designers to add multiple tools to the intention-application-mapping table with the same intention names. An interface for adding new tool names and specifying what actions the system should send to the tools could be added.

This dissertation began with a study of design diagrams to explore the feasibility of using a diagram-based index to retrieve design information from a system like Archie. Evidence was found that diagrams are used conventionally among designers. This supports the feasibility of computer support for visual note taking, and for indexing a database of design information. The study shows that designers can read other designer's diagrams and that they often use symbols to illustrate their ideas. Current CAD and Draw programs provide symbol libraries as tool palettes. But in the diagramming experiment many participants used blobs (indefinite shapes) to make diagrams. When asked, these designers explained that when they were thinking about conceptual design, the specific form was not the main issue and they would not like to be confined to only certain symbols to select from. This suggests that a computational environment should support freehand sketching and the ambiguity of indefinite drawing.

The results from the empirical studies show that designer share drawing conventions. They draw symbols and configurations to think about design concerns. It is, after all, unsurprising that architects share drawing conventions. This might be a product of their design education. Why then, does one need to conduct empirical studies to validate a known fact? This is an interesting question. On one hand, some people oppose and doubt the idea that a computer can recognize designer's drawing. They argue that every designer has his or her own drawing style and therefore there is no consistent drawing convention for a computer to understand. On the other hand, some people argue that design education teaches architects to use drawing conventions and so there is no need to prove it. It might be so. However, even if the conventions do exist, no existence of the design studies and cognitive studies identify the kinds of graphic symbols designers use in design drawings (see Chapter 2). Furthermore, the purpose of the empirical studies is to investigate what types of symbols and configurations shared by designers could be recognized by a computer. If the computer can understand a designer's drawing conventions and their associated design concerns, perhaps it can help designer find relevant information more easily by providing the right tool at the right time for the design task at hand.

The findings from the empirical studies validates the hypothesis that designers draw symbols and arrange their configurations differently when thinking about different design concerns. The conclusions from the literature review and the empirical studies suggest that one can describe the relationships between drawing symbols and designers' intentions in different design contexts. For example, when thinking about lighting concerns, designer draws section diagram of a sun and light ray that penetrate a window. If this is true, then an intelligent assistant (be it human or computer) could offer the right tool for the task at hand. For example, a layout bubble diagram can activate design cases with similar configurations. View lines drawn on a floor plan can bring up a visual analysis tool. The Right-Tool-Right-Time was built to prove this concept. It employs a freehand sketching environment that allows designers to stay involved in the task – drawing and thinking – while providing context-sensitive feedback when detecting different drawing conventions and contexts.

Drawing, of course, is more than just static representation. It is a process and hence contains dynamic information pertaining to many different kinds of activities at once. The speed, pressure and sequence of drawing could reflect the thinking involved during the design activities. The design drawing experiment revealed some information of sequential design reasoning that is worth investigating further.[7] The convention of drawings therefore, is not just merely a collection of symbols, configurations, intentions and context. It could further include the history, the transformation pattern of the drawing (overtracing to modify shapes) and a personal preference of how the pen touches paper.

7.3. FUTURE WORK

There are many future research directions that could extend the scope of this dissertation. For example, one could look into the dynamic character of a design drawing process. The speed, pressure and sequence of drawing may reflect the thinking involved during the design activities. One direction of future research could look at relationships between different types of objects, different levels of hierarchies and compositions. One could also investigate the function of verbal expression in a design process. Furthermore, a study could focus on how diagrams are transformed into sketches, schematic drawings and ultimately construction drafting.

Currently, Right-Tool-Right-Time uses Napkin's 'context' and 'higher level recognizers' to identify meaningful configurations and to determine design intention and potentially useful tools. The idea about context, drawing type and intention could be

[7] Roger's dimensional reasoning described in section 5.3.1 and Mario's repetitive sequence of arranging multiple furniture at the same time (see Appendix H video transcript) are examples of non-static information of a design drawing process.

developed further. One could explore how sub-contexts and sub-intentions can influence the determination of an intention. Future research could investigate other ways to represent design intentions, instead of the current scheme of using a ‘high level recognizer’ for defining both design symbol configurations and intention recognizers. This scheme is simple and it works because the symbols and configurations found from the empirical studies are mostly combinations of design elements with specific spatial relations.

One could do a usability study of the Right-Tool-Right-Time drawing environment. A way to verify this is to invite those designers who participated the empirical studies to design using the Right-Tool-Right-Time system. The experiment would involve designers training the system to recognize their own personal drawing symbols and configurations. Further experiments could compare the advantages and disadvantages of a drawing interface with a keyword searching interface to access knowledge-based design systems. An architectural story can be accessed in different ways, each of which has different virtues. For example, a keyword search maybe be most powerful when the designers know exactly what the design problems are. On the other hand, a diagram might support different readings based on context and shape information. The system might remind designers of a case story by recognizing simple geometric shapes and spatial relationships before they can formulate the correct text specification.

There could also be more ways to use drawing as an programmable interface to access different kinds of knowledge-based systems. The keywords associated with drawing symbols can be entered by the designer. Or the program could build a table of keywords for each recognized symbol and configuration. Or it could gather keywords from other linked databases with similar features of drawing symbols and configurations. One could also investigate how words and images can work together to access extended databases.

It could be interesting to try out RTRT in a studio setting or to conduct empirical studies and observation in a designer's office. The testing might reveal problems with the current prototype system and suggest additional functionality. Furthermore, studio testing would provide an opportunity to study when and why designers turn to different kinds of design support such as visual collections, magazine clippings, a scale, a calculator or the building codes. It would also be interesting to find out how differently designers draw in different settings: at their own desk, in a meeting, or in a discussion with colleagues or clients.

In the other direction, further implementation to extract features of symbol primitives is needed. The current system considers all different rectangles to be instances of a 'box' and both circles and ovals are identified as 'circle.' A symbol can have different aspect ratio and thus should be called different names. For example, a long rectangle and a square, an oval and a circle should be differentiated. It is obvious that though these shapes belong to the same class, sometimes a fine differentiation is needed to identify a configuration. For example, a circle inside an oval can indicate a human figure as seen in plan while a circle inside another circle could represent a well. Another feature to pursue is the identification of the direction of a shape. Currently the program only recognizes shapes (e.g., triangle, arrow) without inferring their directions (e.g., up, down, left, right) even though the data are recorded. For example, an individual symbol can indicate direction such as an up facing arrow, or a triangle pointing to the right.[8]

[8] Recognizing directions of shapes will help identify more complex symbols. For example, to identify a "north entrance" on a floor plan will need the program to infer the compass direction from the arrow that points to the letter N, and identify the "entry" symbols on north facing walls (not just the upper part of the

Another direction is to further extend the user interface of the Right-Tool-Right-Time with a “learning by example” module. For example, the designer can specify how the Right-Tool-Right-Time system should call up different design tools by showing step by step procedures and actions. These actions could be recorded by the system and used later to decide what actions to take when similar drawing intentions are recognized. Then of course, by using appropriate communication protocols, the interaction between the drawing environment and the tools can go across applications and between different platforms.[9]

Furthermore, the Right-Tool-Right-Time system could go beyond just mere activation of design tools. Instead just activating tools from recognition of the design drawing, the system could also bring feedback from the design tools back to the drawing environment. This integration of the drawing environment with different knowledge-based design tools that can inform and provide feedback to the drawing board could be useful and is worth investigating in future work.

---

drawing. Another example is that even though a circle above a triangle can be recognized as a symbol for human figure, a down facing triangle with a circle above it could imply the symbol to be “man” and an up facing triangle with a circle above it to be “woman.”

[9] For example, by using a JAVA applet or a TCP/IP server, we can send commands to different platforms.

APPENDIX A

DRAWING ANALOGIES

Drawing Analogies is a program that uses a Sketchbook module to serve as an interface to various knowledge-based design tools. The Sketchbook module enables designers to keep their personal sketches as visual bookmarks, which can then be used to query several databases, including a commercial CD-ROM of famous architecture called "The Great Buildings Collection" (Matthews 1994), the Archie case based library, various FileMaker Pro and HyperCard databases, and the World Wide Web pages (through the Netscape browser). While browsing the a database, a designer draws a diagram on the Sketchbook to link it with the currently displayed database record, forming a visual bookmark. To retrieve a record, a designer draws a diagram on the Drawing Board, and the Drawing Analogies module compares the query with previously linked diagrams to find the closest match. Designers can keep their personal Sketchbooks and refer to different information for future use. In this drawing environment, the connections to various design tools are direct, one-to-one, and require the designer to take explicit action to invoke them.

A.1. FINDING AND USING VISUAL REFERENCES IN DESIGN

Imagine an architect engaged in the design of a natural history library. She is an experienced designer and sophisticated in layout arrangement, and for years she has kept a journal containing sketches, notes, and clippings from magazines and design folios. We join her in the early stages of design as she seeks inspiration for her building form. Often she finds ideas in her journal for the physical shape of the building and incorporates them into her design. As we join her, she has proposed a basic scheme for the building's spatial arrangement by drawing a bubble diagram on her tracing paper (see Figure A-1). In her diagram, four bubbles surround a central object depicting the four major functional spaces in the library program. The adult's section, children's reading room, reference area, and staff office are all connected with via a central service lobby with to the patron's entrance and the service entrance on opposite sides.

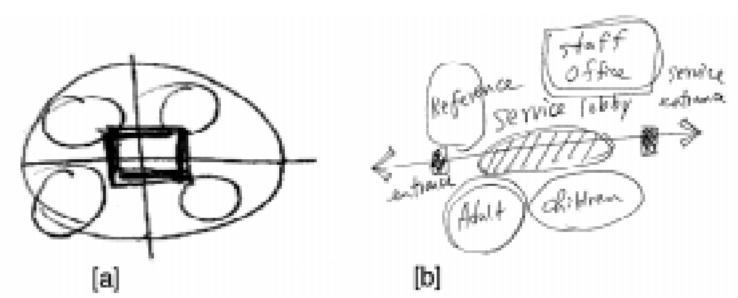


Figure A-1. Designer's basic scheme for spatial arrangement; [a] minimal bubble diagram; [b] developed version with functions labeled and some size and positional information.

Now she wants to arrange the elements of her scheme in an interesting way so that the building's shape will recall the natural history theme. To find appropriate and compelling shapes she flips through her journal to find forms similar to her current drawing. The objective of her search is to find appropriate shapes that can accommodate both the functions of the library program and the spatial arrangements she has already developed. Believing that form should follow function, she searches for shapes that are similar to her drawing, to guide the translation of bubbles into a building form. She decides that if she finds any shape interesting or similar in shape to her bubble diagram, she will draw it on her tracing paper and make notes so later she can later use those drawings for her current design project or include them in her journal for future use. She also decides that the shapes should come from the design theme—natural history. So she browses her journal to find shapes of natural objects. Some notes in her journal point her to books on plants and animals. She copies what she finds in her journal and books (Figure A-2) to her drawing as a reference for design (Figure A-1).

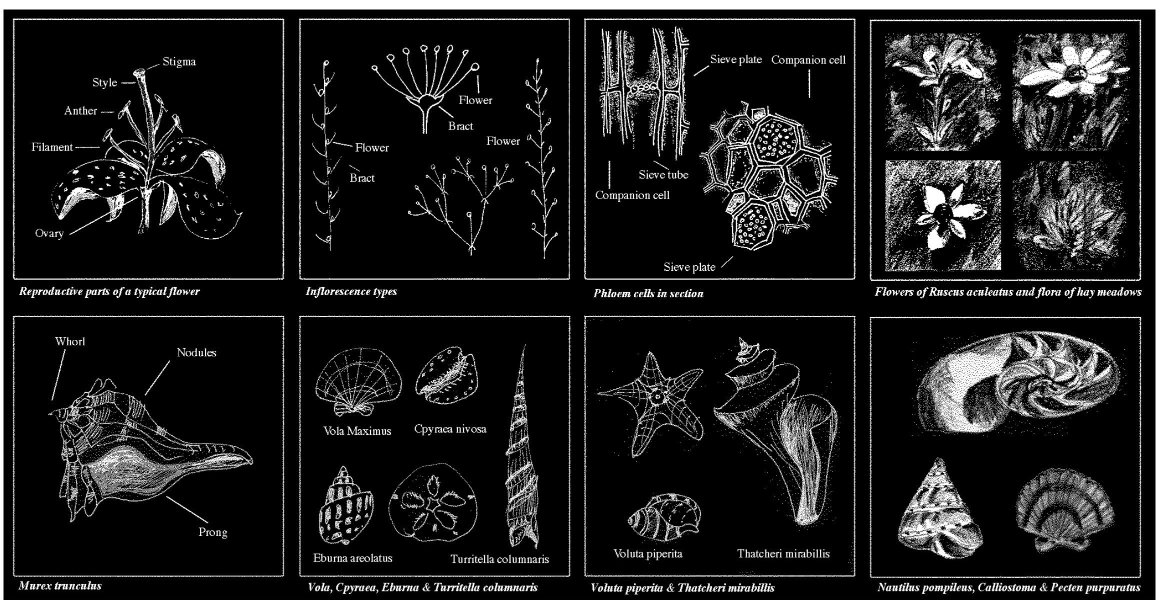


Figure A-2. Forms found in a handbook of the natural world. (Drawings by author, after (Gibbons 1984), and (Webb 1948; Alexander, Ishikawa et al. 1977))

After finding an interesting form in her journal, she makes a diagram to record her thoughts about the object’s shape (Figure A-3b). She might also look in her books for the original picture (Figure A-3c). After finding the photograph she flips through the rest of the book looking for other interesting forms (Figure A-2). For example, she might copy a few plant and shell shapes to her tracing paper and add annotations on describing how she might use the shapes in her design. She considers using flower petals or round seashell forms for her floor plan because both have organize shapes surrounding around a central object. Or she might find an arrangement of plant forms suitable for her atrium column and skylight structure (flower stalks rising from a central point of the stem end to create an umbrella effect), or a fan- shaped sea shell for her arched window (Figure A-3e).

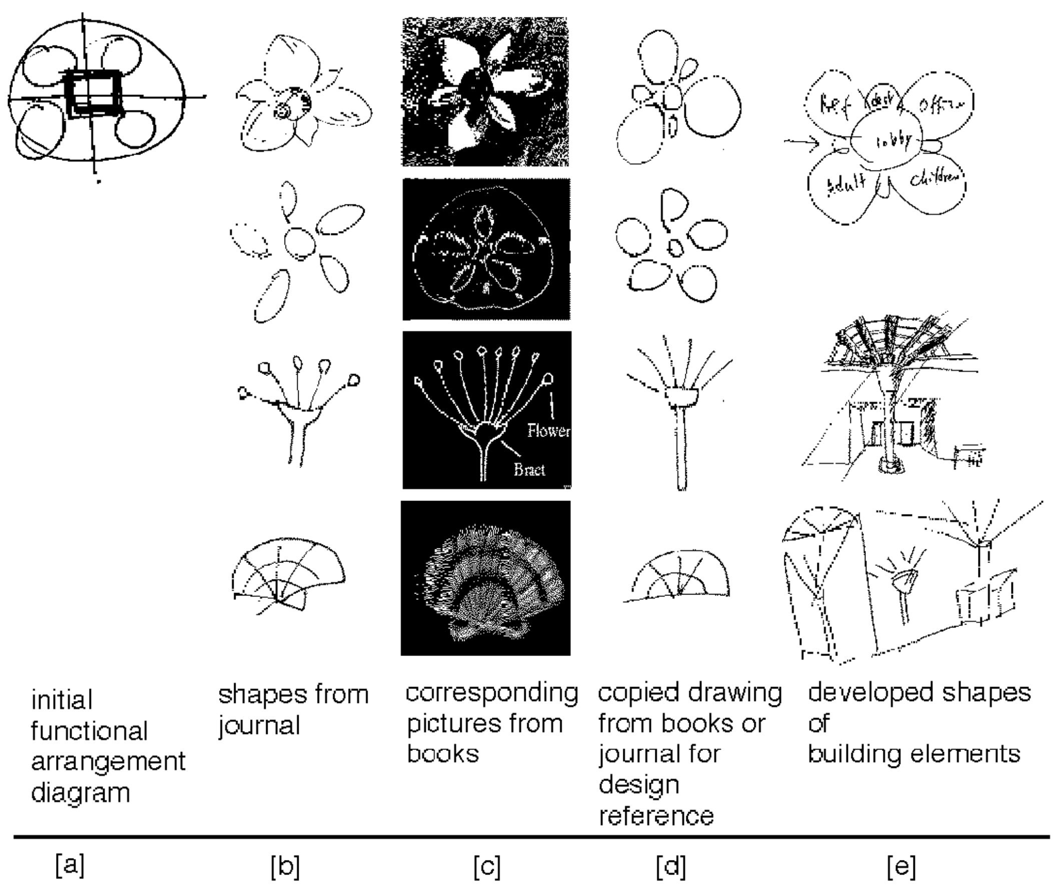


Figure A-3. Graphic representations used in design development.

Our designer's journal also contains magazine clippings and drawings of architectural precedents—buildings she has found interesting in the past. While browsing her journal, perhaps she finds a building whose form seems relevant (it may be a library, a church, or an office building) and she uses tracing paper to copy it into her design drawing. Then she might ask herself, 'How about extending this form to fit the children's reading room?' She gradually traces over the drawing, adjusting the shape to fit the functional program (Figure A-4).

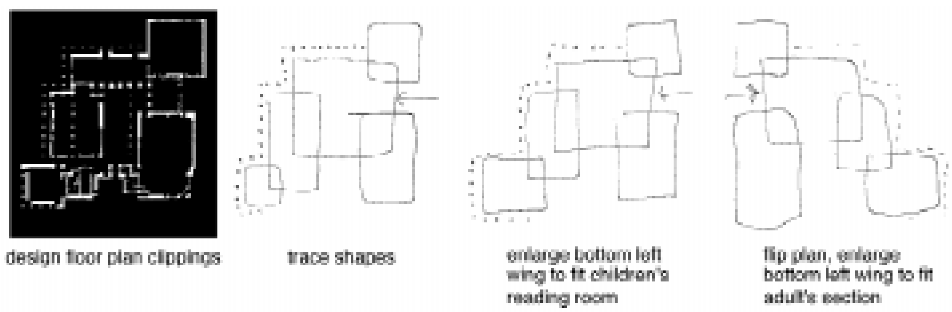


Figure A-4. Tracing a magazine clipping into the design drawing (design clipping by author after Kallmann, McKinnell & Wood's American Academy of Arts & Sciences, (Mackintosh 1977)).

After fitting the children's reading room into the form, she decides to flip the drawing so the entrance will be from the west side as in her functional arrangement diagram (Figure A-1), and she adjusts the drawing again to fit the adult's section. Then she combines her previous flower petal concept (intervals of large and small petals; Figure A-1) with the modified flipped floor plan. She finishes her schematic design of the building shape, arriving finally at the schematic design shown in Figure A-5.

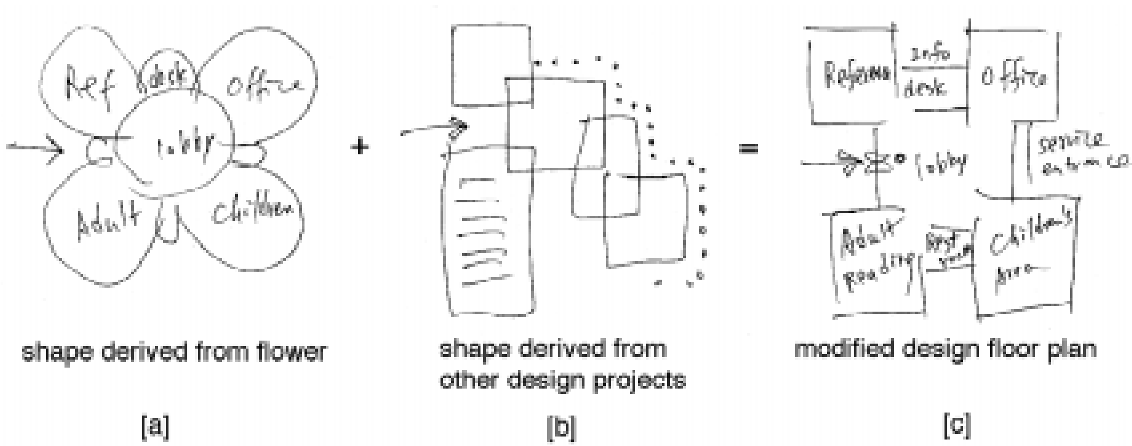


Figure A-5. Combining ideas and stretching the drawing to suit the design.

A.2. REMARKS ON THE SCENARIO

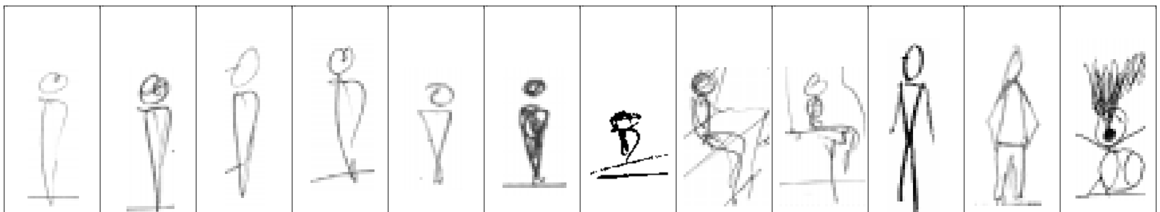
In the scenario describe above, the designer first moves fluidly between different graphic media, which include the drawing board, where the actual design takes shape; a journal, or sketchbook where she stores interesting fragments of designs; tracing paper, by which she transfers items into the journal or into the design; and a collection of books, magazines, and design folios, in which she searches for and copies interesting reference images. The designer knows what task she is doing and what kinds of tools she wants.

Second, the designer uses various operations. She browses her journal and her collection of books. She selects interesting items from them and traces and copies them into her design. She adapts the forms she finds to her design, using one part here and another there, stretching, rotating, and moving her traced references in order to make them ‘work’ with the design. And she finds ways to map her functional library program with the reference forms she adopts.

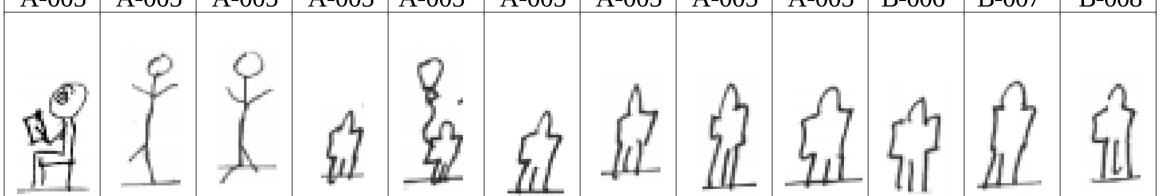
Third, the designer explicitly chooses explicitly to adopt a shape for her building associated that is associated with a concept derived from the architectural program—the natural history theme. It is common that designers borrow forms that are associated quite literally with the building concept.

APPENDIX B

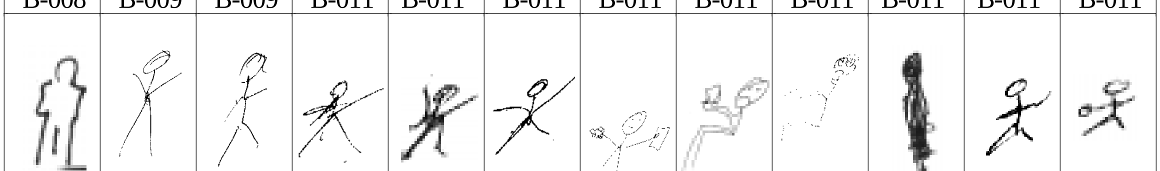
SYMBOLS FOR HUMAN FIGURES FROM THE DIAGRAMMING EXPERIMENT



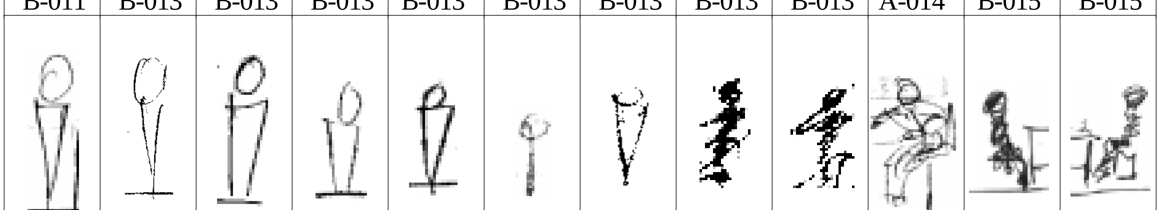
A-005	A-005	A-005	A-005	A-005	A-005	A-005	A-005	A-005	B-006	B-007	B-008



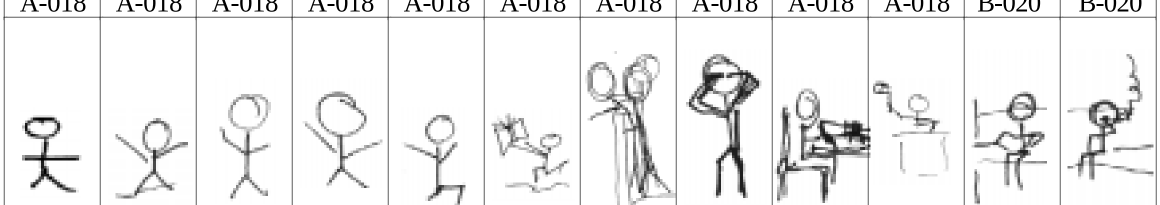
B-008	B-009	B-009	B-011	B-011	B-011	B-011	B-011	B-011	B-011	B-011	B-011



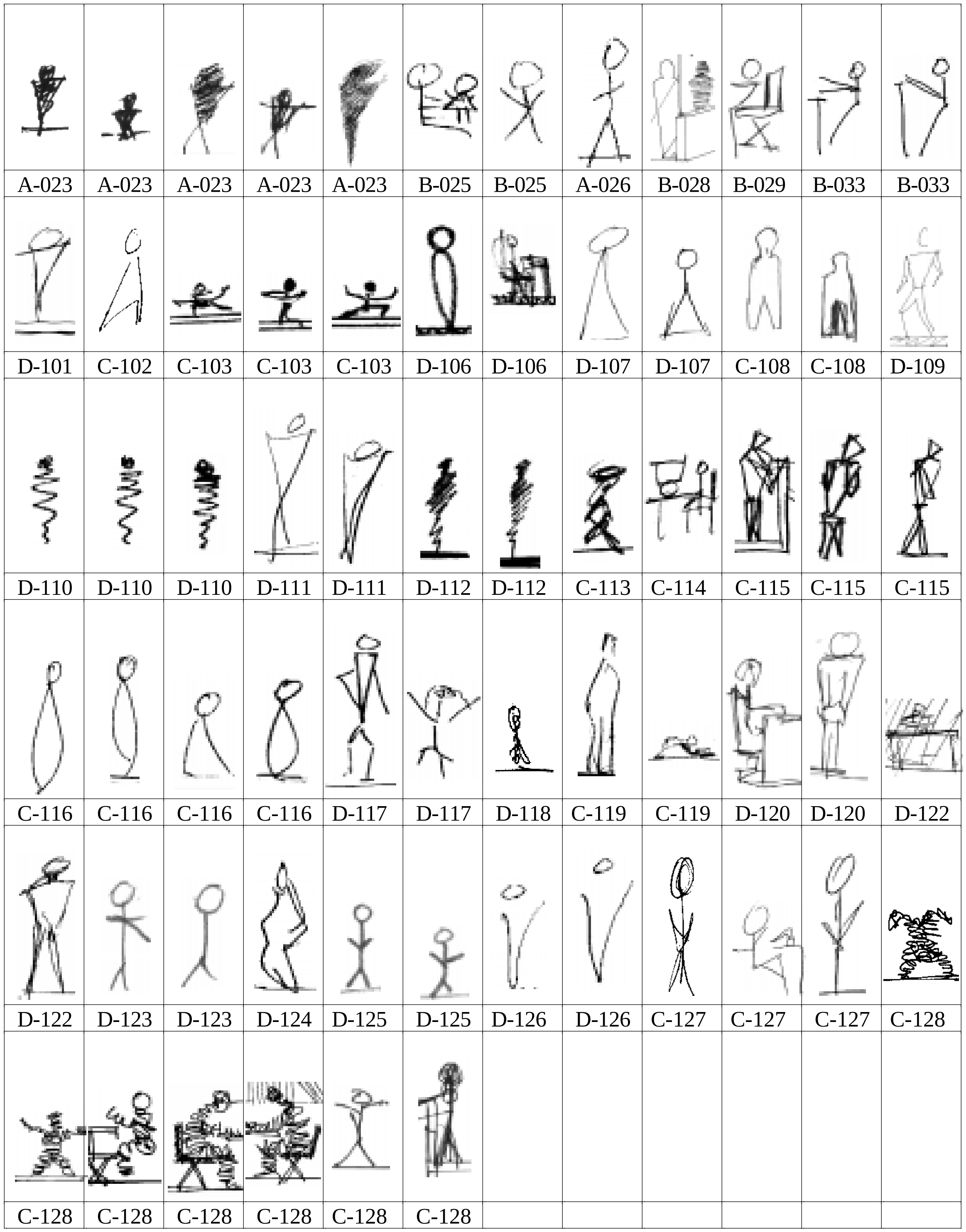
B-011	B-013	B-013	B-013	B-013	B-013	B-013	B-013	B-013	A-014	B-015	B-015



A-018	A-018	A-018	A-018	A-018	A-018	A-018	A-018	A-018	A-018	B-020	B-020



A-022	A-022	A-022	A-022	A-022	A-022	A-022	A-022	A-022	A-022	A-022	A-022



* A-005 means the participating designer belongs to group A with ID 005.

APPENDIX C

DESIGN EXPERIMENT QUESTIONNAIRE

Consent and Release Form

Study of Design Drawing

I understand the purpose of this experiment is to study design drawing. The experiment will take about half an hour.

I understand I will be given a design brief to make conceptual designs. I understand I will be asked to "think out loud" as I design.

I understand that the study involves research and that I may make inquiries concerning this procedure.

I understand that my participation in the research is completely voluntary and I am free to withdraw my consent at any time and to discontinue participation in the project.

I have read and understood the procedures involved in the research and hereby consent to volunteer to participate in this study.

____

Date

______________________________ ______________________________

Participant's Signature Investigator's Signature

______________________________ ______________________________

Participant's Name Investigator's Name

Please tell us about yourself.

Participant ID: ___________________

Participant Name: ______________________ Age: _________ Gender: ______

Occupation (Job Title): _________________________

1. Briefly describe your architecture design experience:

2. How many years of education have you completed?

College (# of years) ________ Graduate School (# of years) ______________

3. How often do you use freehand drawing for architectural design?

Never |——|——|——|——| All the time

4. How often do you use freehand drawing to communicate with another designer?

Never |——|——|——|——| All the time

Design Brief

An architecture firm just rented a one story warehouse to be its new office space. The space is 70' ft x 25', running East-West the long way. The west side entrance faces a main street. All sides of the space may have openings except the south side, which connects to another building. This firm currently has 1 chief architect, 3 designers, 3 CAD operators, 2 contract draftsmen, a secretary and about 1-2 student interns.

The office will be designed to have, besides the area for work groups of designers, CAD specialists and draftsperson, a meeting room, a small kitchenette, a bathroom, and a chief architect's private office, a secretary - receptionist - general affairs section, storage space, printing and plotting area, and space for student interns.

Flexible functional combinations may suit the occupants' various needs. For example, a meeting area can also serve to display past design works, a storage space may include a small reference library, or product catalogue and material samples, and the print room could include Xerox, blueprint and computer plotting machines, etc.

Task 1

Please take no more than 10 minutes to make a conceptual, schematic design for the above program. Please pay particular attention on zoning arrangements. For example, consider where to put the lobby, chief architect's office, meeting room, and the different work group spaces, etc.

Please continue to the following pages. Thank you!

Task 2

Please start with a new sheet of tracing paper. This time, please pay particular attention to lighting issues. For example, where should the window be on the wall? And, perhaps introduce skylight into the office if plausible?

Please take no more than 5 minutes to complete this task. Thank you!

Task 3

Please start with a new sheet of tracing paper. This time, please focus on the visibility and privacy concerns between different spaces.

For example, consider making each work group space commands a fine view to the exterior, and an easy access to the meeting area.

Please take no more than 5 minutes to complete this task. Thank you!

Task 4

Please start with a new sheet of tracing paper. This time, please make sure your meeting room is large enough to accomodate the chief architect's favorite meeting table (4 ft x 10 ft). Also, please make sure the designers' work space would have at least 800 sq. ft.

Please take no more than 5 minutes to complete this task.

This is the last task. Thank you! Please turn to next page and give us your comments.

Please tell us how you feel about this experiment.

Participant ID: ____________________

1. How clearly were the tasks defined?

Task 1 Unclearly |----|----|----|----| Clearly

Task 2 Unclearly |----|----|----|----| Clearly

Task 3 Unclearly |----|----|----|----| Clearly

Task 4 Unclearly |----|----|----|----| Clearly

2. How difficult to perform the tasks?

Task 1 Difficult |----|----|----|----| Easy

Task 2 Difficult |----|----|----|----| Easy

Task 3 Difficult |----|----|----|----| Easy

Task 4 Difficult |----|----|----|----| Easy

3. How often do you use a computer for architecturally related work? What do you use?

Never |----|----|----|----| All the time

4. Do you think you have your personalized symbols and short hands for designing? What are they?

5. Would you like a computer program to have freehand sketching abilities? What should it do?

6. Suggestions and comments for the experiment? Thank you!

APPENDIX D

ROGER'S DESIGN SESSION

Please tell us about yourself.

Participant ID: ________________

Participant Name: ____________________ Age: 21 Gender: M

Occupation (Job Title): STUDENT

1. Briefly describe your architecture design experience:

3 INTERNSHIPS IN ARCH. DESIGN FIRMS
3 YEARS OF ENV. DESIGN SCHOOL

2. How many years of education have you completed?

College (# of years) 3 Graduate School (# of years) ____________

3. How often do you use freehand drawing for architectural design?

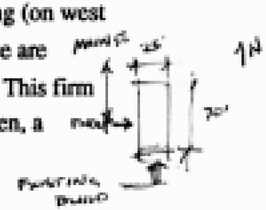
Never |——|——|——|——| All the time

4. How often do you use freehand drawing to communicate with another designer?

Never |——|——|——|——| All the time

Design Brief

An architecture firm just rented a one story warehouse to be used as their new office space. The dimension of the space is 70 ft wide (on north and south sides) and 25 ft long (on west and east sides) with west side entrance facing a main street. All sides of the space are allowed to have opening except the south side that connects to another building. This firm currently has 1 chief architect, 3 designers, 3 CAD operators, 2 contract draftsmen, a secretary and about 1-2 student interns.

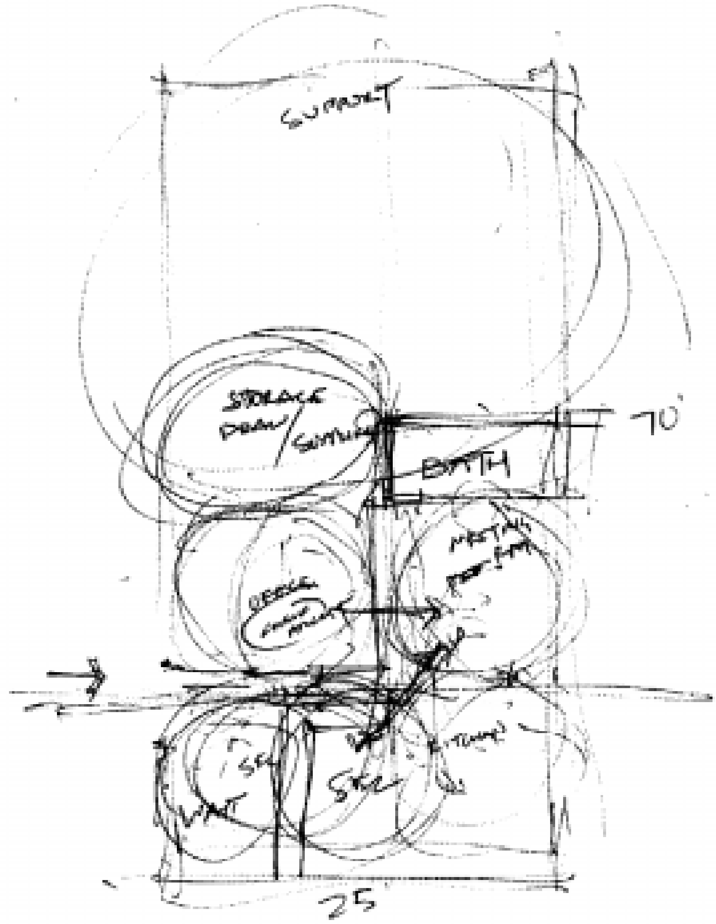


The office will be designed to have, besides the area for work groups of designers, CAD specialists and draftspersons, a meeting room, a small kitchenette, a bathroom, and a chief architect's private office, a secretary - receptionist - general affairs section, storage space, printing and plotting area, and space for student interns.

Flexible functional combinations may suit the occupants' various needs. For example, a meeting area can also serve as displaying of past design works, a storage space may include a small reference library, or product catalogue and material samples, print room could include Xerox, blueprint and computer plotting machines, etc.

Task 1

Please take no more than 10 minutes to make a conceptual, schematic design for the above program. Then continue on the following questions on the next page. Thank you!



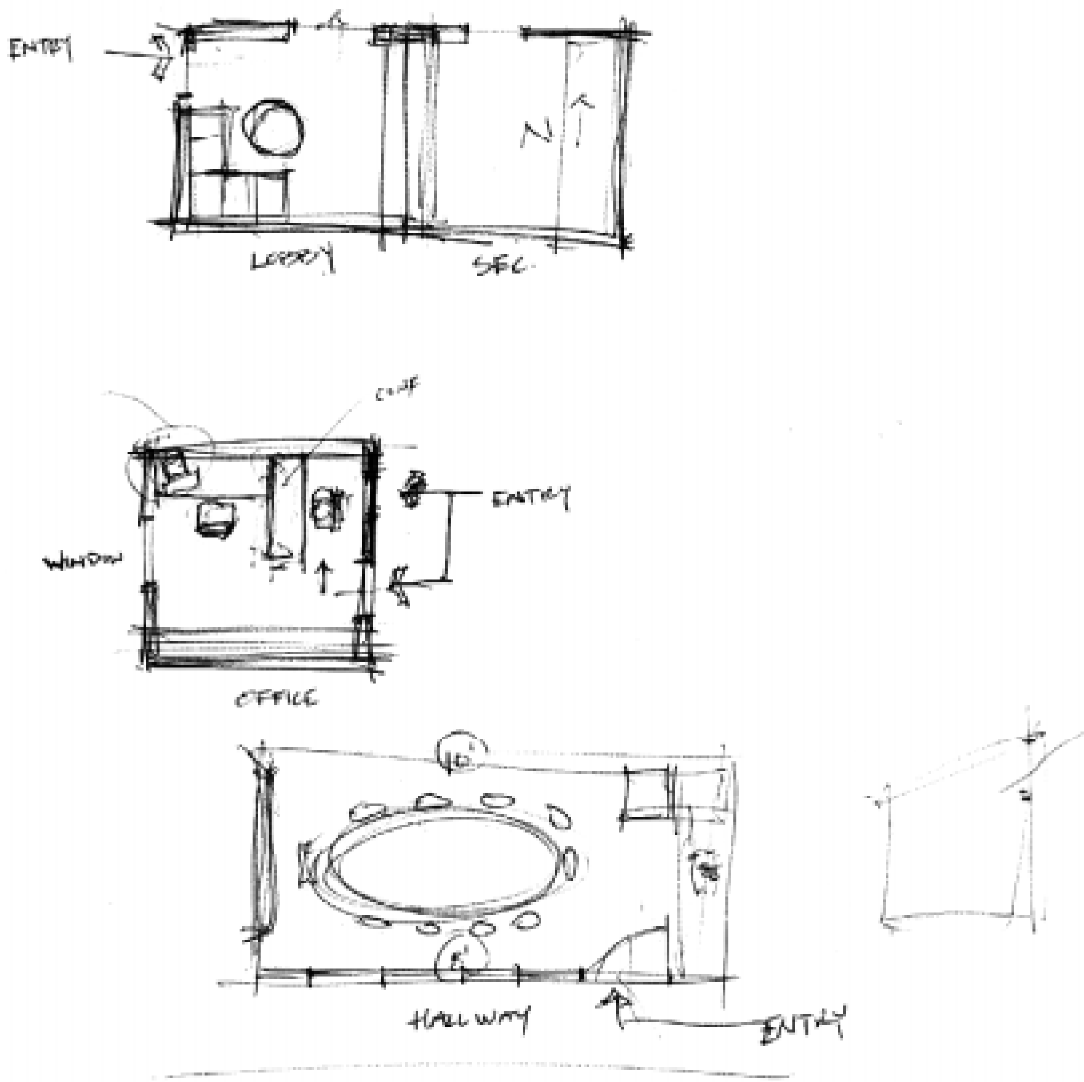
Task 1 6 min

Task 2

Please start with a new sheet of paper to do a design for the brief. This time, please focus on the zoning of the spatial arrangement for the office.

For example, consider how to arrange a lobby, meeting area, different work groups, service area, etc.

Please take no more than 5 minutes to complete this task. Thank you!



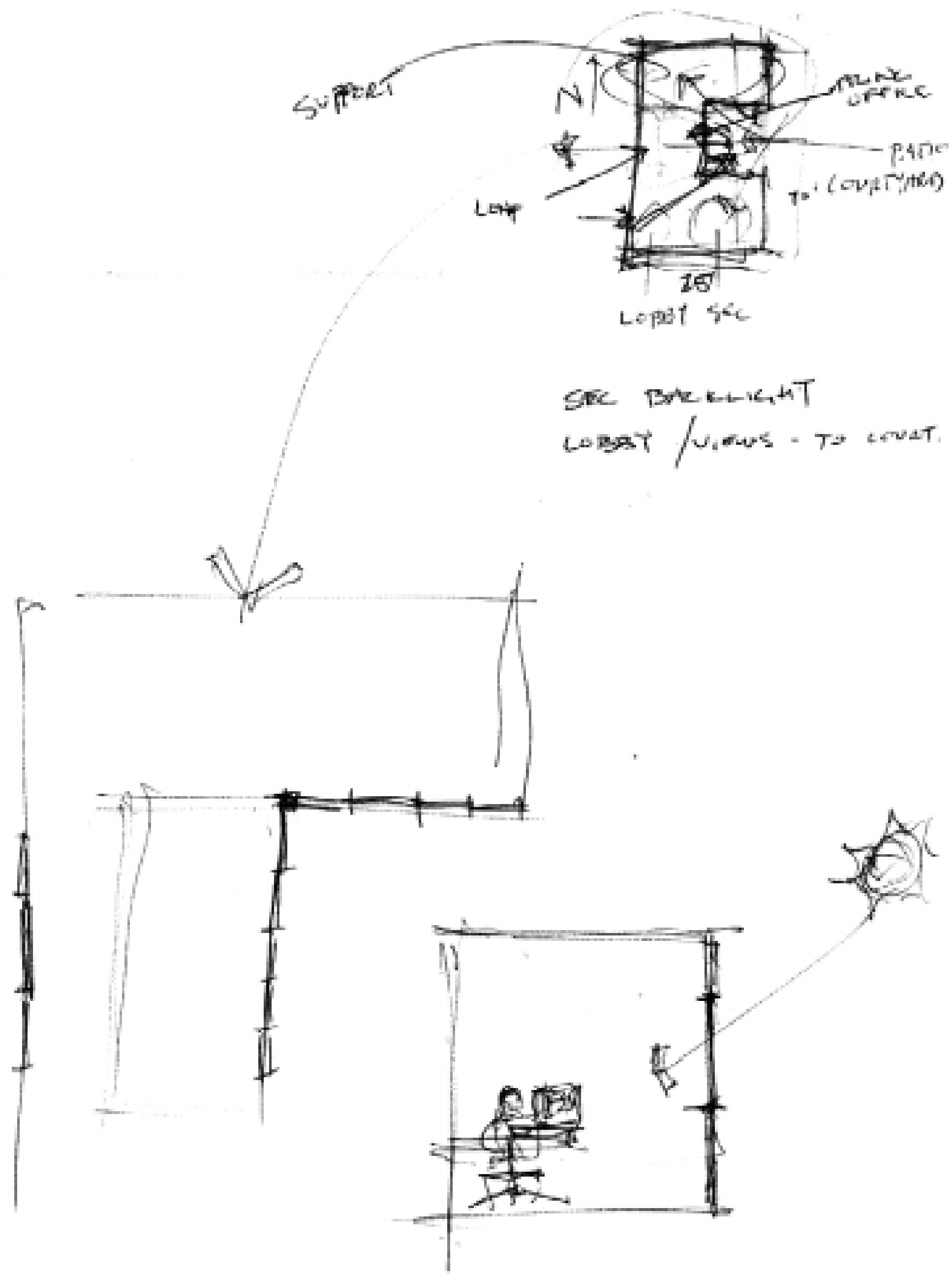
Task 2

Task 3

Please start with a new sheet of paper to do a design for the brief. This time, please pay particular attention to lighting issue about the meeting and working area.

For example, perhaps introduce skylight or a small courtyard into the office if plausible?

Please take no more than 5 minutes to complete this task. Thank you!


SUPPORT
N
25'
LOBBY SEC
SEC BACKLIGHT
LOBBY / VIEWS - TO COURT.

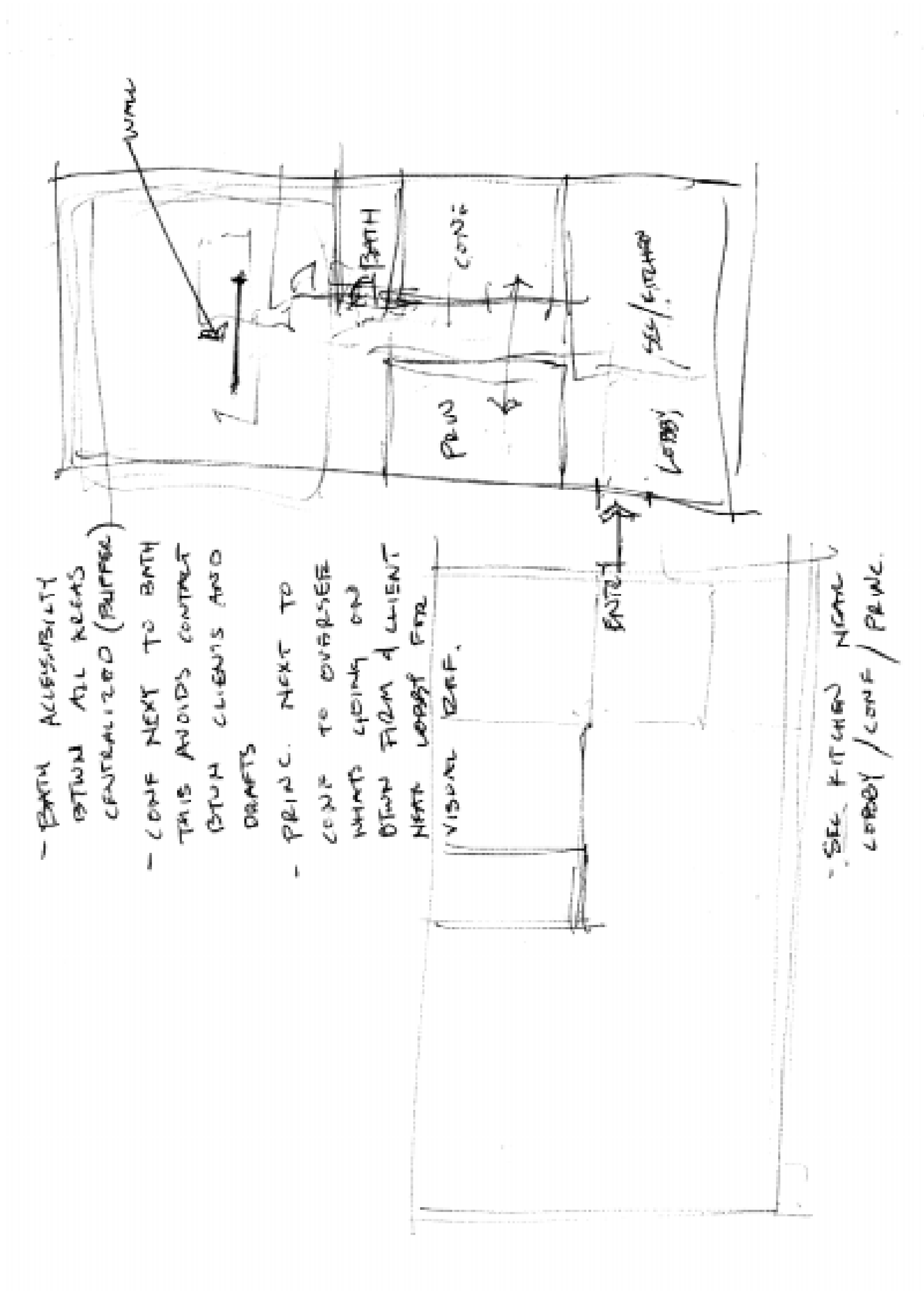
Task 4

Please start with a new sheet of paper to do a design for the brief. This time, please focus on the visibility, noise and privacy concerns between the receptionist, chief architects and the work groups.

For example, consider making each work group commands a fine view to the exterior, and an easy access to the meeting area.

Please take no more than 5 minutes to complete this task. Thank you!

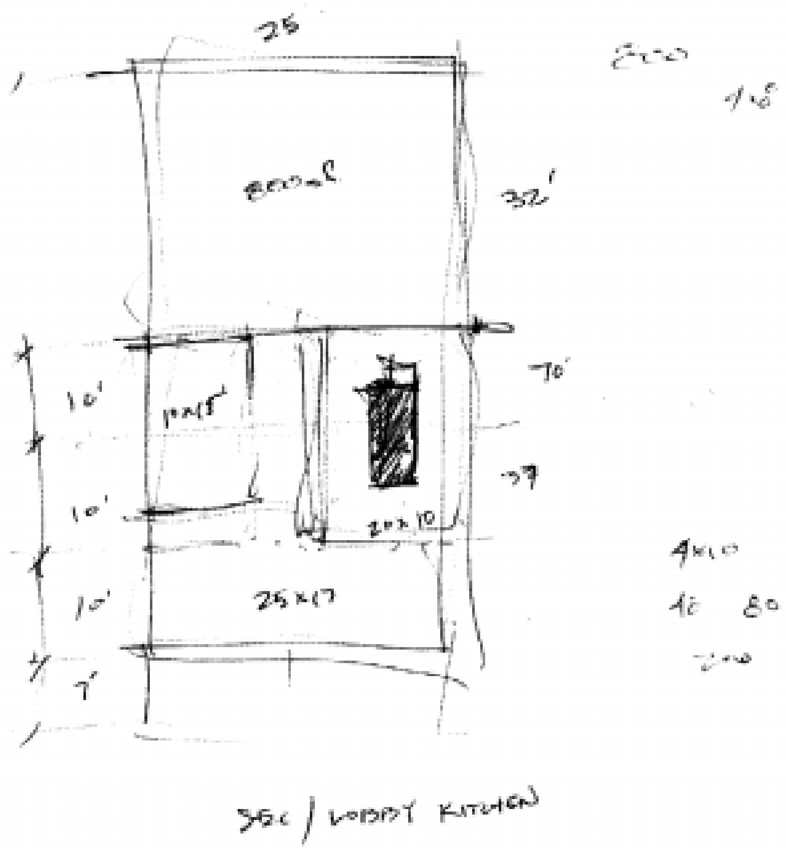
SEE TASK #1


- BATH ACCESSIBILITY BTWN ALL AREAS CENTRALIZED (BUFFER)
- CONF NEXT TO BATH THIS AVOIDS CONTACT BTWN CLIENTS AND DRAFTS
- PRINC. NEXT TO CONF TO OVERSEE WHATS GOING ON BTWN FIRM & CLIENT NEAR LOBBY FOR VISUAL REF.
- SEC KITCHEN NEAR LOBBY / CONF / PRINC.
WALL
BATH
CONF
PRIN
SEC/KITCHEN
LOBBY
ENTRY

Task 5

Please start with a new sheet of paper to do a design for the brief. This time, you are given the condition that the chief architect has a favorite oak meeting table with dimension 4 ft * 10 ft, and he would like to put that table in the meeting space.
After you consider how the table can fit into the meeting room, please make sure each designer's work space would have at least 800 sq. ft.

Please take no more than 5 minutes to complete this task. Thank you!


25
32'
10'
10'
10'
7'
25x17
20x10
4x10
80
LOBBY KITCHEN

Please tell us how you feel about this experiment.

Participant ID:__________________

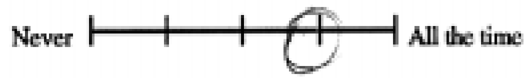
1. How clearly were the tasks defined?

Task 1	Unclearly	├───┼───┼───┼───┤	Clearly
Task 2	Unclearly	├───┼───┼───┼───┤	Clearly
Task 3	Unclearly	├───┼───┼───┼───┤	Clearly
Task 4	Unclearly	├───┼───┼───┼───┤	Clearly
Task 5	Unclearly	├───┼───┼───┼───┤	Clearly ← STRANGE QUESTION

2. How difficult to perform the tasks?

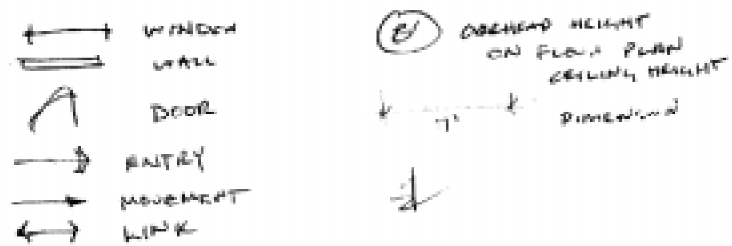
Task 1	Difficult	├───┼───┼───┼───┤	Easy
Task 2	Difficult	├───┼───┼───┼───┤	Easy
Task 3	Difficult	├───┼───┼───┼───┤	Easy
Task 4	Difficult	├───┼───┼───┼───┤	Easy
Task 5	Difficult	├───┼───┼───┼───┤	Easy

3. How often do you use a computer for architecturally related work? What do you use?



4. Do you think you have your personalized symbols and short hands for designing? What are they?

YES



5. Would you like a computer program to have freehand sketching abilities? What should it do?

YES, I WANT TO SEE WHAT I DRAW ON THE PAD I'M DRAWING ON
SAVE ~~##~~ OLD SKETCHES FOR REFERENCE AND COME BACK TO THEM AND DO OVERLAYS AND SKETCHES

6. Suggestions and comments for the experiment? Thank you!

THINK ABOUT CHANGING PROGRAM. AN ARCH. OFFICE MIGHT BE TO EASY FOR ARCH. WHO HAVE ALREADY WORKED IN ONE AND KNOW WHAT IS GOOD AND BAD.

APPENDIX E

NOI'S DESIGN SESSION

Please tell us about yourself.

Participant ID: 1527

Participant Name: ____________ Age: 20 Gender: MALE

Occupation (Job Title): STUDENT

1. Briefly describe your architecture design experience:

STILL IN SCHOOL

VIRICH ARCHITECTS

2. How many years of education have you completed?

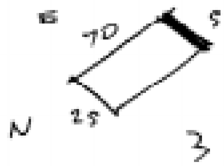
College (# of years) 4 Graduate School (# of years) ____________

3. How often do you use freehand drawing for architectural design?

Never |——|——|——|——| All the time

4. How often do you use freehand drawing to communicate with another designer?

Never |——|——|——|——| All the time



Design Brief

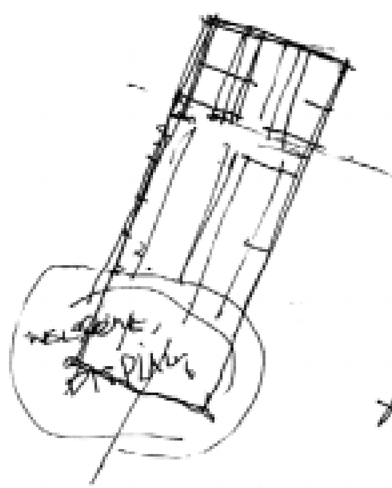
An architecture firm just rented a one story warehouse to be used as their new office space. The dimension of the space is 70 ft wide (on north and south sides) and 25 ft long (on west and east sides) with west side entrance facing a main street. All sides of the space are allowed to have opening except the south side that connects to another building. This firm currently has 1 chief architect, 3 designers, 3 CAD operators, 2 contract draftsmen, a secretary and about 1-2 student interns.

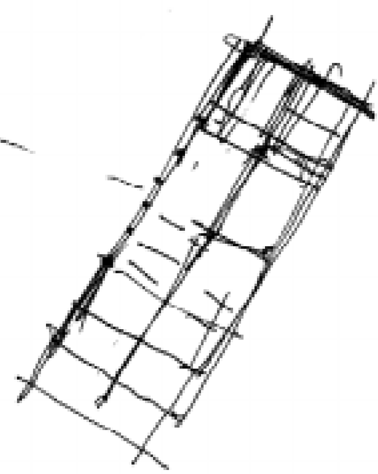
The office will be designed to have, besides the area for work groups of designers, CAD specialists and draftsperson, a meeting room, a small kitchenette, a bathroom, and a chief architect's ~~private office~~, a secretary/receptionist, general affairs section, storage space, printing and plotting area, and space for student interns.

Flexible functional combinations may suit the occupants' various needs. For example, a meeting area ~~can also serve as~~ displaying of past design works, a storage space may include a small reference library, or ~~product catalogue and~~ material samples, print room could include Xerox, blueprint and computer plotting machines, etc.

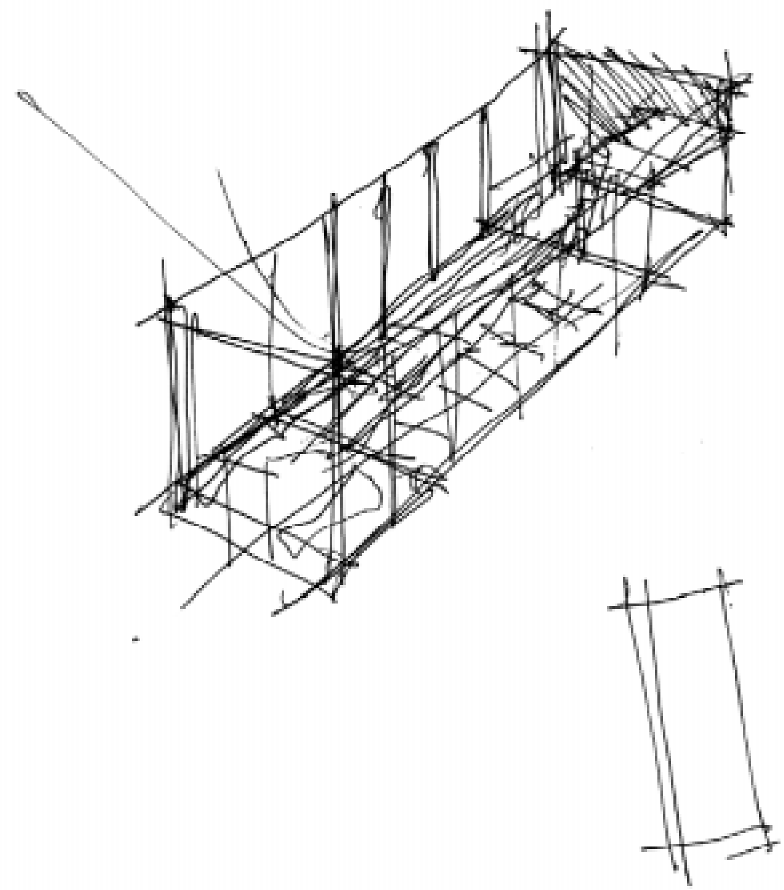
Task 1

Please take no more than 10 minutes to make a conceptual, schematic design for the above program. Then continue on the following questions on the next page. Thank you!





3



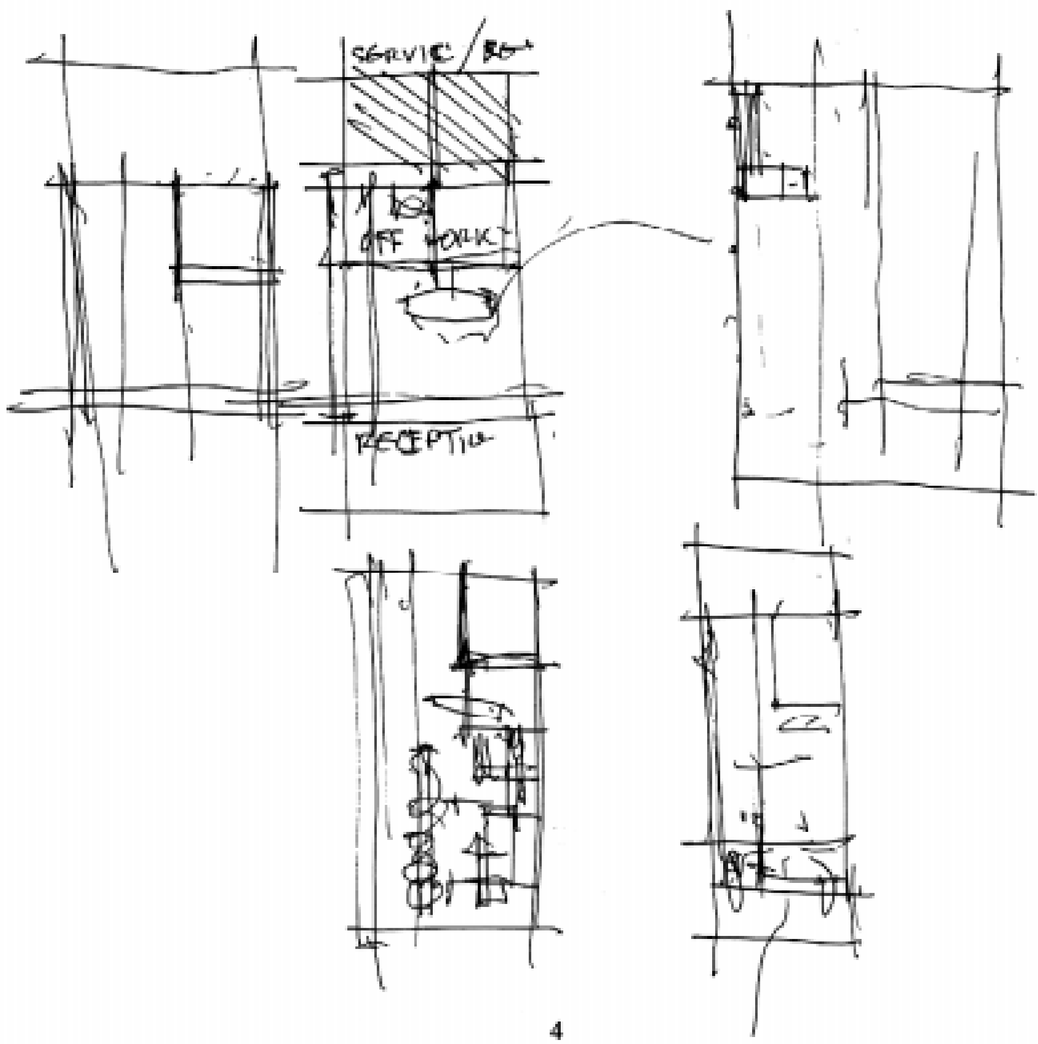
This was drawn on the back of the program page (previous page).

Task 2

Please start with a new sheet of paper to do a design for the brief. This time, please focus on the zoning of the spatial arrangement for the office.

For example, consider how to arrange a lobby, meeting area, different work groups, service area, etc.

Please take no more than 5 minutes to complete this task. Thank you!

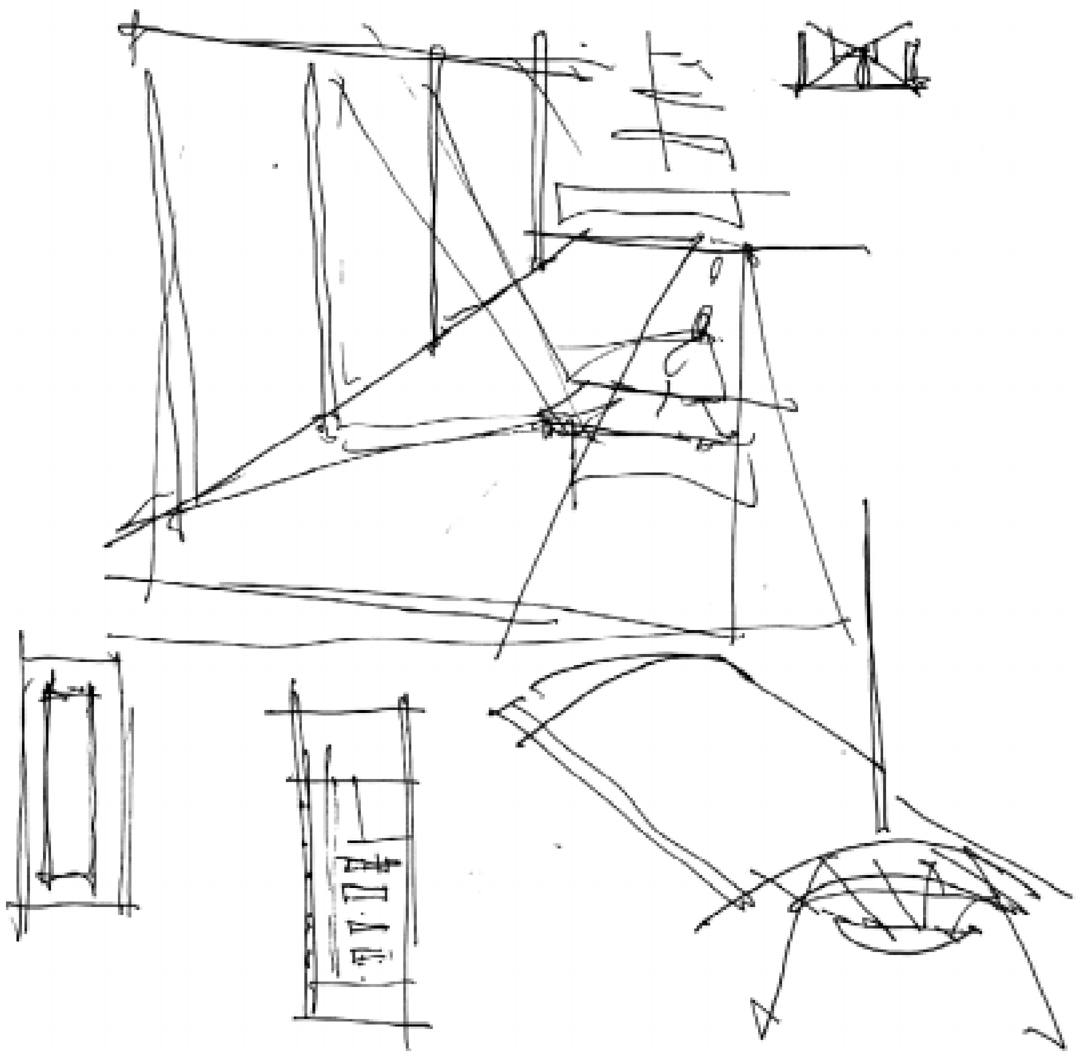


Task 3

Please start with a new sheet of paper to do a design for the brief. This time, please pay particular attention to lighting issue about the meeting and working area.

For example, perhaps introduce skylight or a small courtyard into the office if plausible?

Please take no more than 5 minutes to complete this task. Thank you!

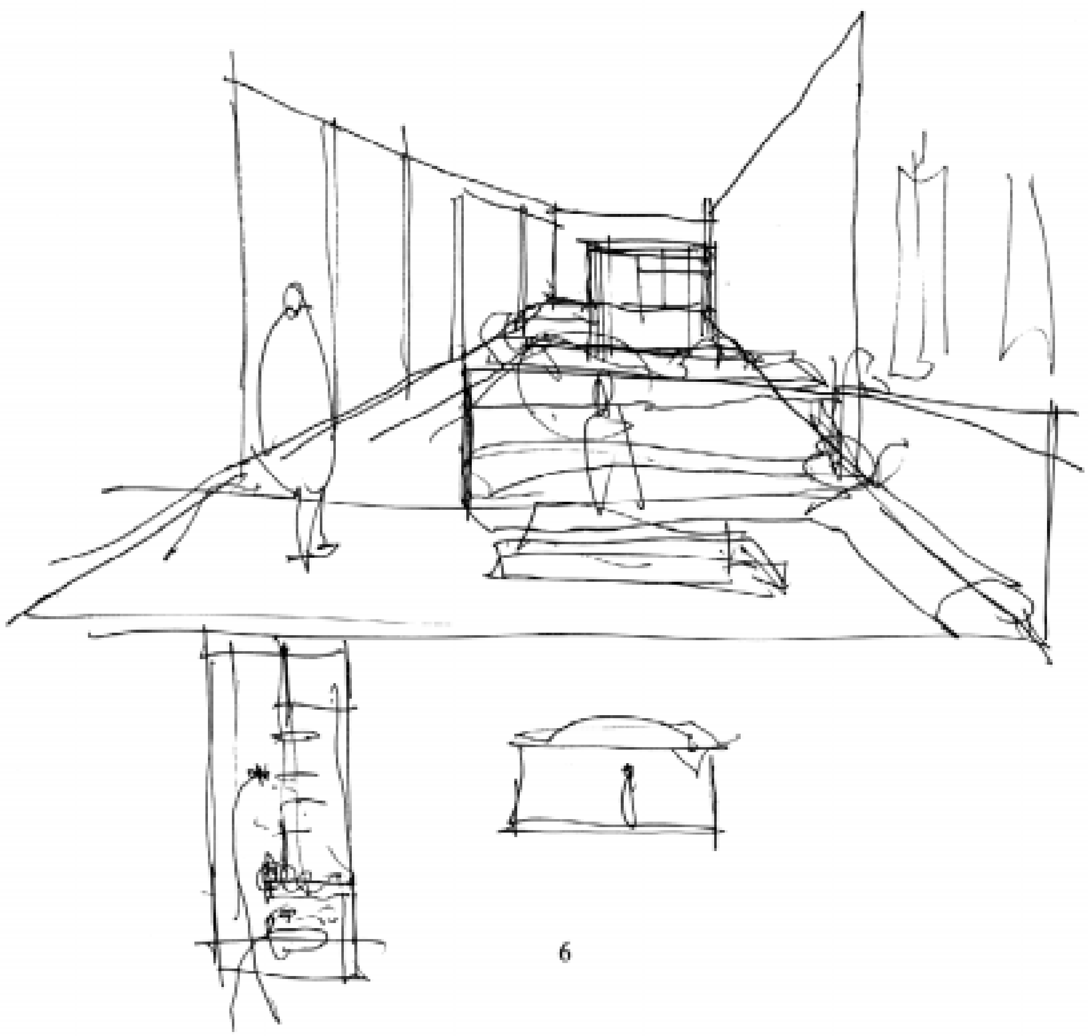


Task 4

Please start with a new sheet of paper to do a design for the brief. This time, please focus on the visibility, noise and privacy concerns between the receptionist, chief architects and the work groups.

For example, consider making each work group commands a fine view to the exterior, and an easy access to the meeting area.

Please take no more than 5 minutes to complete this task. Thank you!

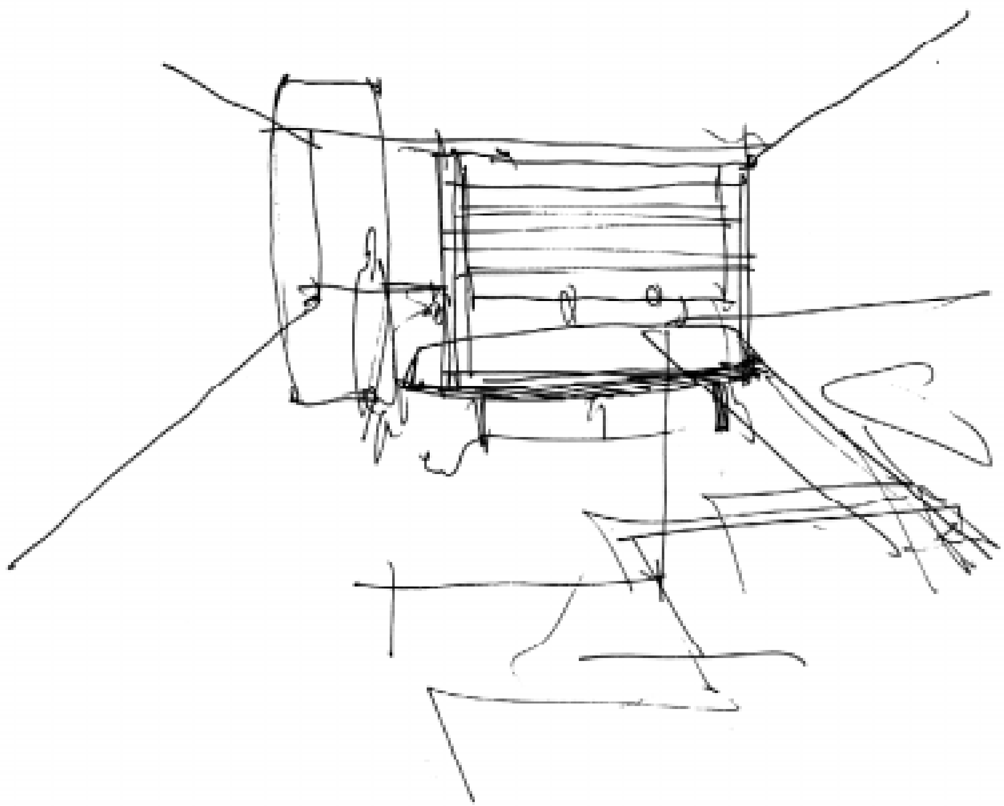


Task 5

Please start with a new sheet of paper to do a design for the brief. This time, you are given the condition that the chief architect has a favorite oak meeting table with dimension 4 ft * 10 ft, and he would like to put that table in the meeting space.

After you consider how the table can fit into the meeting room, please make sure each designer's work space would have at least 800 sq. ft.

Please take no more than 5 minutes to complete this task. Thank you!



Please tell us how you feel about this experiment.

Participant ID: 1527

1. How clearly were the tasks defined?

Task 1 Unclearly Clearly

Task 2 Unclearly Clearly

Task 3 Unclearly Clearly

Task 4 Unclearly Clearly

Task 5 Unclearly Clearly

2. How difficult to perform the tasks?

Task 1 Difficult Easy

Task 2 Difficult Easy

Task 3 Difficult Easy

Task 4 Difficult Easy

Task 5 Difficult Easy

3. How often do you use a computer for architecturally related work? What do you use?

Never |—|—|—|—| All the time

4. Do you think you have your personalized symbols and short hands for designing? What are they?

YES, CIRCLE REPRESENT TREE, PLANT.

→ HUMAN.

5. Would you like a computer program to have freehand sketching abilities? What should it do?

YES, IT SHOULD HAS ALL ABILITY TO SKETCH, BUT ALSO HAS ABILITY TO UNDO OR OVERLAY.

AND. IT SHOULD ACT LIKE A REAL PENCIL

6. Suggestions and comments for the experiment? Thank you!

It's very flow. AND. THE TASK CAN BE MORE ORGANIZED. THANK YOU.

APPENDIX F

SAMUEL'S DESIGN SESSION

Please tell us about yourself.

Participant ID: ______________

Participant Name: ______________ Age: 28 Gender: M

Occupation (Job Title): Design Consultant / Instructor

1. Briefly describe your architecture design experience:

⑤ Vn ① Worked as draftsman for Allen Harlow 1½ years
② Worked at communication Arts: schematic design shopping malls and retail spaces
③ Design a 4800 sq. ft. house on Coal Creek
④ Worked with Architect E. I. Made modeling houses in 3D

2. How many years of education have you completed?

College (# of years) ~~7~~ 6 Graduate School (# of years) ______________

degree in ENVD: architecture

3. How often do you use freehand drawing for architectural design?

Never |—|—|—|—| All the time

4. How often do you use freehand drawing to communicate with another designer?

Never |—|—|—|—| All the time

Design Brief

An architecture firm just rented a one story warehouse to be its new office space. The space is 70' ft x 25', running East-West the long way. The west side entrance faces a main street. All sides of the space may have openings except the south side, which connects to another building. This firm currently has 1 chief architect, 3 designers, 3 CAD operators, 2 contract draftsmen, a secretary and about 1-2 student interns.

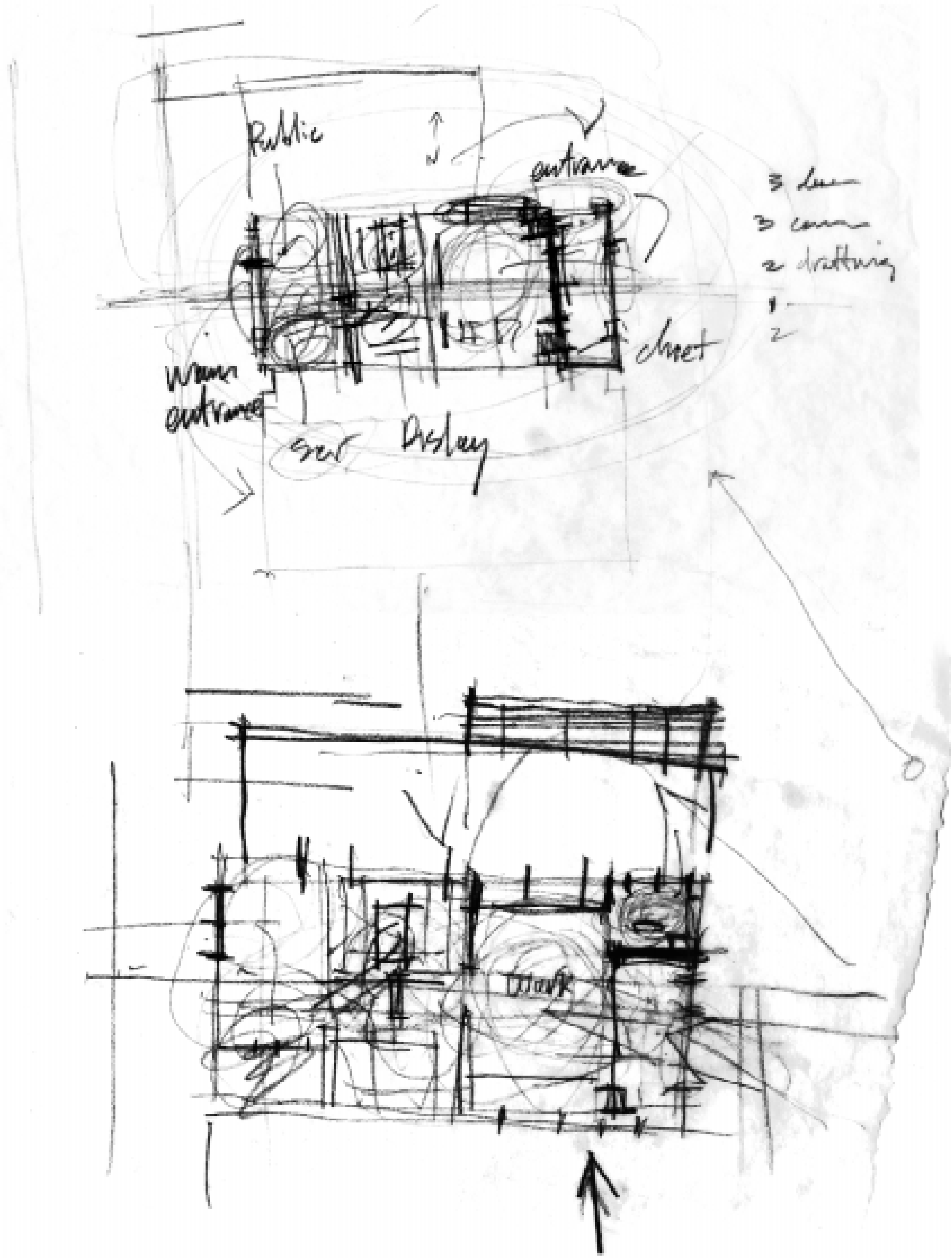
The office will be designed to have, besides the area for work groups of designers, CAD specialists and draftsperson, a meeting room, a small kitchenette, a bathroom, and a chief architect's private office, a secretary - receptionist - general affairs section, storage space, printing and plotting area, and space for student interns.

Flexible functional combinations may suit the occupants' various needs. For example, a meeting area can also serve to display past design works, a storage space may include a small reference library, or product catalogue and material samples, and the print room could include Xerox, blueprint and computer plotting machines, etc.

Task 1

Please take no more than 10 minutes to make a conceptual, schematic design for the above program. Please pay particular attention on zoning arrangements. For example, consider where to put the lobby, chief architect's office, meeting room, and the different work group spaces, etc.

Please continue to the following pages. Thank you!

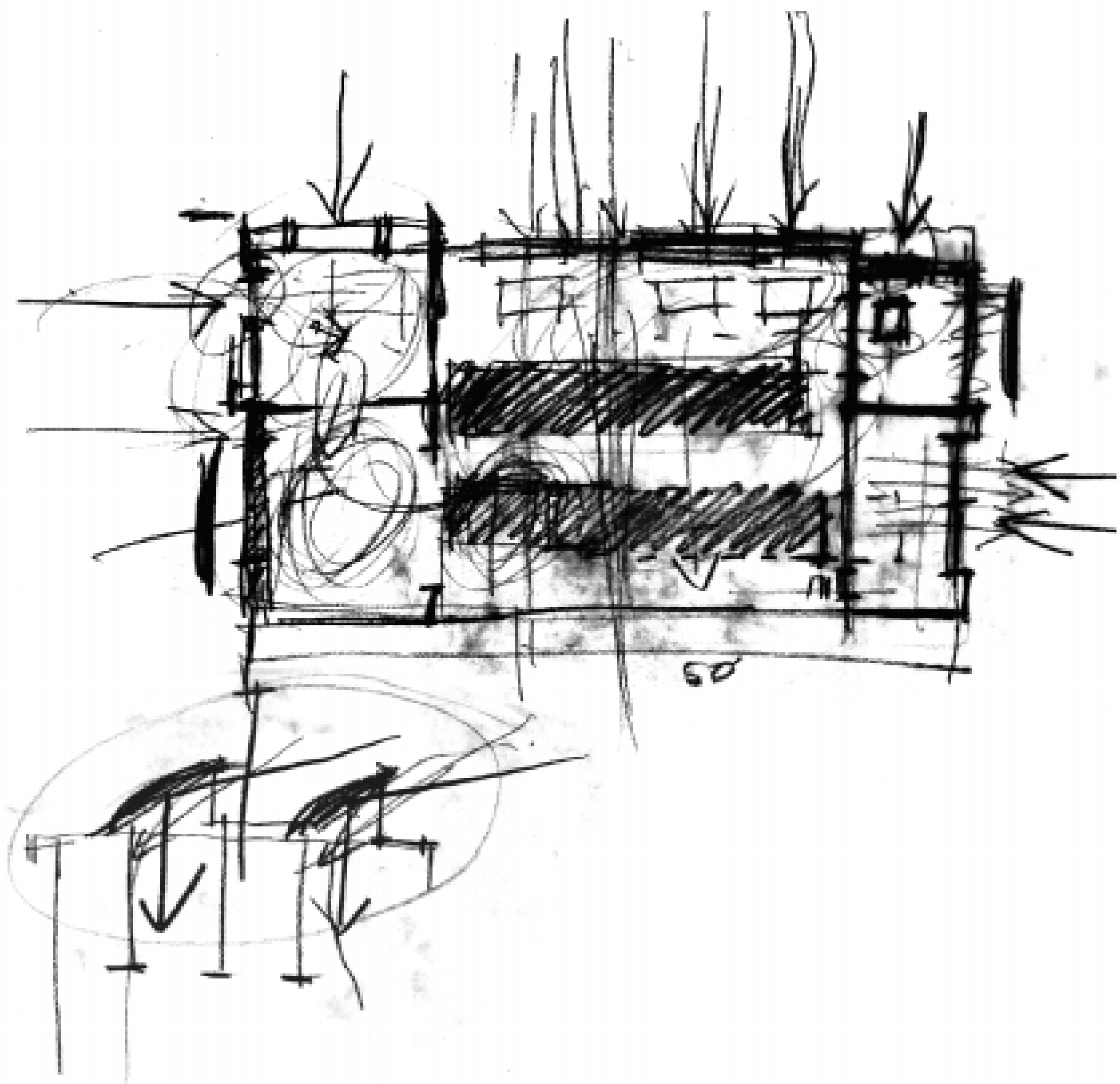


These two drawings were drawn on a yellow trace of 18" width.

Task 2

Please start with a new sheet of tracing paper. This time, please pay particular attention to lighting issues. For example, where should the window be on the wall? And, perhaps introduce skylight into the office if plausible?

Please take no more than 5 minutes to complete this task. Thank you!



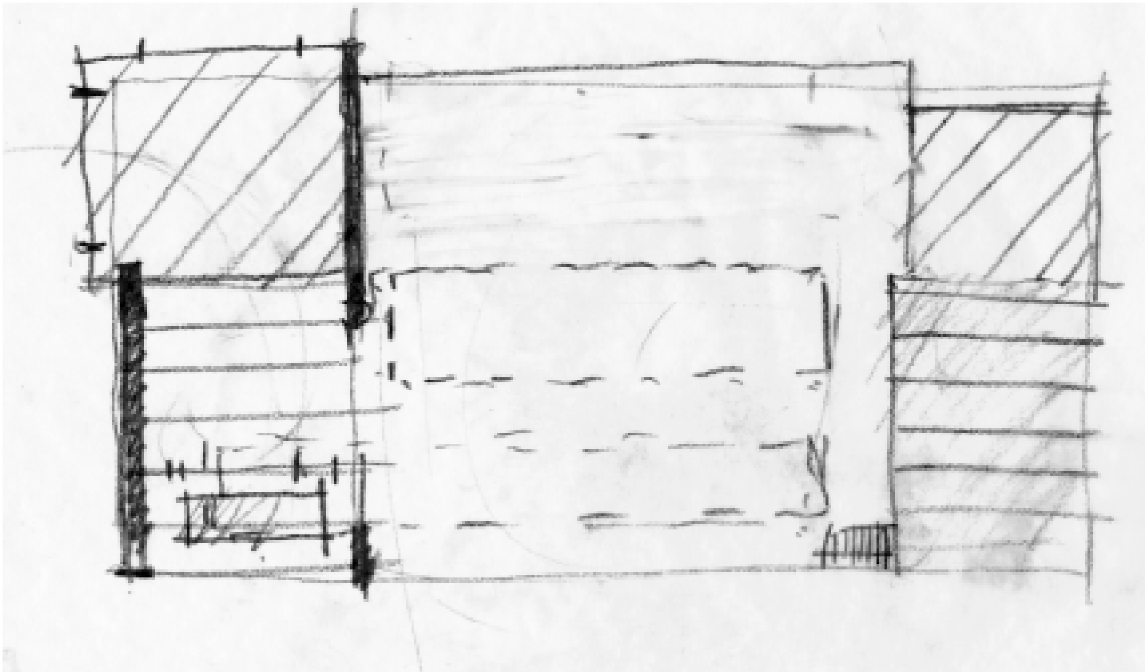
This drawing is on a yellow trace of 18" width.

Task 3

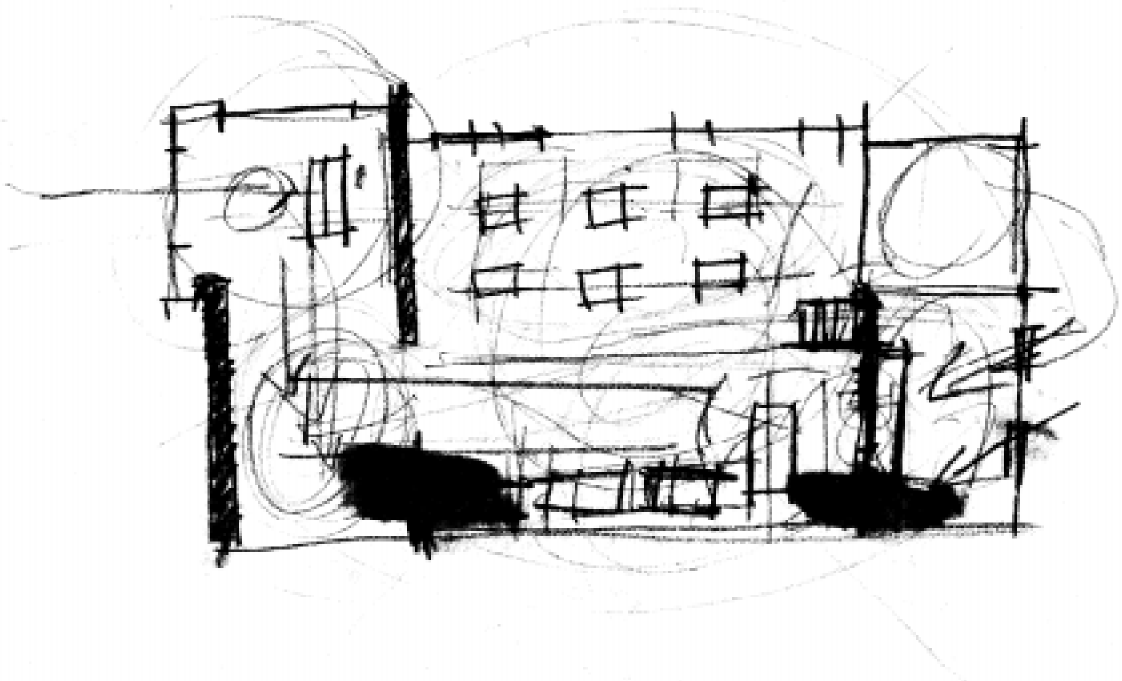
Please start with a new sheet of tracing paper. This time, please focus on the visibility and privacy concerns between different spaces.

For example, consider making each work group space commands a fine view to the exterior, and an easy access to the meeting area.

Please take no more than 5 minutes to complete this task. Thank you!



This drawing was drawn on the upper part of a yellow trace of 18" width and length. The lower part of the trace was folded up to add more detail as shown below.



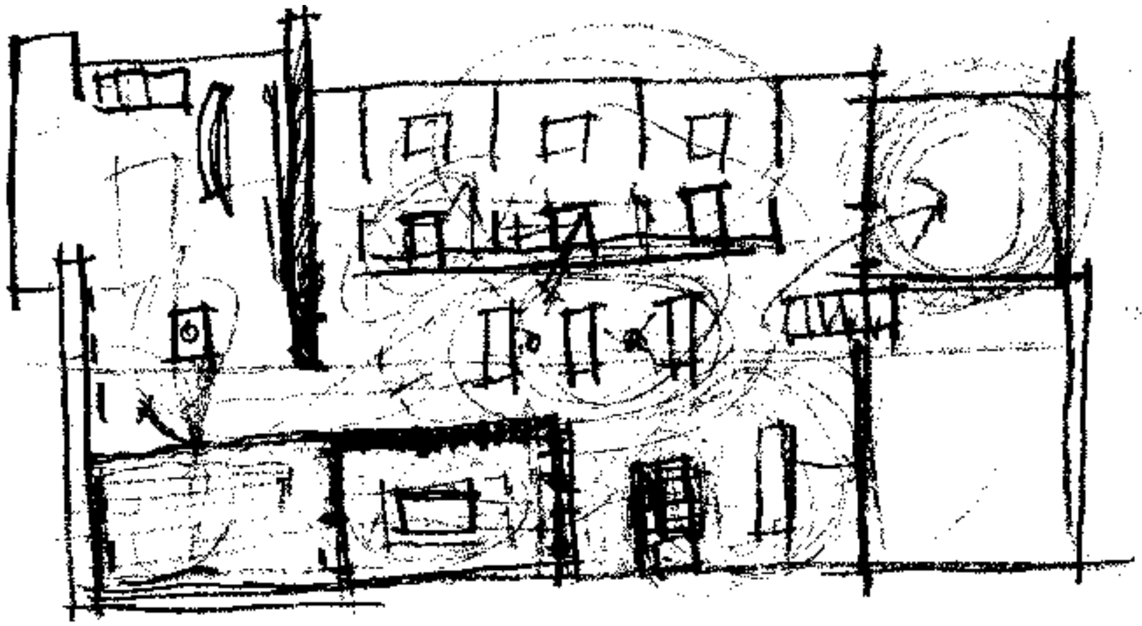
This bottom part was traced on a fold up yellow trace on top of the above drawing. Paper width 18".

Task 4

Please start with a new sheet of tracing paper. This time, please make sure your meeting room is large enough to accomodate the chief architect's favorite meeting table (4 ft x 10 ft). Also, please make sure the designers' work space would have at least 800 sq. ft.

Please take no more than 5 minutes to complete this task.

This is the last task. Thank you! Please turn to next page and give us your comments.



This drawing was drawn on a yellow trace of 18" width.

Please tell us how you feel about this experiment.

Participant ID: ___________________

1. How clearly were the tasks defined?

Task 1	Unclearly	Clearly
Task 2	Unclearly	Clearly
Task 3	Unclearly	Clearly
Task 4	Unclearly	Clearly

2. How difficult to perform the tasks?

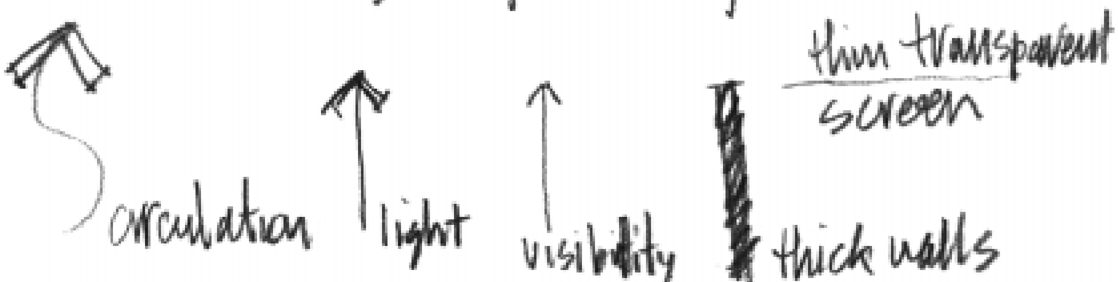
Task 1	Difficult	Easy
Task 2	Difficult	Easy
Task 3	Difficult	Easy
Task 4	Difficult	Easy

3. How often do you use a computer for architecturally related work? What do you use?

Never All the time

4. Do you think you have your personalized symbols and short hands for designing? What are they?

if I do, they are very few



5. Would you like a computer program to have freehand sketching abilities? What should it do?

The ability to hide itself. The idea being when I sketch - I want to SKETCH, thats it, I dont want to fumble with a clumsy interface or be distracted from my sketches

6. Suggestions and comments for the experiment? Thank you!

You need to make it so that you forget you are using the program

APPENDIX G

MARIO’S DESIGN SESSION

Please tell us about yourself.

Participant ID: ____________

Participant Name: ____________ Age: 51 Gender: male

Occupation (Job Title): research arch.

1. Briefly describe your architecture design experience:

worked two or three years in an architectural office in mexico around the time I graduated.

2. How many years of education have you completed?

College (# of years) 8 Graduate School (# of years) 5

3. How often do you use freehand drawing for architectural design?

Never |—|—|—|—| All the time

4. How often do you use freehand drawing to communicate with another designer?

Never |—|—|—|—| All the time

8 units
23 units
35)70
6
-70

Design Brief

An architecture firm just rented a one story warehouse to be its new office space. The space is 70' ft x 25', running East-West the long way. The west side entrance faces a main street. All sides of the space may have openings except the south side, which connects to another building. This firm currently has 1 chief architect. 3 designers. 3 CAD operators. 2 contract draftsmen, a secretary and about 1-2 student interns.

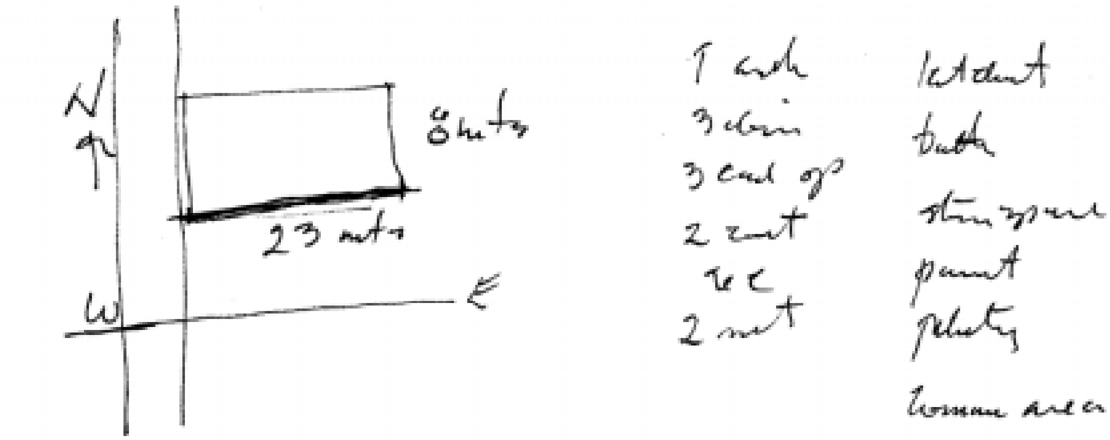
The office will be designed to have, besides the area for work groups of designers, CAD specialists and draftsperson, a meeting room, a small kitchenette, a bathroom, and a chief architect's private office, a secretary - receptionist - general affairs section, storage space, printing and plotting area, and space for student interns.

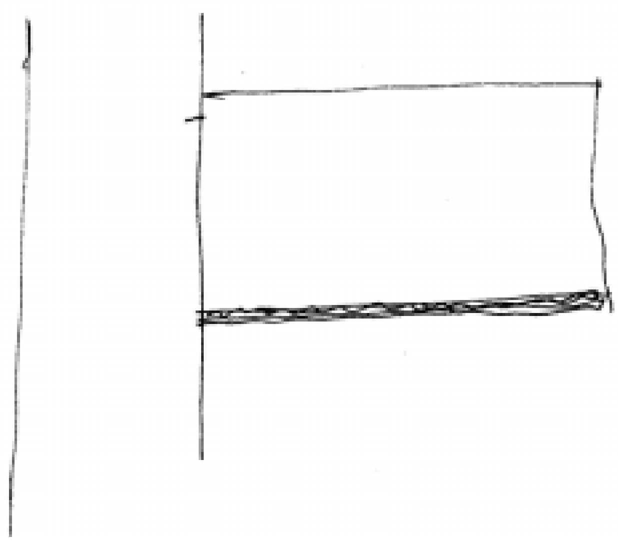
Flexible functional combinations may suit the occupants' various needs. For example, a meeting area can also serve to display past design works, a storage space may include a small reference library, or product catalogue and material samples, and the print room could include Xerox, blueprint and computer plotting machines, etc.

Task 1

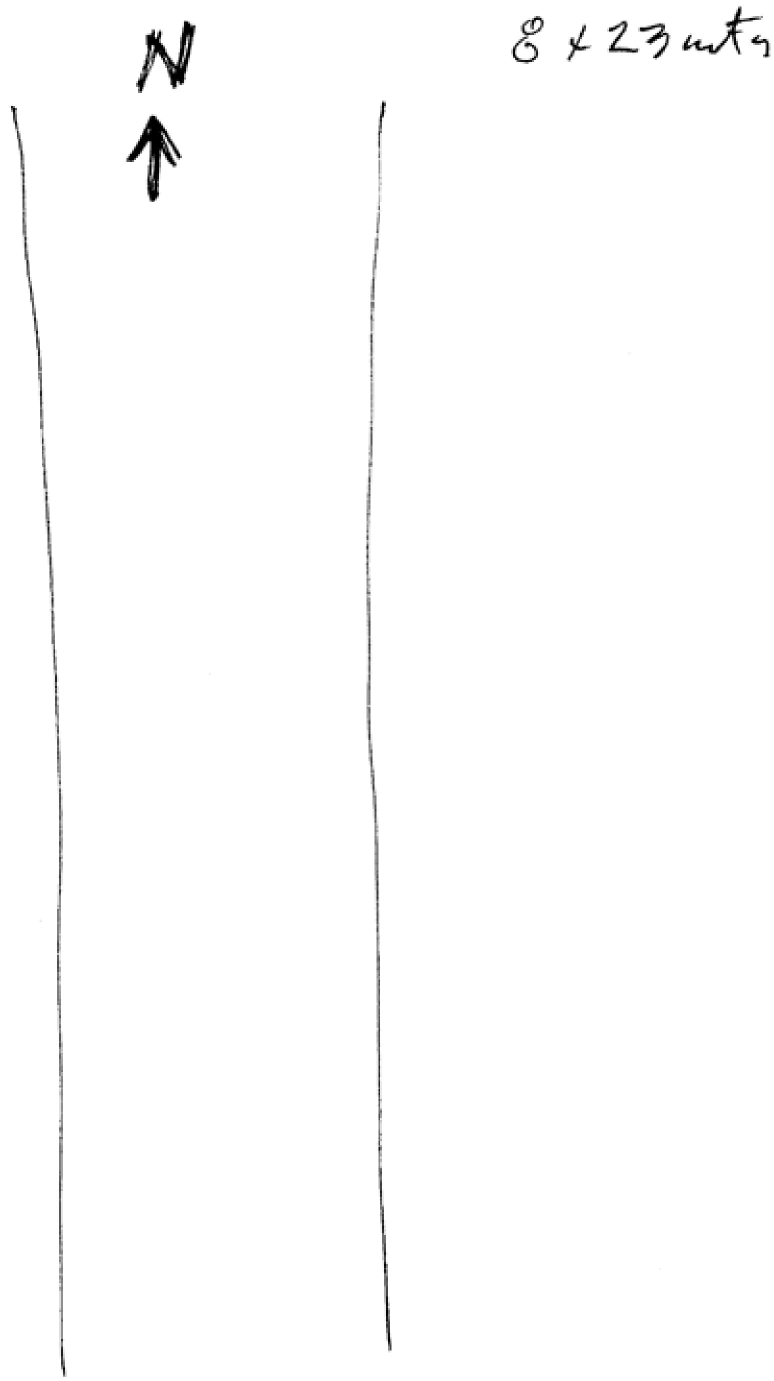
Please take no more than 10 minutes to make a conceptual, schematic design for the above program. Please pay particular attention on zoning arrangements. For example, consider where to put the lobby, chief architect's office, meeting room, and the different work group spaces, etc.

Please continue to the following pages. Thank you!

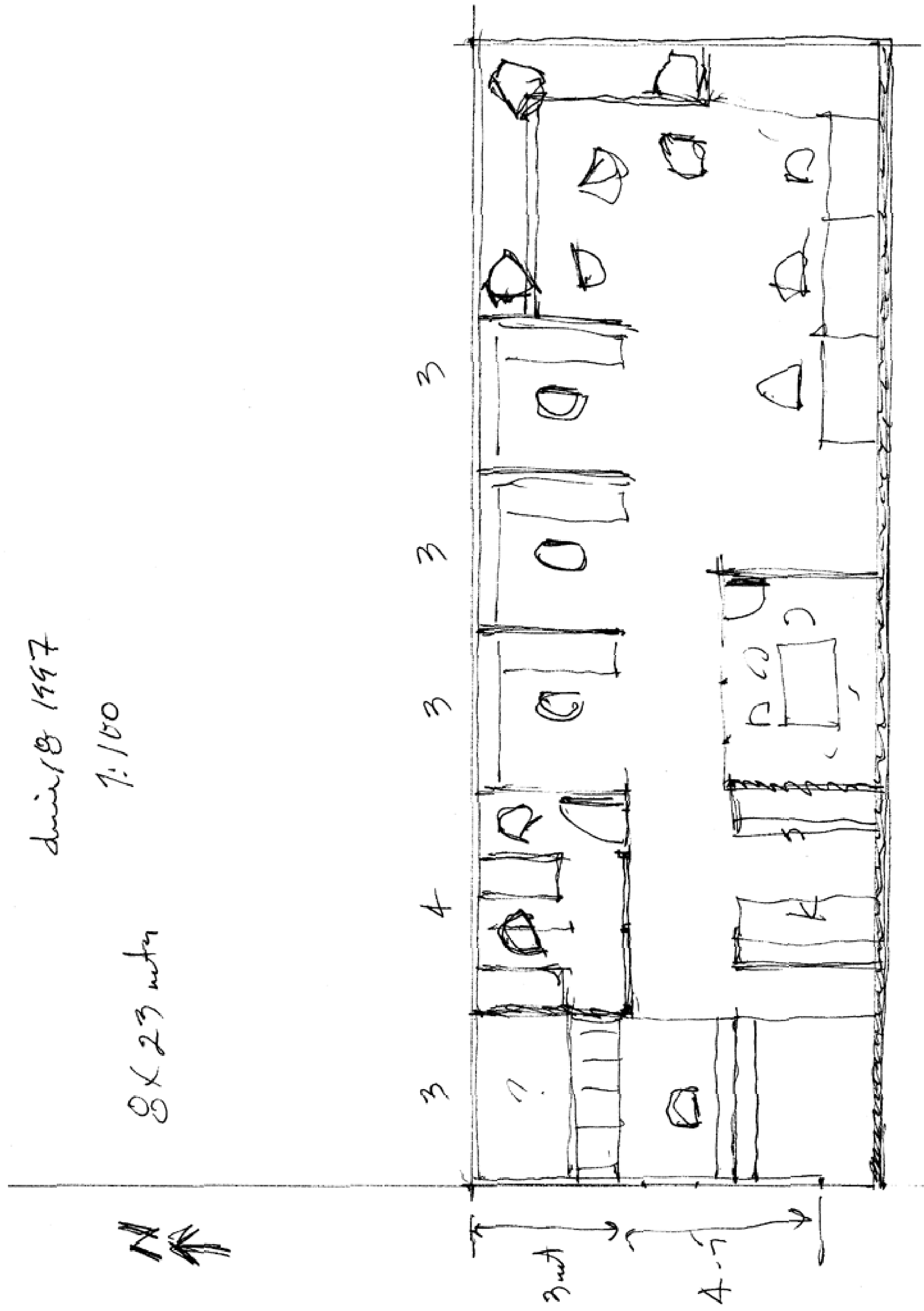




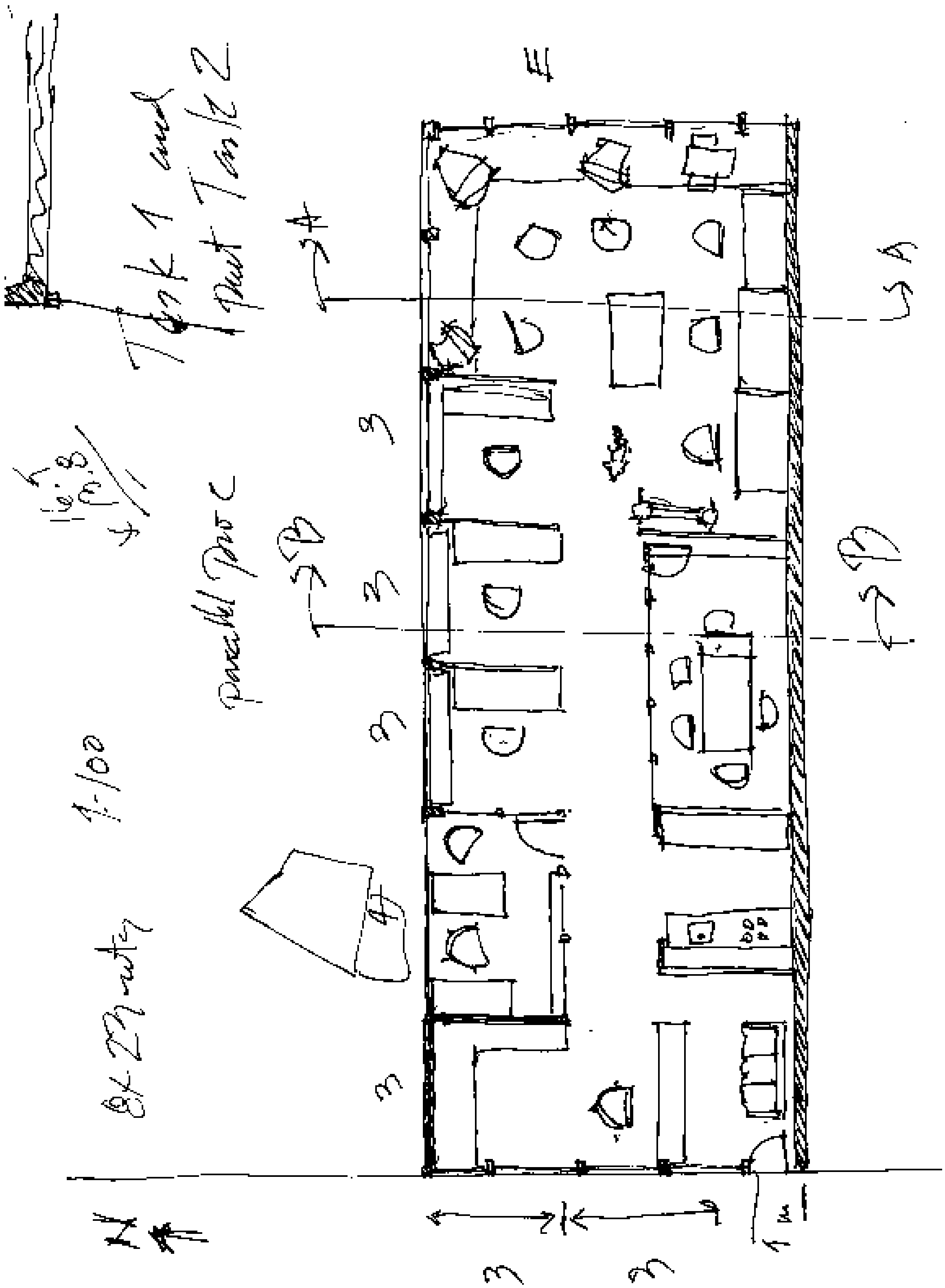
This drawing was drawn on white xerox paper 8.5" X 11".



This drawing was drawn on white xerox paper 8.5" X 11".



3
3
3
4
3
1:100

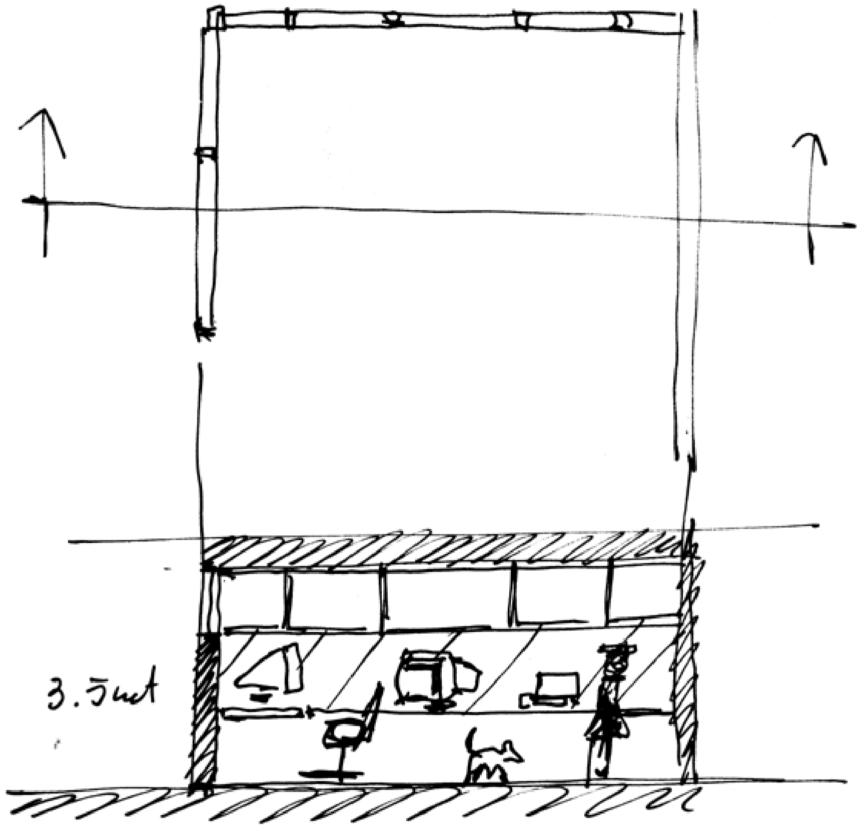


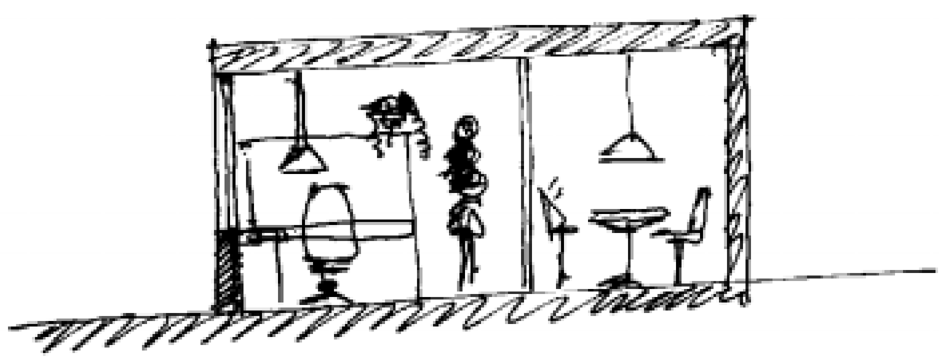
This drawing was drawn on white trace on top of previous drawing (previous page)..

Task 2

Please start with a new sheet of tracing paper. This time, please pay particular attention to lighting issues. For example, where should the window be on the wall? And, perhaps introduce skylight into the office if plausible?

Please take no more than 5 minutes to complete this task. Thank you!





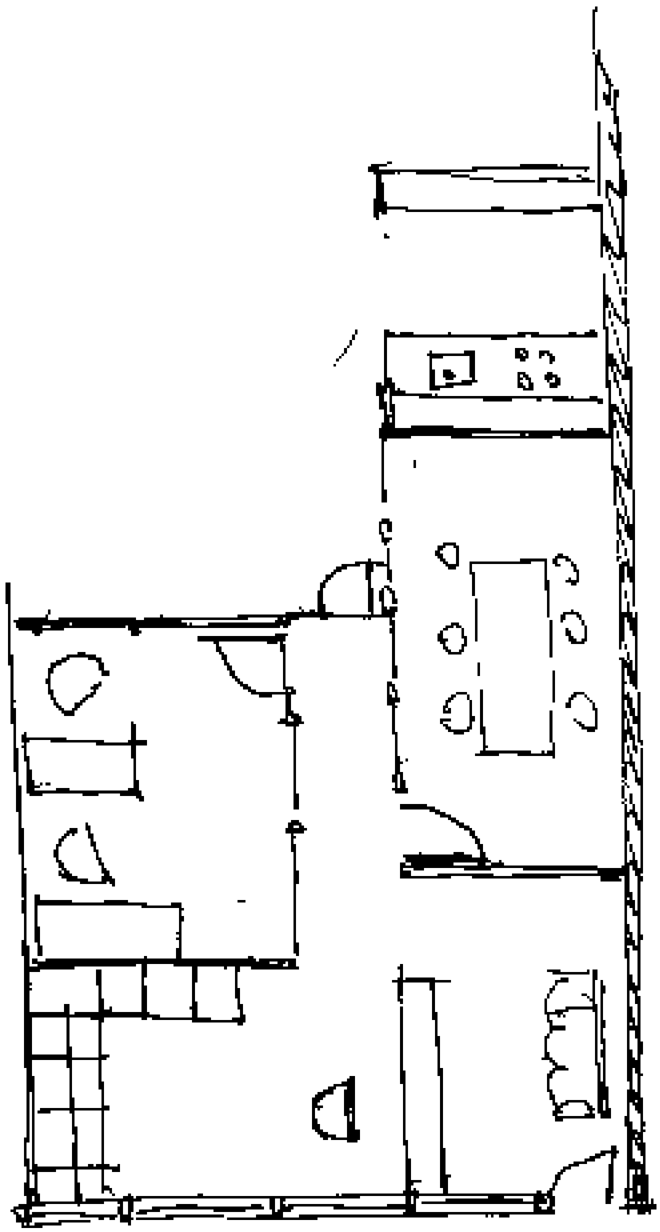
These drawings were drawn on two white trace paper.

Task 3

Please start with a new sheet of tracing paper. This time, please focus on the visibility and privacy concerns between different spaces.

For example, consider making each work group space commands a fine view to the exterior, and an easy access to the meeting area.

Please take no more than 5 minutes to complete this task. Thank you!



This drawing was drawn on a white trace paper of 8.5" X 11".

Task 4

Please start with a new sheet of tracing paper. This time, please make sure your meeting room is large enough to accomodate the chief architect's favorite meeting table (4 ft x 10 ft). Also, please make sure the designers' work space would have at least 800 sq. ft.

Please take no more than 5 minutes to complete this task.

This is the last task. Thank you! Please turn to next page and give us your comments.

1.25
x 3.20

20 x 40
6.7 13

13
3) 40
3
10

213.0
6.7
91.0
280

213
7
91

9
5
45

Please tell us how you feel about this experiment.

Participant ID: ____________________

1. How clearly were the tasks defined?

Task 1	Unclearly	├───┼───┼───┼───┤	Clearly
Task 2	Unclearly	├───┼───┼───┼───┤	Clearly
Task 3	Unclearly	├───┼───┼───┼───┤	Clearly
Task 4	Unclearly	├───┼───┼───┼───┤	Clearly

2. How difficult to perform the tasks?

Task 1	Difficult	├───┼───┼───┼───┤	Easy
Task 2	Difficult	├───┼───┼───┼───┤	Easy
Task 3	Difficult	├───┼───┼───┼───┤	Easy
Task 4	Difficult	├───┼───┼───┼───┤	Easy

3. How often do you use a computer for architecturally related work? What do you use?

Never ├───┼───┼───┼───┤ All the time

4. Do you think you have your personalized symbols and short hands for designing? What are they?

of course

solid wall roof solid
chairs kitchen
people windows
dogs computers
plants

5. Would you like a computer program to have freehand sketching abilities? What should it do?

I don't know

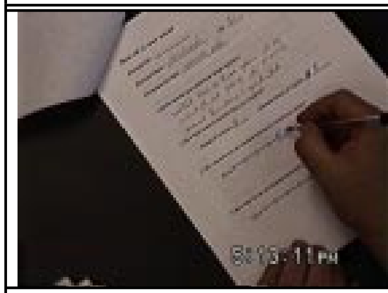
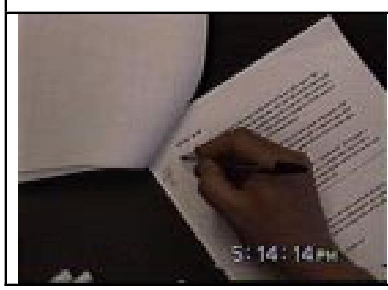
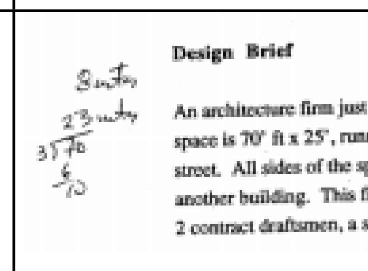
6. Suggestions and comments for the experiment? Thank you!

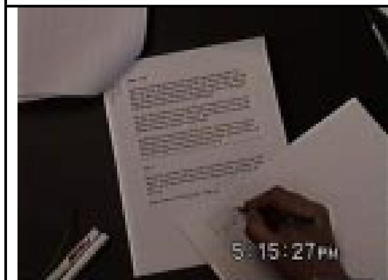
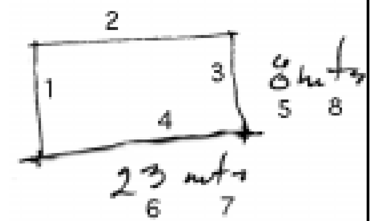
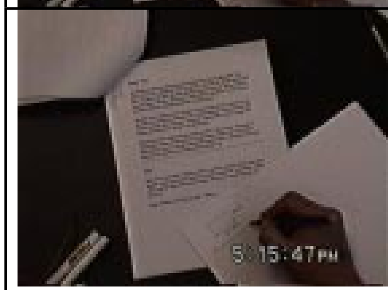
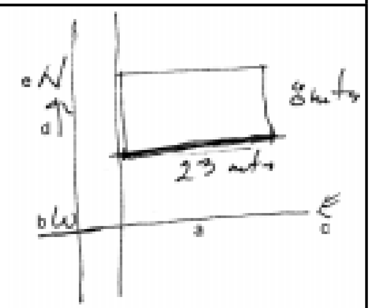
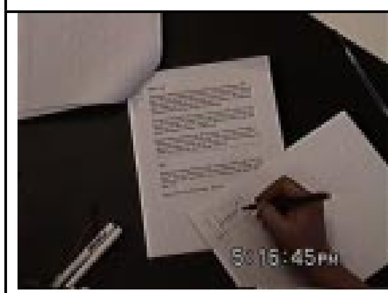
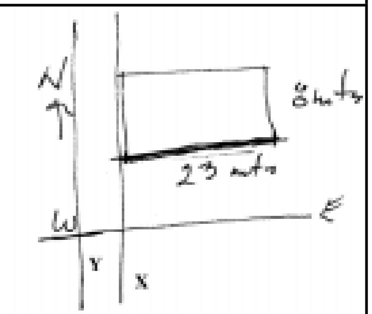
none ok

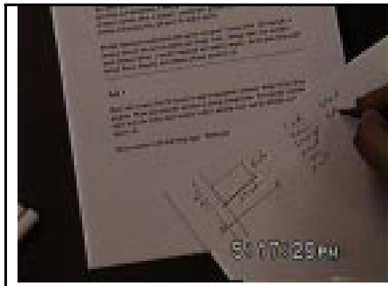
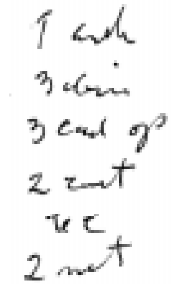
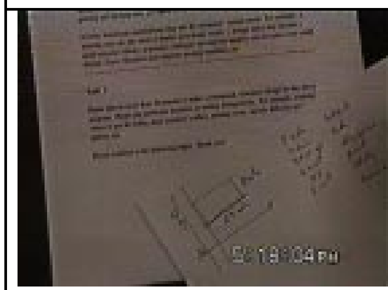
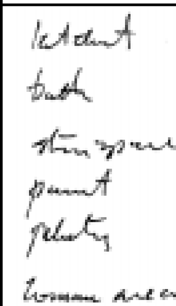
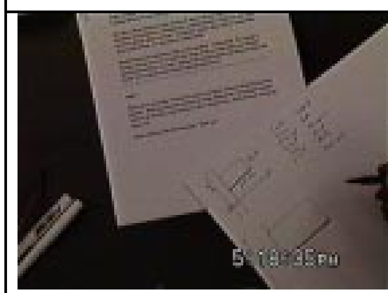
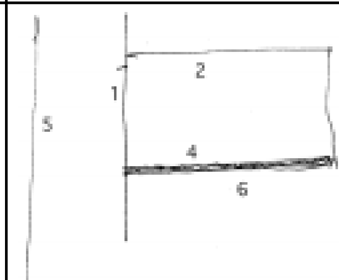
None -
place the
camera in
a more
comfortable place

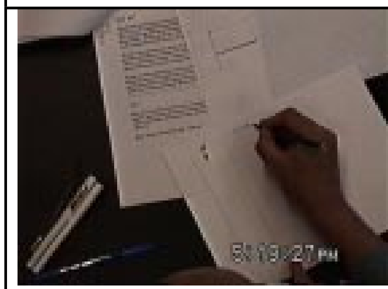
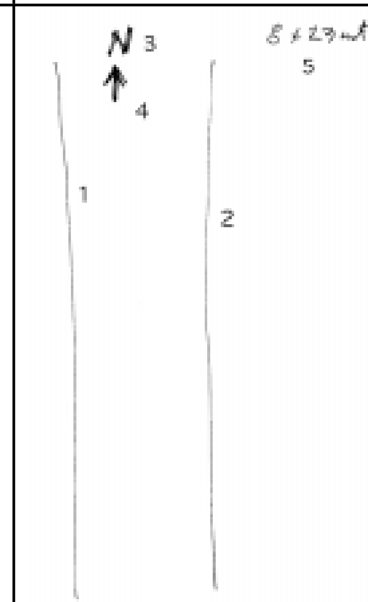
APPENDIX H

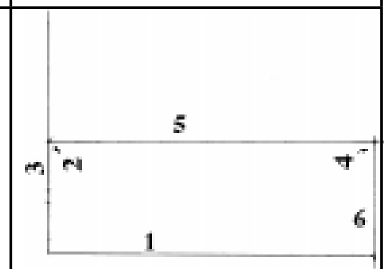
VIDEO TRANSCRIPT OF MARIO'S DESIGN SESSION

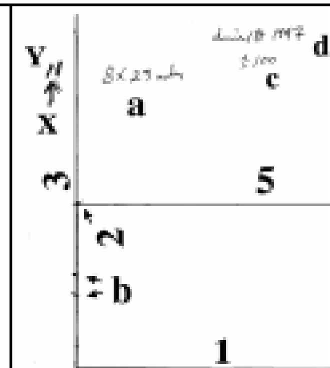
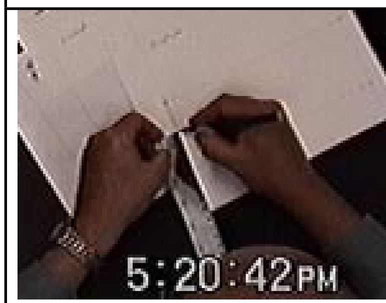
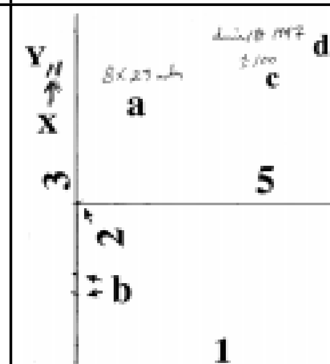
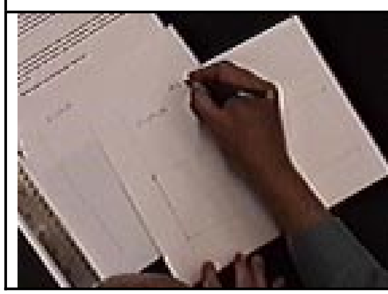
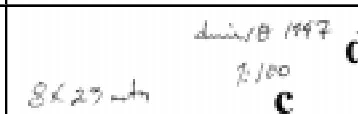
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	00:00	started reading the evaluation sheet, & filling in the pre-test questionnaire	video tap started at 5:11:00 date: June 18, 1997 name: Mario age: 51 gender: male	see Appendix for consent and release form, pre-test questionnaire
	02:22	started reading the program	designer converts program dimensions with number calculation	see Appendix for design program
	03:02 03:03 03:07 03:15 03:22	converting feet to meter dividing numbers by 3 (70/3=23, 25/3=8) **"70 ft by 25 ft, what is 70 ft? divided by 3 is what?"** wrote 70, put lines to divide, put 3, start calculating **"meters"** wrote down "meters" beside 23 **"25 ft is?"** looks at 3, said "8" wrote down "8 meters"	Mario is from Mexico, uses metric system metric conversion calculation was done on the corner of the design program while reading it	

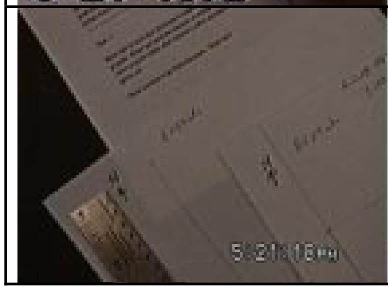
	04:05	started a new piece of paper (page #1, site plan)	this page consists of 4 blocks, and each block consists of several chunks	see Figure 5-19, analysis of page #1
	04:21 04:23 04:25 04:28	drew a small rectangle (sequence #1-4) to represent site wrote 8 (#5) added dimension: "23" (#6) and "meters" (#7) wrote "meters" beside "8" (#8)	• Block 1 -- Site * Chunk 1 -- site & dimension left west top north right east bottom south width length	
	04:47 05:14 05:20	drew a line below site (#a), wrote W (#b), & E (#c), & drew an arrow (#d) wrote N (#e) **"I am finished"**	*Chunk 2 -- orientation horizontal line west east arrow up north	
	05:30 05:36	read the program, **"yes, north side... west side facing main street.."** add streets, two vertical lines (#X, & Y).	*Chunk 3 -- street vertical street parallel to the left	

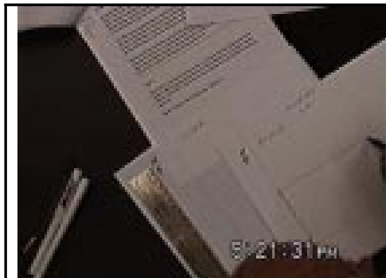
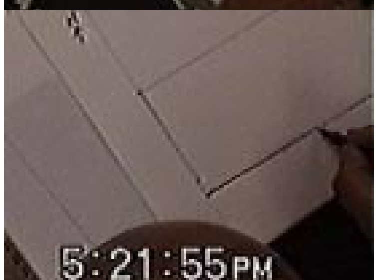
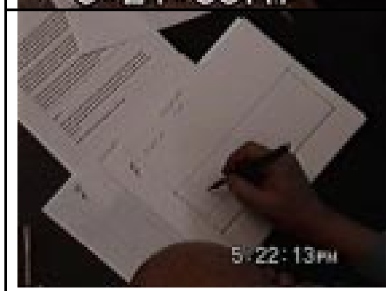
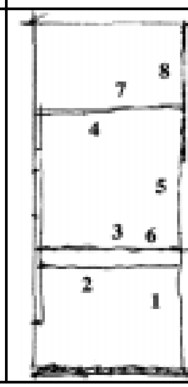
	05:57	wrote program requirements of people while reading the brief: **"1 architect, 3 designers, 3 CAD operators, 2 contract draftsmen, ... secretary, 2 students"**	<u>• Block 2 -- program, people</u> 1 arch(tect) 3 design(ers) 3 CAD op(erators) 2 cont(ract draftsmen sec(retary) 2 st(udents)	
	06:26	wrote & read next paragraph about space requirements from the design program: **"kitchenette, bathroom, storage space, printing, plotting, and common area"**	<u>• Block 3 -- program, space</u> kitchenette bath(room) storage space print plotting common area	
	07:21 07:22 07:23 07:24 07:25 07:31	started drawing the site again, first a vertical line as street and site boundary (#1) then continue to complete a rectangle (#2-4), drew top boundary (#2) right/east side (#3) bottom/south side (#4) then add street (add line #5) over trace bottom horizontal boundary (#6)	<u>• Block 4 -- site again, in larger scale</u> vertical (left) west (face main street) top north right east bottom south (sequence same as Block #1, Chunk #1) vertical street parallel #1 to the left south side (connects to another bldg.)	

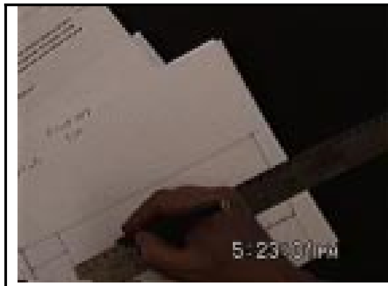
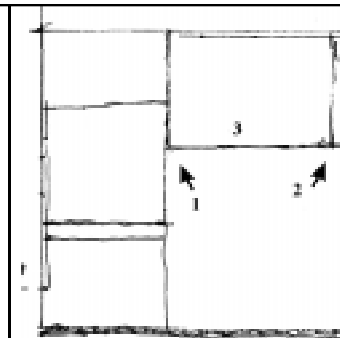
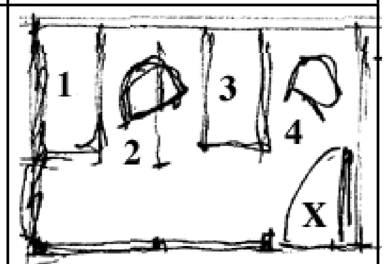
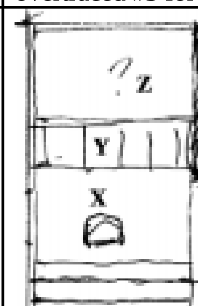
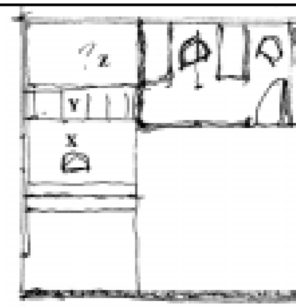
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	07:40	reached a new piece of paper (page #2, enlarged site) lined this piece of paper with the rest		see Appendix for page #2
	07:44 07:52 08:06 08:11 08:27	drew lines for street (#1-2) then asked **"can I have a scale?"** wrote a letter N (#3) drew an arrow pointing up (#4) wrote down "8X23 meters" (#5)		

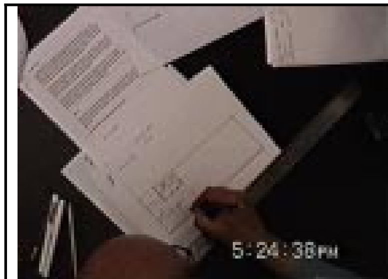
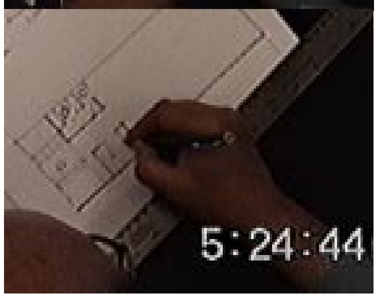
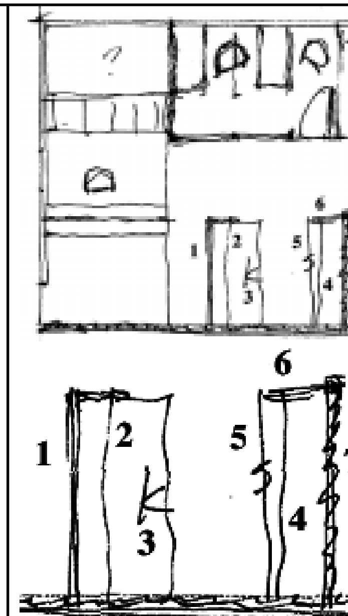
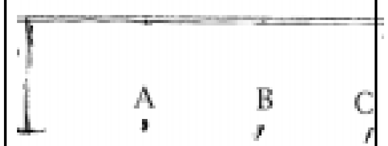
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	07:32	scale arrived, designer reached for new paper (page #3, floor plan)		see Appendix for page #3
	08:41 08:42 08:45 08:50 08:51 08:53 08:55 09:00 09:01 09:03 09:05 09:06 09:07 09:09 09:12 09:16 09:23	put scale on the new tracing paper #3, landscape orientation drew horizontal line (#1) along ruler turned paper (90° CW) to portrait orientation dotted along scale (#2), drew horizontal line (#3) turn papered (90° CCW) to landscape orientation, put scale to see where 23 meter is turned paper back (90° CW) to portrait orientation dotted on 8 measure on scale ruler (#4), turned paper (90° CCW) back to landscape orientation **"Scale gives me an idea about proportion... this is the way I do it"** turned paper back to Landscape orientation and put down scale along the dots drew horizontal line (#5), moved scale to vertical position and lined up with dot #4 and end of line #1, drew vertical line #6 **"I need to get an idea how the proportions are".** searched for old trace pages	<u>• Block 1 -- Site</u> <u>* Chunk 1 -- boundaries of the site</u> started new paper (page #2), drew rectangle site on landscape orientation, (horizontal is the long side) 5:20:05 observer asked "why use scale?"	 see Appendix for page #3

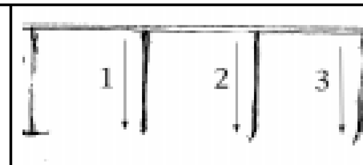
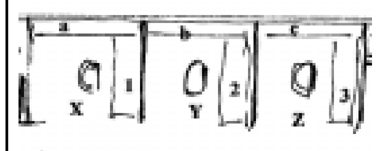
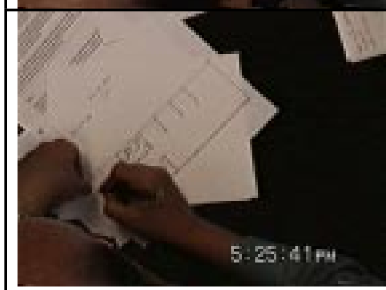
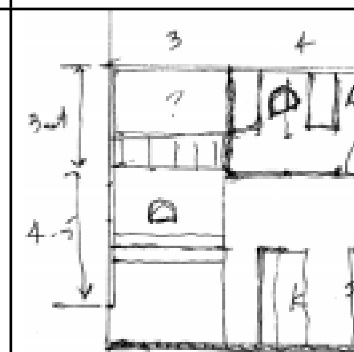
	09:24 09:25 09:29 09:30	(observer commented) **"still have to get the right sense of dimension"** pointed to trace page #1, **"this is the way I did it first time, the proportion turns out was all wrong** ..." hand moved back to current tracing paper	5:20:24 observer commented "this is only for conceptual design"	
	09:34 09:36 09:40	**"okay,. It's 8 by 23 meters,** wrote "8 x 23 meter" (#a) put scale down by line #3, **"size of the door"** put two dots (#b)	* <u>Chunk 2 -- dimensions & scale</u> dimensions 8 x 23 meters scale -- size of a door	
	09:49 10:02 10:08	**"so, what do we got?"** wrote down 1:100 (#c) wrote down date (#d, Junio 18, 1997)	* <u>Chunk 3 -- notations</u> scale 1/100 date -- June 18, 1997	

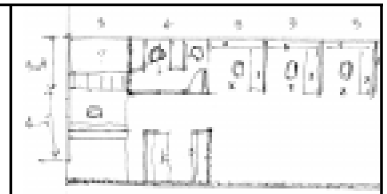
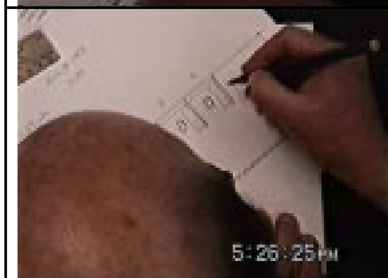
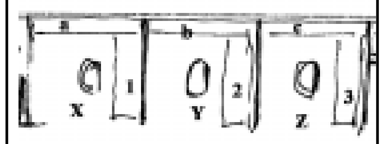
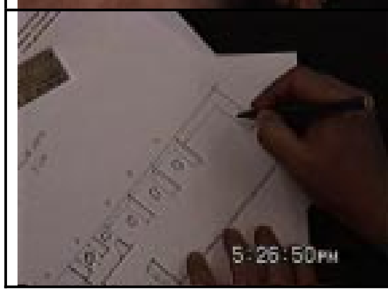
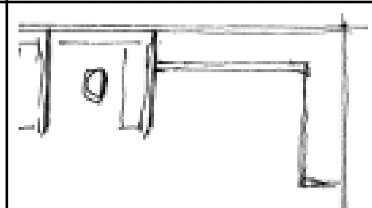
	10:13 10:15	drew arrow (#X) wrote "N" (#Y)	<u>* Chunk 4 -- orientation</u> arrow up north	
	10:18	looked at program and site		

 	10:24 10:25 10:26 10:27 10:30 10:38 10:40 10:42 10:47	dotted on the bottom of west wall, added line on top of west wall add dots for windows on west wall, overtraced west wall, trace north wall overtrace bottom wall, (observer questioned) hatched bottom to be wall, **"these are walls, right? You can't penetrate them"** (hatching between the wall lines) overtraced right wall	* Chunk 5 -- site again distance from dot to bottom is size of door west north south (make solid wall with hatching) east wall 5:21:40 observer asked "what are those?"	see Appendix for Page #2
	11:09 11:12 11:13 11:14 11:16 11:18 11:26 11:30	drew small rectangle on left bottom corner (#1) traced the top horizontal line (#2) added another parallel line on top (#3) drew one more line closer to top wall (#4) added vertical line connecting them all (#5) overt raced top wall for top of bottom room (#6) over traced top bottom of top rectangle (#7) over traced right walls of top rectangle (#8)	* Chunk 6 -- entrance & reception area	

	11:40 11:55 12:00 12:03 12:12	**"okay, so we have got one architect, three, three what?"** used scale to measure vertical line, dotted (size of architect's office) used scale to measure horizontal line, dotted, (width of architect's office) drew horizontal line (bottom wall of architect's office) **"we have space for receptionist... and...."** (pointed to space on the left)	* <u>Chunk 7 -- architect's office</u> considered the next space for 3 designers drew south wall for architect's office considered previous receptionist's space	
	12:20 12:30 12:44 12:59 13:01 13:02	overtraced wall, window, for the architect's office which is next to receptionist room added door (#X) drew desk (#1), chair (#2) drew desk (#3), chair (#4) observer questioned **"chairs, these are chairs for architect and client"** (pointing and overtrace the 2 half circles in the room)	* <u>Chunk 8 -- furniture in architect's office</u> door desk, chair desk, chair 5:24:01 observer asked "what's that? " designer responded by overtracing #2, said it's for the architect, and overtraced #3 for the client	
	13:10 13:17 13:20	"this is for receptionist" (drew chair #X) **"file cabinet"** (while making 5 vertical lines from left to right, #Y) **"service area?"** (drew a question mark on the space above file cabinet, #Z)	 drew chair for receptionist	

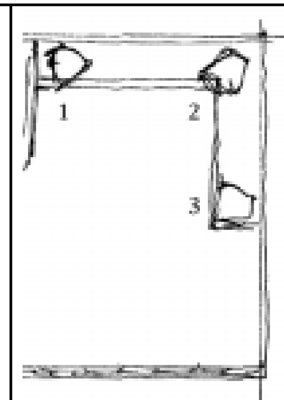
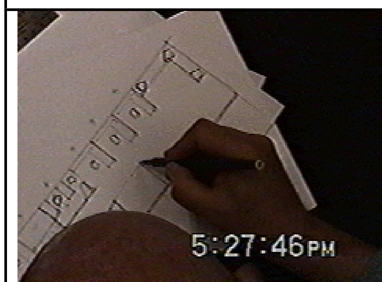
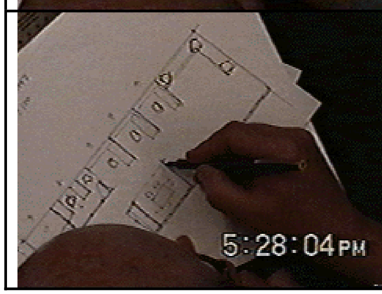
	13:29 13:34 13:41 13:44 13:47	drew vertical line to define a space #1 add square lower left (#2) **"kitchenette"** (wrote K inside, #3) drew another square (#4), next to the vertical wall line, said **"storage"** traced the square (#5), hatched the vertical wall line (#6), and wrote "S" (#7)	designer first defined the space then added kitchen counter (wrote "K") and then added storage counter (wrote "S")	
	13:53 13:56 14:06	**"draftsman, what?"** **"three designers, okay, so let's line them up here"** used scale to measure a space, dotted one (#A), two (#B), three (#C)	designer read the program, identified a need for 3 designers designers used scale to make 3 even designers' space	

	14:10 14:15 14:18	drew vertical line from the dot (#1) drew another vertical wall line (#2) drew another vertical wall line (#3)	designer drew three repeated wall lines from the top of the wall to the dots made previously with measurement from the scale	
	14:26 14:31	horizontal line, one, two, three (table # a, b, c) vertical square (table #1?)	designer drew 3 horizontal lines in the 3 previously defined space then added a table on the left most space	
	14:38 14:50	moved to the left of the drawing, used scale, and wrote dimension "3 meters" used scale, for lower part, wrote "4-5"	designer adding dimensions for the length of the rooms	

	15:05	wrote on the top: 3, 4, 3, 3, 3	designer put down dimensions for the width of individual space	
	15:15 15:28	drew table, table (#2. 3) drew chair chair chair (chair # X, Y, Z)	continued to draw table, and then drew 3 chairs	
	15:38 15:58 16:01	drew L shape table on the top right corner use scale to adjust width drew a parallel line to make the table wider	after considering the 3 designers workspace, designer moved on to draw table to prepare CAD workstations	

<table>
<tr>
<td>
</td>
<td>16:09

16:17</td>
<td>drew shape, shape, shape (monitor #1 on left of table, #2 on the corner of L, #3 on ends of L shape table)
overtraced monitor 1, then 2, 3</td>
<td>designers drew shapes for computer monitors

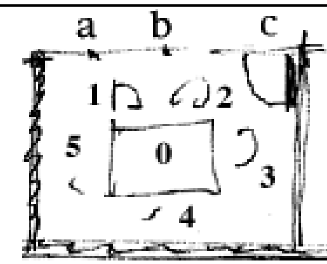
the shapes could be identified as computer monitors because
1) 3 CAD operators is the next item after 3 designers
2) at 23:40 designer pointed to them and said "these are monitors"</td>
<td>
</td>
</tr>
<tr>
<td>
</td>
<td>16:43
16:44</td>
<td>drew new vertical line (right boundary of conference room, #1)
drew horizontal line (top boundary of conference room, #2)</td>
<td>designer defined a new space for conference room</td>
<td>
</td>
</tr>
<tr>
<td>
</td>
<td>16:49
16:50

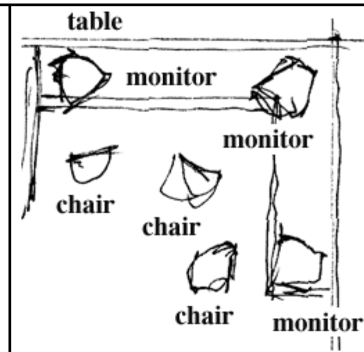
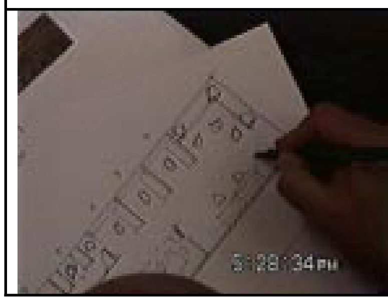
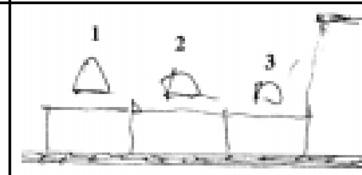
16:55

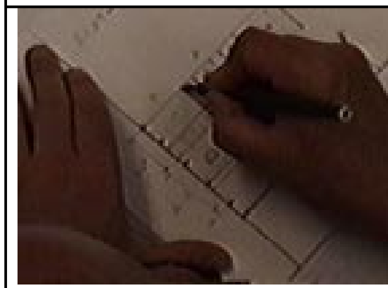
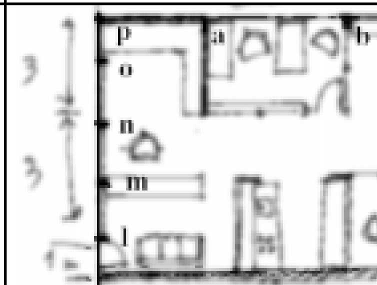
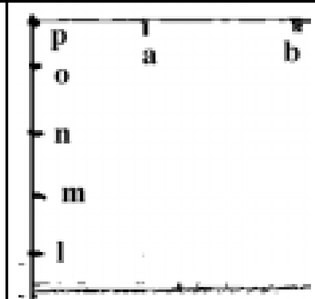
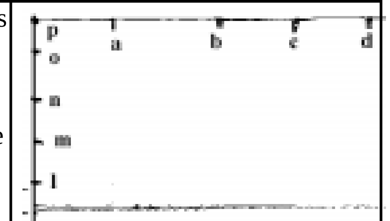
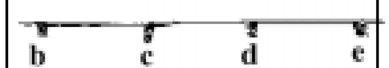
17:03

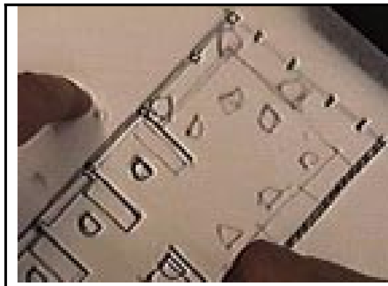
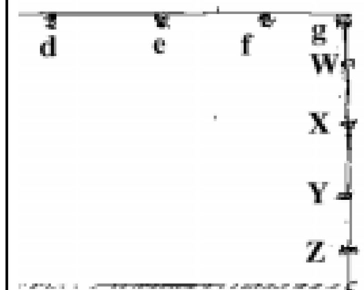
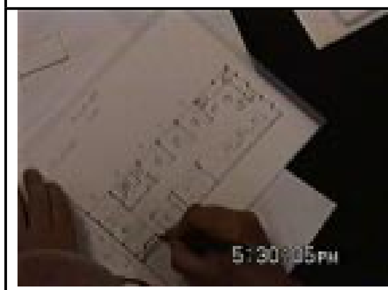
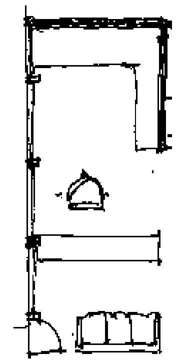
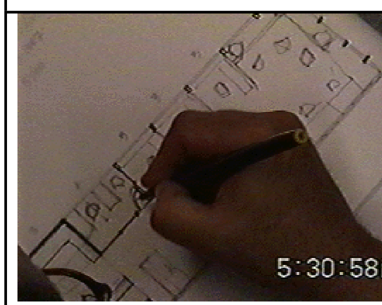
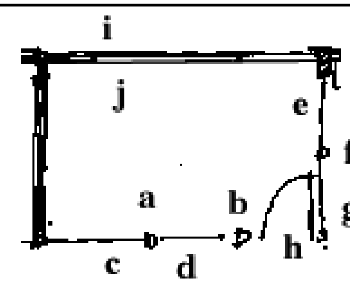
17:09</td>
<td>drew a rectangle (big table # 0)
drew chair, chair chair (half circles #1-3) and then curve, curve (chairs? #4, 5)
"table done? general storage, plotting"

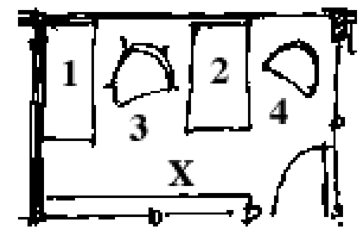
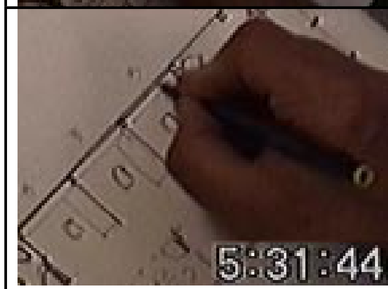
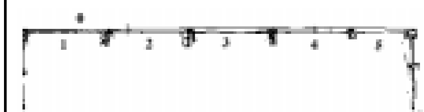
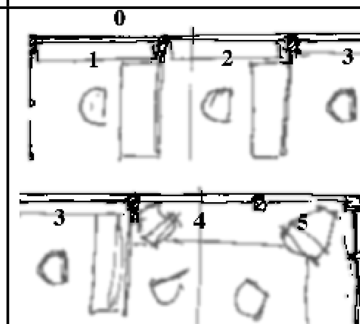
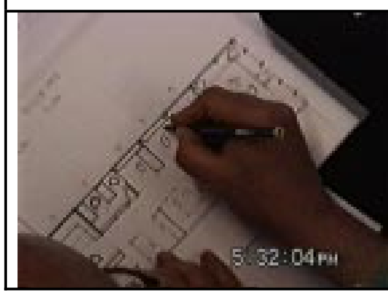
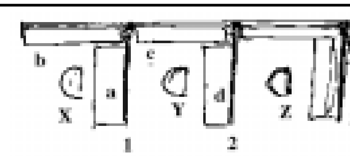
dotted along top wall to make window partitions (# a, b)
added a door (#c)</td>
<td>designer first drew a conference table, then added 5 chairs

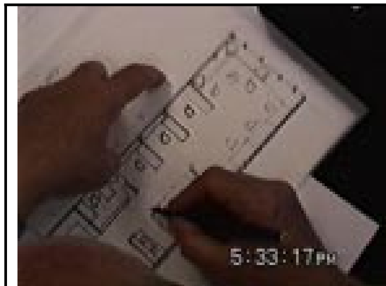
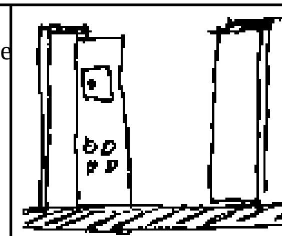
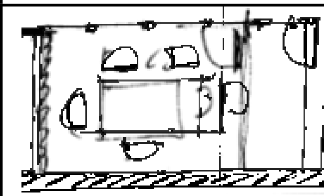
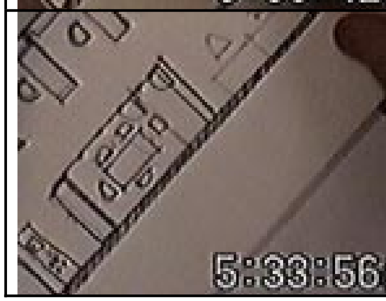
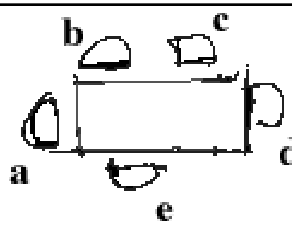
designer looked to the right of the conference room and commented about a plotter (not drawn until 23:46)</td>
<td>
</td>
</tr>
</table>

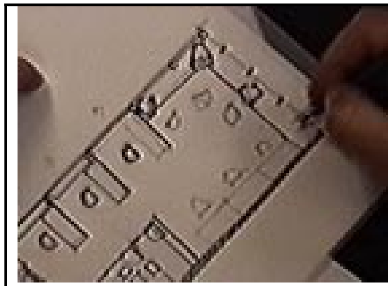
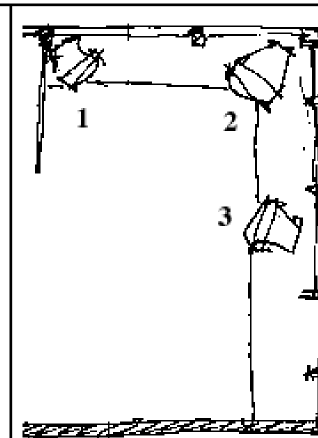
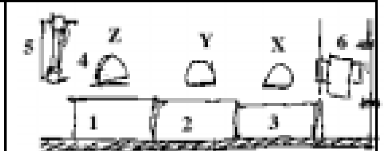
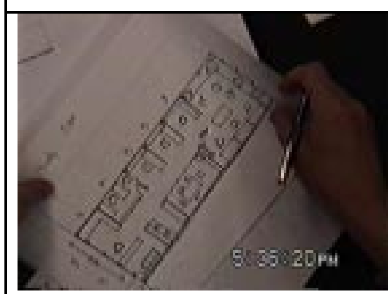
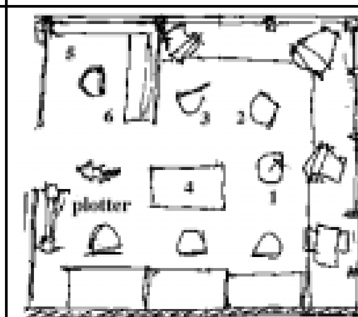
	17:12 17:14 17:15	**"sitting here"** (drew half circles 1, 2, 3) observer questioned **"Umm"**	5:28:14 observer asked "what's that? a chair?" designer moved to the CAD area and added chairs for the computer workstations	
	17:24 17:31 17:33 17:38 17:42	extended L table and added a mirrored lower L table divided desk to three parts (from right to left) drew half circles (chair #1, chair #2, chair #3) **"they are all here, that's it?"** **"you have tracing paper?"**	after CAD workstations area, designers went on to create more working space for the interns	

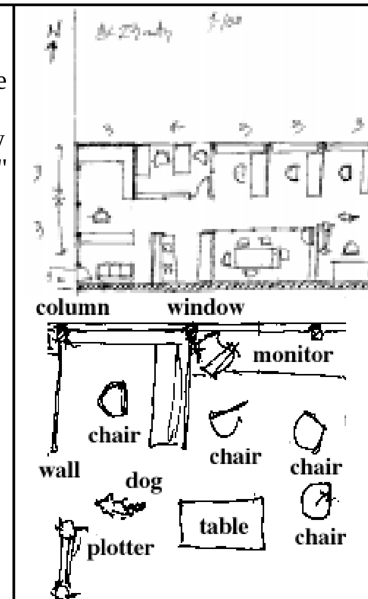
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	17:50	new tracing paper (page #4, trace, plan)		
	18:04 18:10 18:16	lay tracing paper on top of the drawing just made drew left most vertical line added 5 dots for divisions on vertical line (#l, m, n, o, p)		
	18:26 18:28 18:29 18:31 18:33 18:35 18:36 18:37	added 7 rectangular dots from left to right -- overtraced small rectangle repeated 7 times on top horizontal line column #a column #b column #c column #d column #e, column #f, column #g)	#l is for the door, #m for receptionist's desk, #n, o for windows #a is for right wall of receptionist area #b is for right wall of architect's office #c, d, e for right partitions for designers space (#e also for left of CAD space) #f is another column, and #g lands on the corner of walls	
	18:37	**"columns"**	designers said "columns" while finishing the last small rectangles	

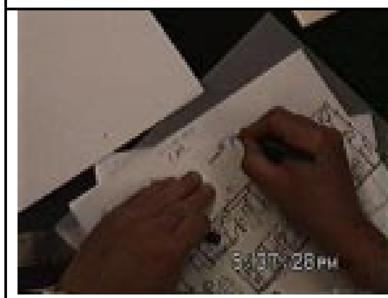
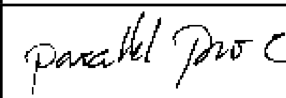
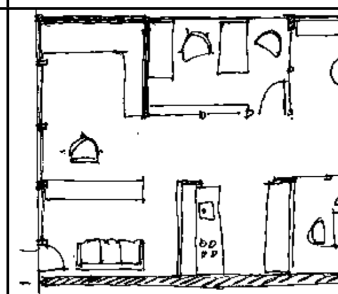
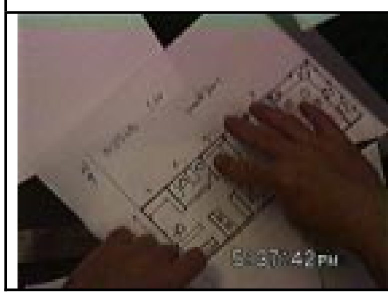
	18:41 18:44 18:48 18:51	moved hand to left division dot, and moved horizontally and dotted an horizontal bar on the right wall (repeated 4 times) (#x) repeated action, on a lower division dot, moved and dotted horizontal bar on right wall, moved down (#y) moved up and added another horizontal bar (#w) moved down, another horizontal bar (#Z)	the partitions on the right walls (# w, x, y, z) lined up with the partitions of the left wall (#o, n, m, l)	
	18:55 19:03 19:09 19:20 19:23 19:31 19:35 19:40	added door to left bottom room added sofa over traced left wall, added window dots, double line wall drew a rectangle (receptionist's table) drew half circle (receptionist chair) drew upper left horizontal line made a new L shape, overtraced and darken walls	designer drew furniture on top of the underlay (page #3) this is the reception area new furniture this layer sofa L shape desk / file cabinet	
	19:48 19:50 19:52 19:54 19:55 19:56 20:00	on right of reception area, added partitions (2 dots, #a, b) on horizontal wall (bottom boundary of architect's office) overtraced horizontal walls (#c, d) overtraced higher vertical wall (right of office, #e) added a column (#f) overtraced lower vertical wall (#g) added a door (#h) overtraced top wall, double lined (for windows, #i, j)	this is the architect's office, places right next to reception area by the L shape desk designer drew partition wall lines to define the space window mullions/columns single line windows window, column, window door double lined window/wall	

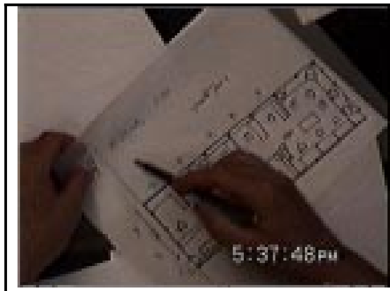
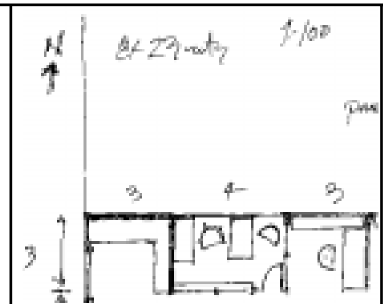
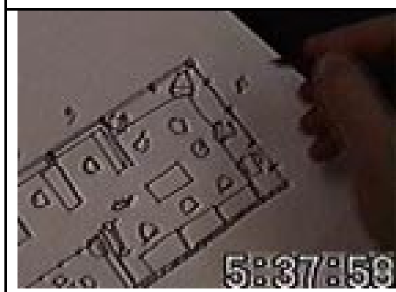
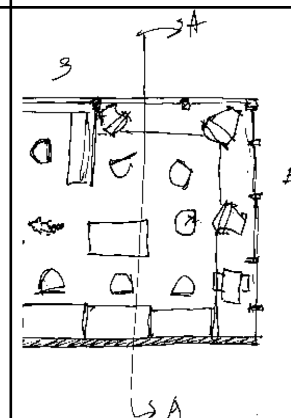
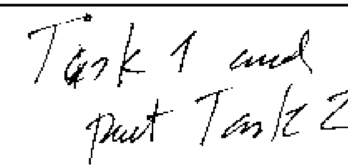
	20:03 20:12 20:15	table 1, table 2 (#1, 2) added horizontal line parallel to bottom wall (#X) chair 1, chair 2 (#3, 4)	placement of furniture to define space desk chair desk chair long narrow counter/file cabinet?	
	20:35 20:41	drew one line, overtraced top wall (#0) drew parallel walls connecting columns 5 times from left to right (#1 - 5)		
	20:53 20:58 21:02 21:07 21:09 21:10	traced vertical over three time (#1, 2, 3) drew vertical table, horizontal table (a, b) drew horizontal table, vertical table (c, d) chair (X) chair (Y) chair (Z)	vertical partition 1-2-3 vertical table 1-2 horizontal table 1-2 horizontal table 3, vertical table 3 chair 1-2-3	

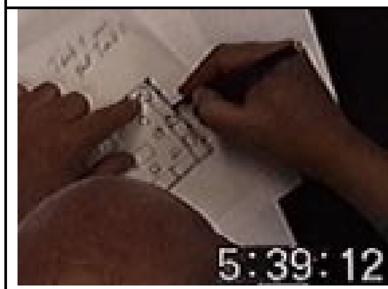
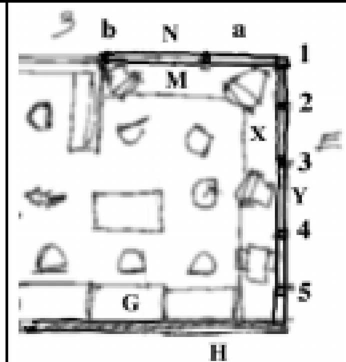
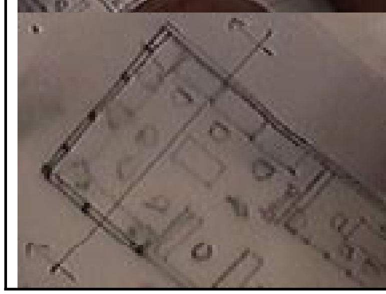
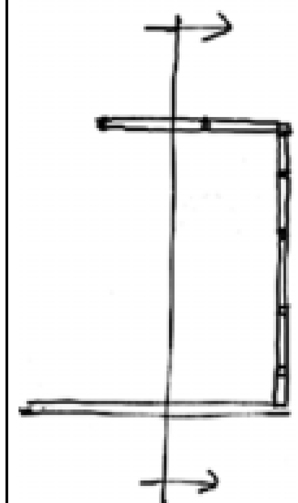
	21:20 21:30 21:49 21:50 22:04 22:13	**“too small”** while looking at conference room lower kitchen darken lower walls hatch the wall over trace left box of kitchen drew a box, and small circles stove **“put stoves here”**	too small referred to the conference room, in next chunk, designer made the conference larger by moving the wall and door to the right of underlay plan	
	22:28 22:33 22:35	drew a vertical wall at the right of the vertical wall in the underlay, and traced this new line over added 4 dots on the top wall, and drew a door next to vertical wall (moved the underlay door to the right) **“yup”**	(this extends the conference room larger to the right) 5:33:35 observer asked “are you making it larger?”	
	22:48	drew a table, then 5 chairs (CW: chair # a, b, c, d, e)	clockwise	

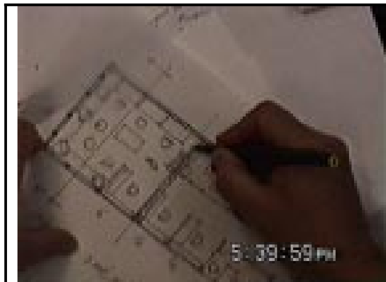
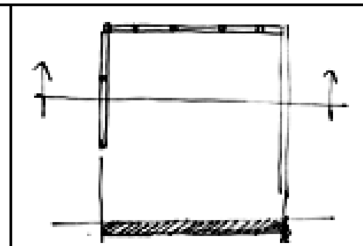
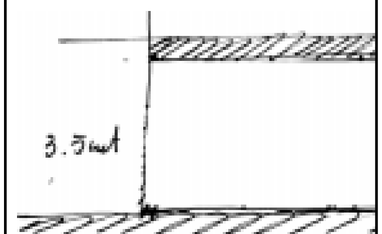
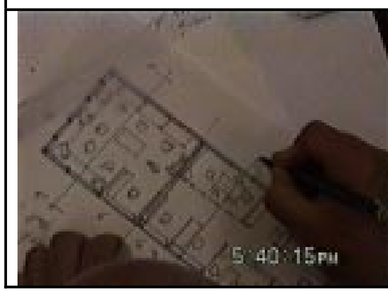
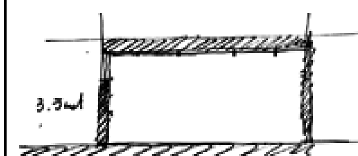
	23:02 23:20 23:21 23:25	overtraced monitor, 1-3 traced L shape table (where the monitors sit on) overtraced right wall (double lines) dotted window partition on the right wall	monitors table wall window mullions	
	23:34 23:39 23:40 23:46 23:59 24:08	table #1-2-3 trace left side of chair (4) **"those are monitors"** box line box plotter **"let's put a plotter here"** (#5) drew box on the L shape table on the right, **"a printer"** (#6) drew 3 chairs chair, chair, chair, right to left (#x, y, z)	designer pointed to the monitors drawn on last chunk	
	24:09 24:18 24:20 24:21 24:24	drew chair, chair, chair (for computers) (#1, 2, 3) drew a box (table #4) observer questioned **"needs a table to put stuff, to put plans on"**, pointing to the plotter, #5 of last chunk) drew horizontal, vertical tables (#5, 6) **"anyway, to put things"**	5:35:20 observer asked "what' that for?"	

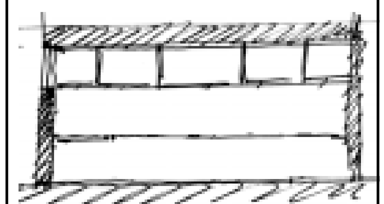
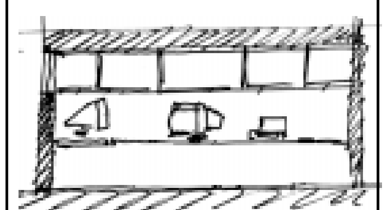
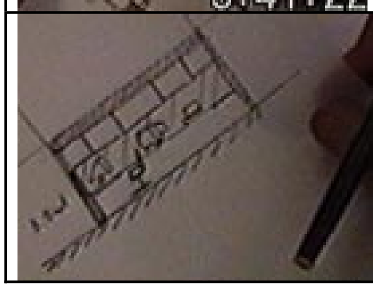
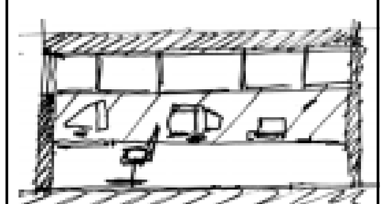
	24:31 24:40 24:50 24:55 25:10 25:20	wrote 3 4 3 3 3 (dimensions copied from underlay trace) wrote 3, 3, 1 m (on the underlay previously he had written "3 meter", and "4-5" on that relative location) arrow N 8X 23 scale 1:100 drew a little dog **"a dog is here"**	Designer has previously written on the bottom layer some dimension indicators: "3 meters" and "4-5". Now he translated the "4-5" to be "3 and "1" (meter)	

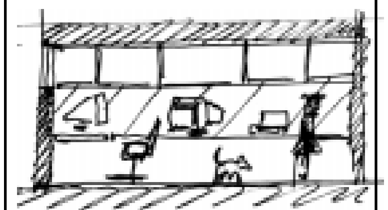
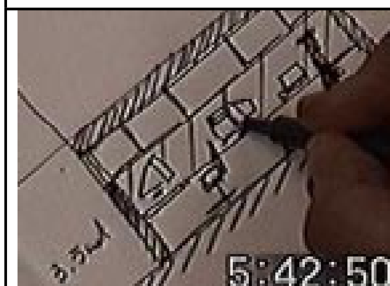
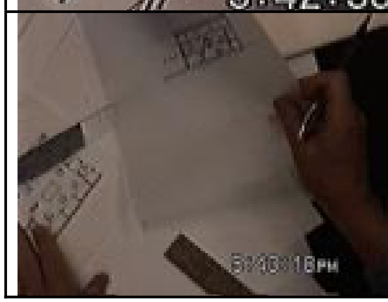
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	25:58	**"okay, next question"** started reading the next task	Task 2 - lighting (explanation) designer explained about lighting only did not engage in new design drawing	still operated on page #4 plan trace
	26:16 26:20 26:21 26:27 26:30	**"this task is doing the same thing"…. "it said lighting"** observer questioned **"I already did it" "it's from the window"** **"This is parallel processing"** wrote "parallel processing" **"Lighting come from the windows"**	5:37:20 observer asked "tell me about lighting" Designer thought that each different task would be different design. It surprised him that Task 2 has same design program except a focus on lighting.	
	26:35 36:40	drew partitions on top horizontal walls **"These windows, lighting are from north, so you got the front street light.** (left hand pointed to the front of the entrance, right hand simulated lights come in)	Therefore, he went on to justify and explain he already thought about lighting in his previous design of spatial layout. He used hand gesture to point out where the light would come from -- front entrance on the left, north windows on the top....	
	26:46	**"You got the light coming in all over the place."** (gesture of right hand fingers on top of the windows showing lights coming in)		

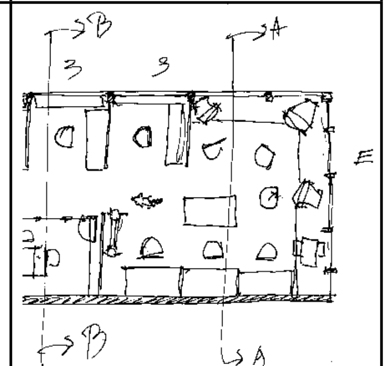
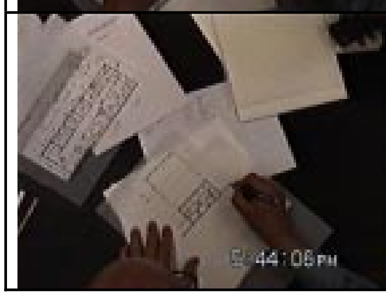
	26:49 26:51	**"So this is north"** (hands pointing the top of the paper) **"so that's north, it's good" "except these guys are going to get fried"** (hand pointing at the right side of the plan)	pointed to north then pointed to right side of the plan	
	26:56 26:57 27:02 27:03	observer questioned **"because this is east"** (wrote down word E) **"barely light in the afternoon"** observer questioned **"So, they are going to have glare, Yes, they are." "okay, you know architects, they don't care." "Give me another tracing paper"** (reached for tracing paper)	observer asked "why?" designer wrote "E" to indicate direction "East" observer asked "now what? are they going to have glare? " (remembering there are stories in the Archie case library stating computer monitors have glare because of the lighting coming in from the window)	
	27:27 27:38	**"This is task 1, and .. partial of task 2"** (wrote "Task 1 and part Task 2") **"We got most of it done" "Except er.., should I redo it?"** (arranged tracing paper to line up with underneath white paper)	Designer wrote down "Task 1 and part Task 2" to conclude this trace have involved concerns not only about spatial arrangement but also lighting issue However, designer noticed something were not done, and wondered if he should redo Task 2. Therefore, he arranged paper neatly together and getting ready for the new trace.	

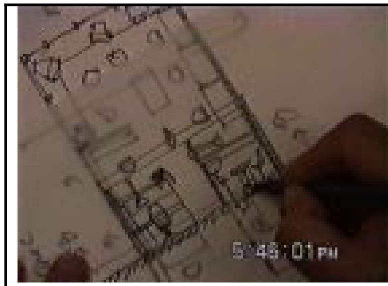
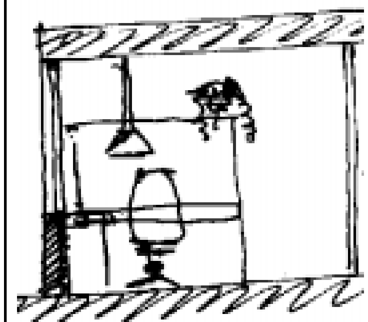
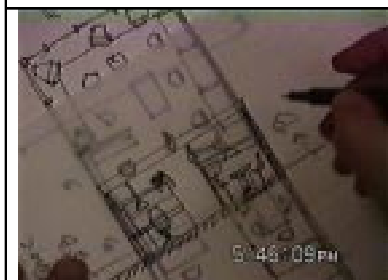
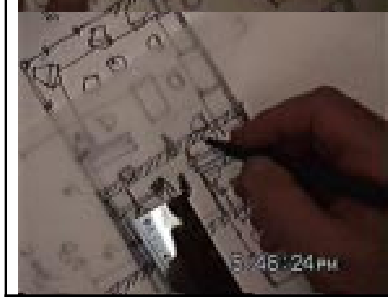
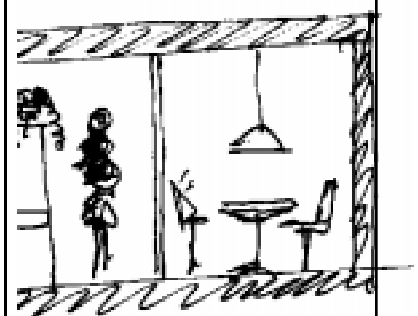
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer’s Notes & Observation	Details of Actual Drawing
	27:42	reached for new trace (page #5, section)	Task 2 - lighting (design drawing)	see page #5, sectional drawing
	27:54 28:00 28:04 28:22 28:24 28:32	**“Okay, so... we use this”** **“you can fake it, you see, because this is the wall”** drew double line on the right wall, make windows, dotted 5 times (#X, Y, 1-5) overtraced top wall, top line first (#M, N) doted 2 times for window (#a b) double lined bottom walls (#G, H)	On the previous session designer spent time explaining, justifying that he had considered lighting issue in the space layout, then he decided to redo/rework to target lighting issue. This session designer grabbed a new tracing paper and started by tracing over window mullions of the right wall, then top and bottom.	
 	28:35 28:36 28:43 28:44 28:45	drew a vertical line (section) added 2 arrows perpendicular to the vertical line **“okay so”** observer questioned **“Umm”** (while turning paper)	Vertical line and perpendicular arrows are later identified by the designer as section symbols. 5:39:44 observer asked “what are you doing, cut section? “ This cut section symbol would be repeated and drew 2 more times, 32:47, 32:52 on trace page #4 to indicate cut section A-A and B-B.	

	28:46 28:56 29:00 29:01	turn paper around (90 degree CCW, the first line become horizontal) added another horizontal line, parallel to the first line, added parallel and hatch observer questioned **“It’s a roof”**	 5:40:00 observer asked “what’s that?” the hatched walls is a roof	
	29:10 29:12 29:13 29:20	moved scale to measure, dotted **“this should be 3.5 meter”** wrote “3.5m” drew a line 3.5 below the roof drew bottom line (ground) and then drew downward hatch to indicate ground	perpendicular to the roof, designer put scale to measure the height for a story -- 3.5 meters, dotted, wrote "3.5 meters" and then drew the wall line to the ground, and added a horizontal line, and hatch beneath it	
	29:24 29:28	drew right walls and hatch in drew a left wall line and hatched it	after having finished the roof and the ground, designer went on to drew right and left wall right wall is facing south and therefore solid as program indicated left wall had windows right under the roof	

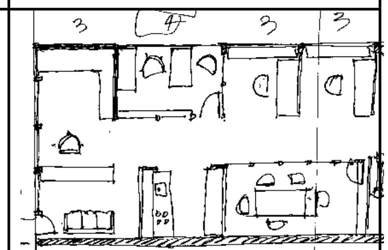
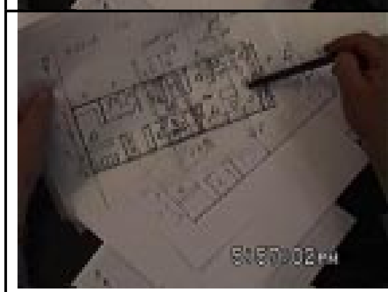
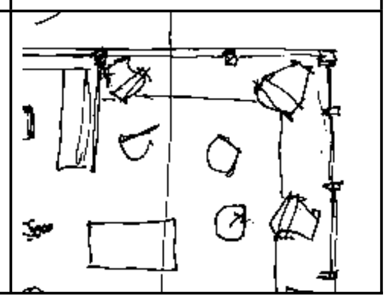
	29:30 29:33 29:44 29:47	put scale to measure (table height) drew horizontal line from measured dot (table top) darken lines the left wall, and the horizontal line (table) drew horizontal line (window) below the roof, and right under the window from the left, north facing wall	the vertical space was divided 1) height of table 2) the bottom line for windows	
	29:49 29:50 29:51	observer questioned **"window on top"**, (hand pointing to floor plan window partition dots) added window partition vertical lines according to dots drawn on plan (left hand pointed to the window partitions on the floor plan while right hand drawing vertical lines)	5:40:49 observer asked "what's that," ...."oh, I see" (acknowledged seeing those vertical mullions for windows)	
	30:05 30:15 30:20 30:30	drew table line drew a monitor drew another monitor (left hand pointed on the plan where the monitor is, right hand drawing) drew another box (printer)	table top monitor monitor printer	
	30:38 30:39 30:40 30:42	designer slashed several lighting below windows, observer questioned **"light"** **"chair"** drew a side view of a chair	5:42:39 observer asked "those are lighting?"	

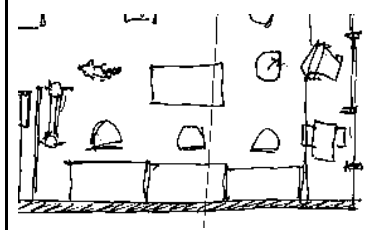
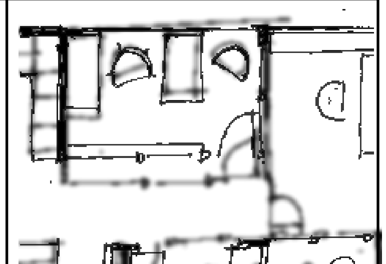
	30:52 31:00 31:09 31:20 31:24 31:33 31:37 31:39	**“I am not finished yet”** took away underlay trace and left white paper as underlay Compared the old plan trace with the current section trace drew a person **"oh, we have a person here"**. drew a dog **"and a dog**." drew horizontal bar on top of the person **“the person is wearing a hat”**	5:42:20 Observer questioned: "what’s that? 5:42:39 Observer asked "What’s that? a hat?"	
	31:43 31:55 31:59	**“okay” “this is the AutoCad menu”** drew lines on the monitor (2 horizontal, and one vertical line) **“I have to get realistic”** **“okay, so that’s computer, so what else now?”**		
	32:07 32:10 32:13 32:16	**“so that’s the lighting”** **“you can’t see, because they will get blinds,”** **“light not good, darkish, and over here”** **“here..”** reach the floor plan		

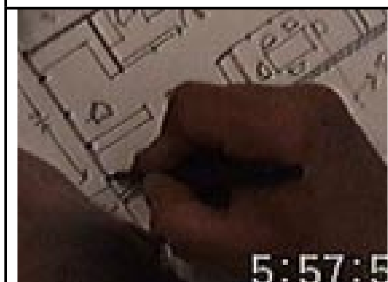
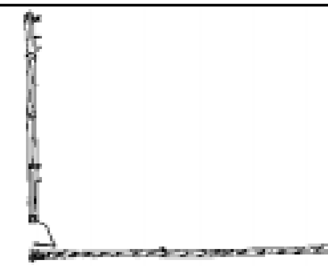
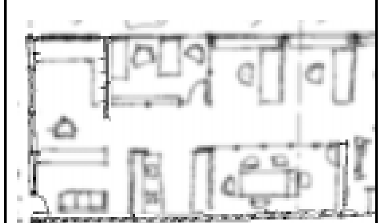
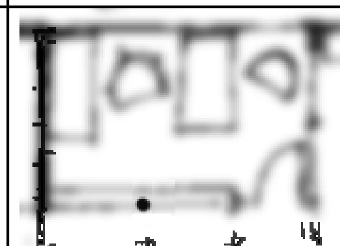
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	32:25	designer grabbed a new trace paper (page #6, section)	designer added section cut symbols on page #4 and then drew sections on page #6	see page #4, plan and then #6 section
 	32:38 32:40 32:44 32:45 32:46 32:47 32:50 32:51 32:54 32:56	**"okay, so now is the paper"** put this new trace close to section **"that's this section, where we cut from the roof"** (pointed plan) reached the tracing paper of plan (page #4) moved the plan trace on top of section drew a vertical line drew 2 arrows perpendicular to the section line, and wrote A for both observer questioned drew another vertical **"B section"** wrote B in both ends, took out the plan	Designer drew on page #4, plan 5:43:51 observer asked "so we've got A section?"	
	33:00 33:05 33:17 33:20	put the trace back on the section traced and drew the ground line & hatch under it traced over the roof line traced left walls (on top of the section trace)	drew section B-B ground, roof, left wall	

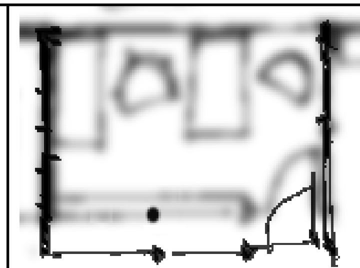
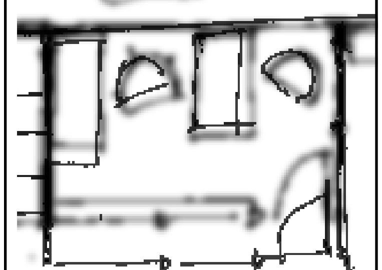
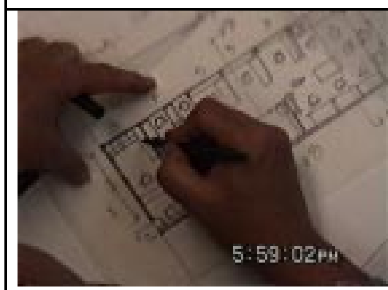
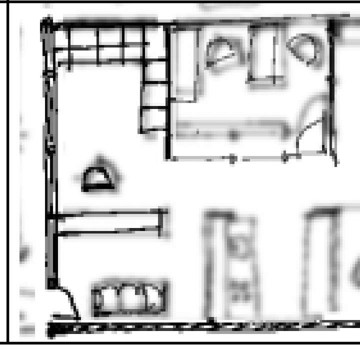
	33:28 33:42 34:05 34:07 34:12 34:18 34:23 34:34 34:47 34:50	**"so these guys won't have any light from here..."** moved plan under new section drew table section, **"doesn't fit .... well. everything in the wall"** added a chair trace left walls added top shelf drew plant on top of shelf drew light hang from ceiling drew vertical double line wall double lined chair	designer drew furniture for the designer's workspace	
 	35:01 35:02 35:06 35:07 35:13 35:15 35:23 35:28 35:30 35:41 35:48 35:49 35:52	drew a table drew a chair, another chair **"okay"** drew light hung from ceiling hatched the roof drew head for a person, **"oops, scale is all wrong"** moved scale to measure the figure **"too tall, the scale is all wrong"** **"okay, this, figure that out, this person is right now" this is too tall, sort of okay** **"now what?"** observer questioned. "**Umm,"** hatched wall **"okay, this is interesting, what else?"**	designer drew furniture for the conference room 5:46:48 observer asked "you done? ready for next task? take a break?"	

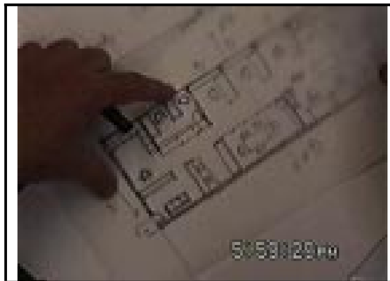
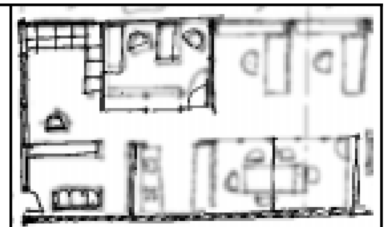
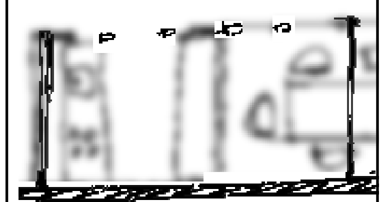
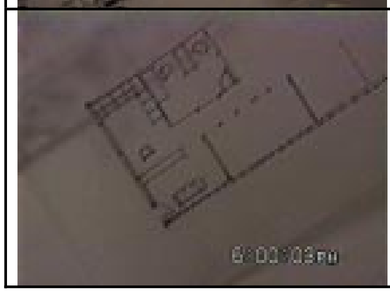
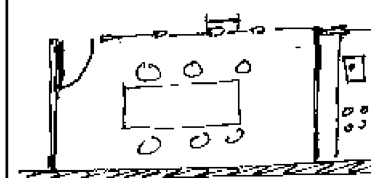
	 44:21 44:25 44:34 44:40 44:49	break played with the scale ruler, **"this is all inches, I got it."** took a new piece of paper grabbed the steel ruler, **"this is metric" "1 of 100"** **"okay, now what"**	a break is taken between task 2 & 3 observer offered to find another scale ruler for designer, but designer expressed it's sufficient that he already have a scale that have one side inches and the other side centimeters	

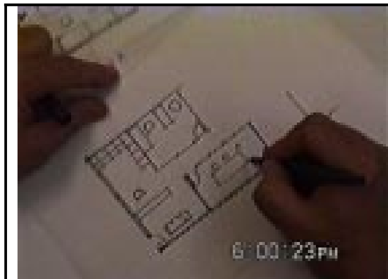
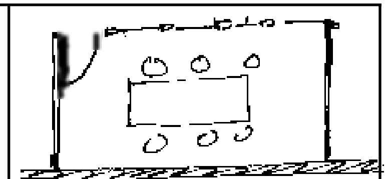
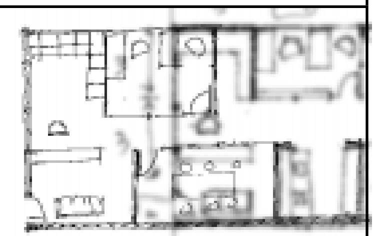
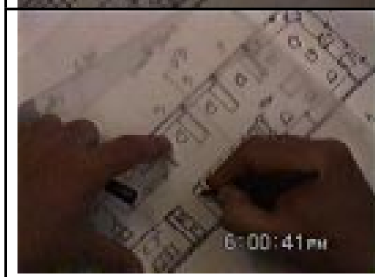
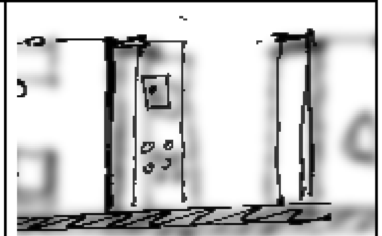
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	45:01	started up a new tracing paper (but did not draw on it)	**task 3, visibility & privacy issue**	
	45:04 45:12 45:30 45:36	**"visibility?"** searched for plan found the plan trace (page #4) **"okay, all right, visibility, well, it's all visibility"** **"all visible, it's all here, it's all solved, yes it is."** pointing at the plan	designer picked up a new trace and was ready to start task 3 however, after reading the task, he believed he already solved visibility issue in previous drawing, therefore he searched for the tracing paper that had floor plan layout and started to explain to observer how the layout worked concerning visibility issue	
	45:44 45:50 45:55	**"flow, here we got secretary here, secretary has control over the office"** **"here people weren't be bothered in the workspace, it's true"** **it's obvious, secretary can see inside, when somebody got a call, the secretary can scream down, right?"**	designer pointed the receptionist area, and explained how secretary would control visitors access and screening visitors from directly going to the working area in the back	
	46:03	"**you know people got their lighting, except for these guys, they need some screen"**	designer pointed the right corner computer monitors and chairs (the CAD stations) and explained that area (east facing windows) need to have blinds to block sun light	

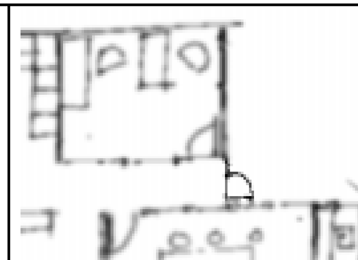
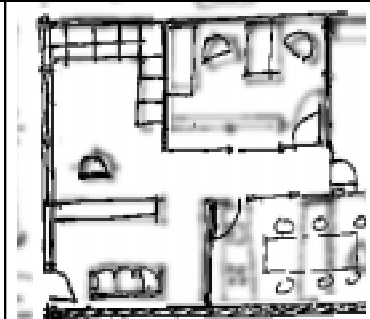
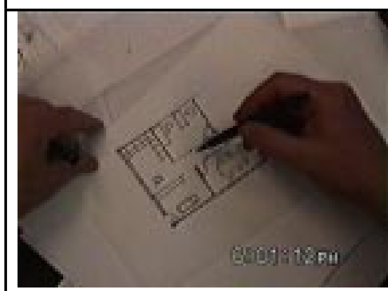
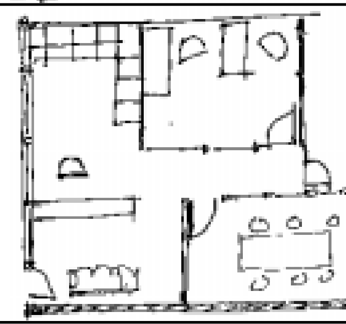
	46:09	**"students, they are lowest of the totem pole, they got this tough use dark corner"**	designer pointed to the lower 3 drafting tables and explaining that the area is the worst space (no window)	
	46:13 4:46:15	"**and this guy (architect) got the best thing here"** (pointed to the window on north) "**it's up front and"**		
	46:23	"**probably you should close it off here"** pointing possible new wall on the right of architect's office	see trace #7 for the newly added door that separate the architect's office and the work space in the back	

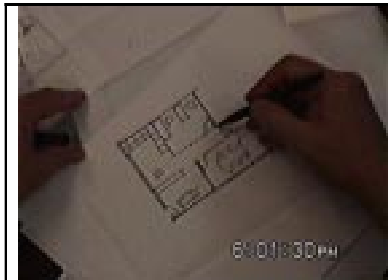
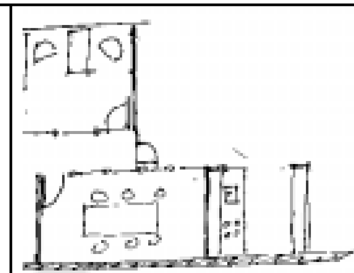
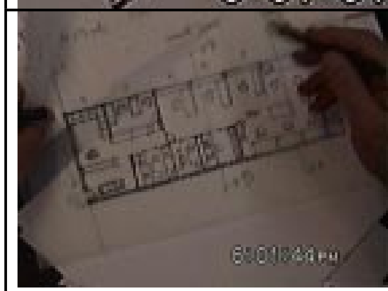
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	46:28 46:35 46:37	"**actually, actually what I will do, now we come to it"** "**we have it, we got it backwards"** lined up the new trace paper with the rest	designer put away trace (page #4) then grabbed a new trace (page #7) to put it on top of plan (page #4) "backwards" would become clear in 48:34	page #7, partial plan
	46:42 46:55 47:00 47:05	dotted on new trace the window partitions (whistled few notes), **"so we got"** added lines between window partitions added a door double lined bottom walls	designer traced over window partitions (mullions)	
	47:10 47:12	added a wall on the right of conference room overtraced the right wall of the reception area (left wall of architect's office) but made it longer	designer extended the architect's office first he extended the wall	
	47:20 47:21	dotted on the bottom window/wall of architect's office dotted 2 points lined up with the dots on the window above on the underlay and the end of the elongated wall	then dotted on the old window partition (where the windows were on underlay) then accordingly window partition end of window end of right wall	

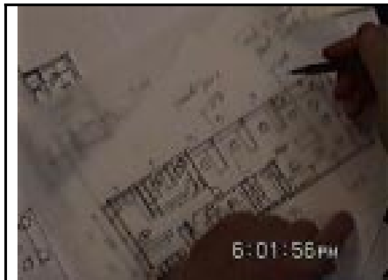
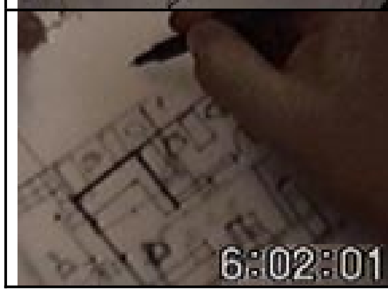
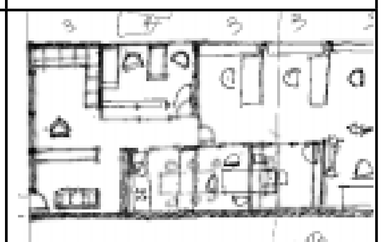
	47:27 47:27 47:33 47:38 47:39	what are you doing? **"make it bigger"** dot dot, along the right wall of architect's office, drew wall line passed the 3 dots, **"this guy, is chief, and the office, is kind of small"** added door traced over bottom wall		
	47:40 47:44 47:45 47:48 47:53	traced over table (but made it larger), **"this stays here",** then chair, table, chair top wall	the designer traced over furniture, slightly enlarging them to see if the office will still work after making the space bigger	
	47:58 48:11 48:15 48:22	"**file cabinet"** cross over rectangle "**receptionist chair"** drew a chair drew table couch	note: the design history this file cabinet and reception area was the question mark on page #3 was turned into a table/counter on page #4 and now became divided file cabinet on page #7	

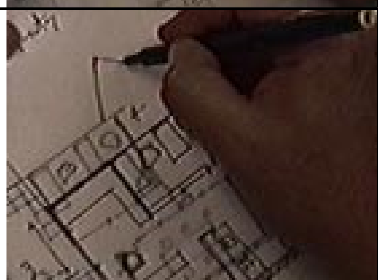
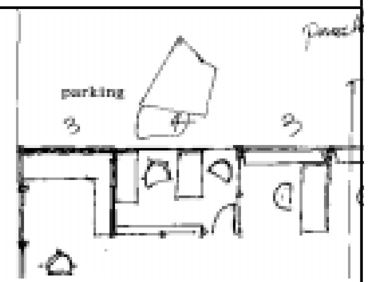
	48:26 48:32 48:34 48:40	moved trace paper dotted where the kitchen wall was from the underlay adjust wall of conference room by moving trace paper to the right, so this new dot lined up with conference room's left wall over traced the wall, **"conference room move to here"**	design decided to switch the location of kitchenette and conference room, see 46:25 when he said "we got it backwards"	
	48:44 48:56	dots 4 window partitions on the top wall hatch bottom walls	designer defined the boundary for the conference room	
	49:00 49:03	**"so we got"** move underlay tracing away the new trace landed on the white paper	designer moved out the underlay trace	

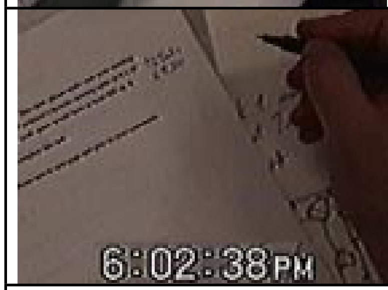
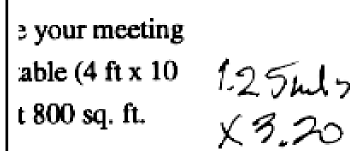
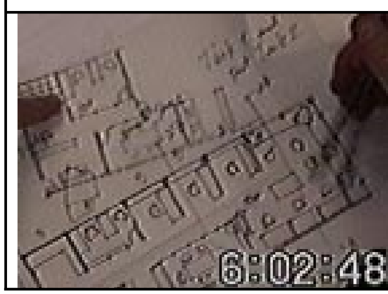
	49:05 49:13 49:16 49:22 49:26	pointed to architect's office and the lower space **"we got a door here" and** ... use the newer trace, added a door made window/wall lines between window dots drew a table chair chair chair chair chair chair (clockwise)		
	49:28 49:16 49:22 49:26 49:32	**"now let's get the kitchen on the other side"** moved the new trace and put the plan underneath lined up right side of new conference room wall with underlay kitchen's left wall **"kitchen now sits here"**	designer started rearrangement of the space	
	49:33 49:35 49:36 49:37 49:38 49:41 49:42	overtraced the left wall, and extend to the rectangle (counter) traced the top box stove traced the stoves (4 circles) **"so when visitors come in, you don't see, you don't see a kitchen,"** **see a kitchen with dirty dishes..."** traced over right counter	explaining about kitchen while tracing over kitchen counters for stoves and sink, and counter	

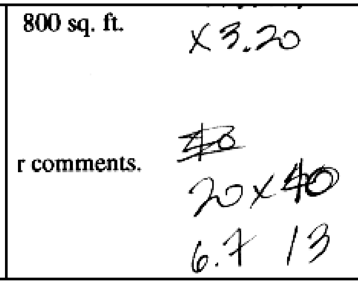
	49:45 49:50 49:55 49:56	moved trace paper, **"right?"** put another door in the middle of corridor **"let's put a door here"**		
	50:03 50:04	okay "**is that it?"** overtraced the right wall of chief architect's office		
	50:10 50:12 50:16	observer questioned **"yeah, the guys come in, the guys come in"** pointed from the left wall to the corridor t**he clients come in**	6:01:10 observed said "I want you to tell me about visual access"	

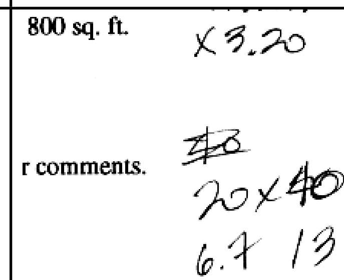
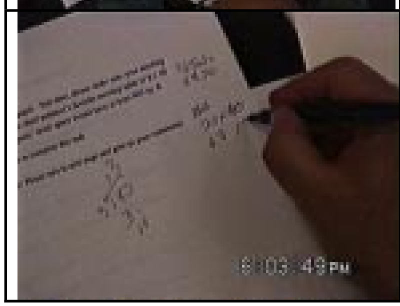
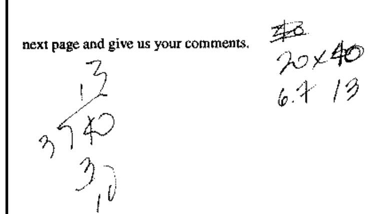
	50:17 50:24 50:25 50:28	**“and the architect comes in here, the secretary, in fact, this is all very elegant,** **“and in the back here can be a mess”** **“well that’s normal”** **“okay so, and the other guys, if they want to go to the meeting, they can come up here”**	designer used hand gestures to explain circulation and visual access 6:01:25 observer agreed "yes"	
	50:33	**“and the secretary, people here all. people make the coffee or whatever can come back here, can come inside, so you don’t see it, right?”**		
	50:43 50:43 50:45	moved trace #4 under current trace #7 **“but if there is no client, then you can leave the door open, then all these”** **these guys can all go around here**		
	50:48	**right, is that it?**	6:01:48 observer agreed "okay".	
	50:54	read the program: **“consider each group space has a fine view of the exterior”**		

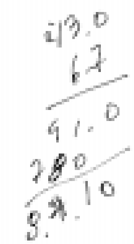
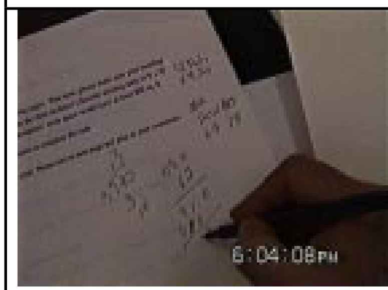
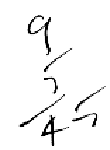
	50:57	**"yeah, they all have fine view of the exterior**		
	51:00	arranged trace paper		
	51:01	moved the underlay trace (page #4) on top of current trace (#7)		

	51:03	**"they have fine view okay"**		
	51:04	**"you see, you got this car park here, a car is here "** drew a shape	designer joked about having a car parked outside would be a view for the office	
	51:09	**"okay, next?"** moved the trace paper		

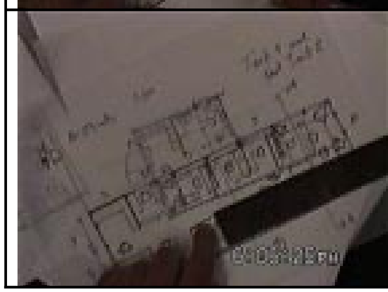
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	51:12	**"next?"** turn the program to next page	Task 4, dimensions	
	51:19	**"now what" "please start a new sheet of tracing paper. this time please make sure your meeting room is large enough to accommodate the chief architect's FAVORITE meeting table**	designer read the task 4 descriptions	
	51:26	**"this is what? 1.25 meter by 3.20 meters"** wrote 1.25 meters X 3.20	designer annotated the conversion of the 4 ft x 10 ft to be 1.25 x 3.20 meters he did the conversion calculation in his head by dividing the figures by 3.2 (?)	
	51:40	**"also, designers" "designers work space"**	designer continued to read on the second part of the Task 4	
	51:45 51:47	**okay, designers' work space,** moved trace **"these are the designer's work space here"** pointed to the three desks/cubicles	designer pointed to his drawing where the 3 desks and cubicles are 6:02:48 observed responded 'yup'	

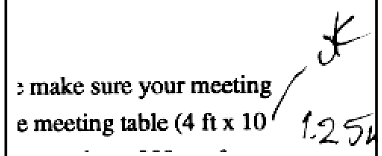
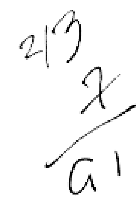
	51:53 51:54 52:00 52:01 52:03 52:05	**“800 sq. ft?”** **“how much is that?”** **“designer workspace”** **“40?”** wrote 40, and then crossed it out, **“20 by 40”** **“this is 20 by 40 meters, right,** **“20 by 40 feet?”**	after reading the dimension, designer tried to convert it to metric system	

Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer’s Notes & Observation	Details of Actual Drawing
	52:12	designer used scale to measure		
	52:34	**“this is THREE.... six point seven meter “** pointed to program and wrote “6.7”	designer wrote the number directly under the number 20, he said “three” and divided 20 by 3 and derived the answer 6.7	
	52:41 52:44 52:45 52:46 52:48 52:49	**“this is”** wrote 40 started calculation 40 divided by 3, drew divided bar wrote 1, put 3 underneath 4 wrote 10 wrote 3 by 1(result is 13) copied 13 to underneath 40	designer performed the calculation of dividing 40 by 3, and derived the answer of 13, so he wrote 13 next to 6.7 (also under the number 40)	

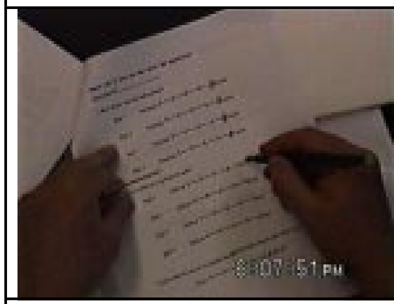
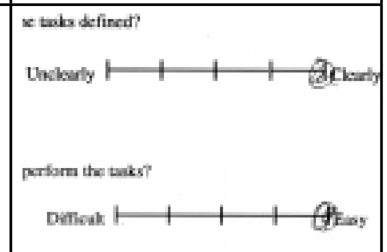
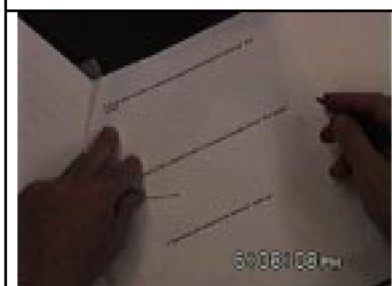
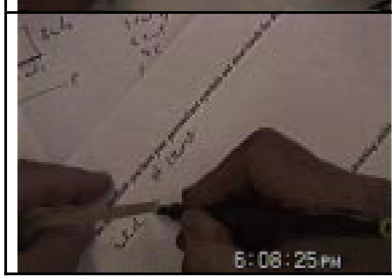
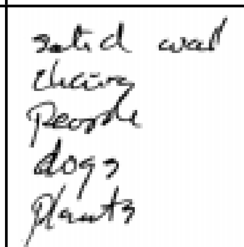
	52:53 52:55 52:56 52:57 52:58 52:59 53:00	**"6 point 7"** wrote 6.7, wrote 13 on top drew horizontal bar wrote ".0" beside 13, started calculation 13.0 (x) 6.7 wrote result, 0 . 1, wrote 2 by 13, wrote 9 from right to left (91.0) **"multiple thirteen"**	designer performed the multiplication of 6.7 and 13	
	53:01 53:02 53:04 53:05 53:06 53:07 53:08 53:24 53:28 53:29	wrote 0 wrote 8 move up by 130 and dotted wrote 7 (780) traced over "8" and horizontal bar wrote 1, pointing at the numbers wrote 7 dot 8 dot **"8 point 7 1 square meters"** **"eighty- seven square meters?"** over wrote "7"		
	53:37 53:39 53:48 53:54	**"this is 5"** **"of all the designers? I mean the people at the CAD station, or what?"** observer commented "just designers" **"designers space at least 8 hundred square feet"**	designer took the scale ruler to measure the designer work space 6:04:48 observer responded the dimension requirement is mean for designers' work space only 6:05:02 observer said "just the designers" again	
	54:04 54:10 54:11	**"this is nine"** looking at the scale and wrote "9 " on the program paper moved scale to vertical position **"this is 5"** wrote 5 underneath 9, bar, then wrote "45"	designer measured the space, calculated the dimension, and measure the space both vertically and horizontally	

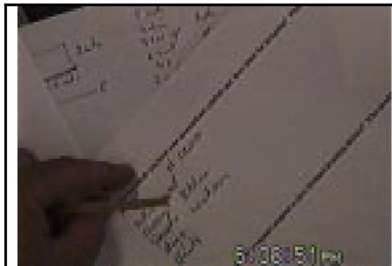
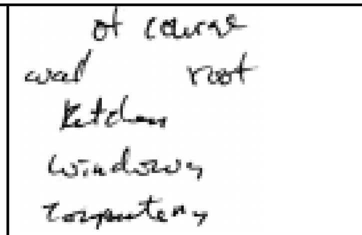
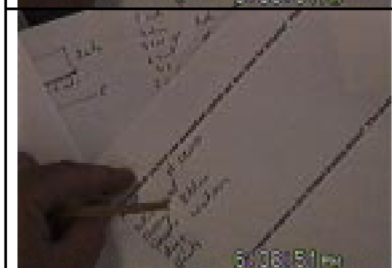
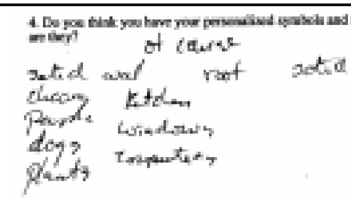
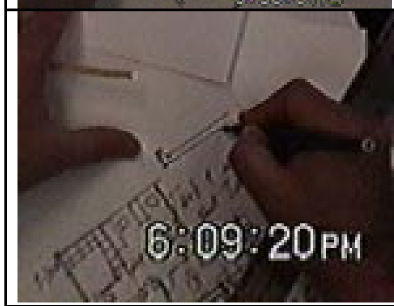
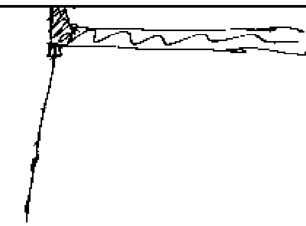
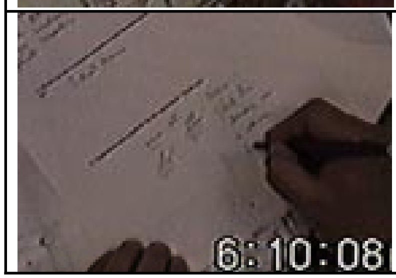
	54:27 54:28 54:29 54:30 54:33	**"I can't put them"** **"what do we got"** **"these guys already got huge amount of space"** looking at scale measure of the length **"this is ten, 10 by "**, move the scale vertical, **"six", I got it, they have enough space"**	 6:05:03 observer responded, "they got it, that's good" 6:05:44 observer said, "they got it, good"	

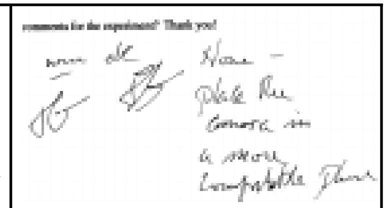
Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	54:48 54:55 54:57 55:06 55:08 55:14 55:19 55:22	put scale horizontal to measure one cubicle at 3 mm **"3"** move to vertical position **"this is 3"** **"this place is huge, they can even do roller skating"** pointed to the top space, **"yeah, I mean, work space, three by three meters?"** **"that's a big area, that's a big area, isn't it?** **"three by three, yeah it's big"** **"okay"** **"all right"**	designer used the scale ruler to check if all the space are large enough 6:06:03 observer laughed, "okay" 6:06:10 observer agreed, "3 by 3, it's cool"	
	55:28 55:29	**"okay"** moved scale to measure **"okay there, this time, the meeting room should be larger enough to accommodating"**	designer looked back and read the question	

	55:34 55:38 55:39	**"okay, one"** horizontal measure **"this is one"** vertical **"this is"** horizontal measure, **"well it's close"**	designer checked the dimensions and realized it's not exactly right, so next, he started drawing again to modify the size (described in next column)	
	55:44 55:47 55:53 55:54 55:57	overtraced the table, according to measure to make it larger **"okay"** reached to write "OK" by the program **"make the thing a little bigger"** pointing at the table **"well, I don't see why?** **"I did a conversion here"** put down the scale along the top of conference room" "**there is something wrong**"	designer draw to adjust the dimensions after adjusting the table size, he wrote "okay" by the program where the table dimension (4 ft by 10 ft) was indicated 6:07:00 observer responded, "conversion wrong?"	
	56:02 56:05 56:06 56:08 56:09	**"6 times thirteen?"** **"let me see" "thirteen?**" wrote 13 **"seven"** wrote 7, and a bar **"twenty one"** wrote 2 by 13 wrote "9" **"nine one," "ninety one square meters"** **"I don't get it** **"that's not possible" "nobody else is gonna fit anyway"**	designer checked if the dimension conversion was correct he wrote down 13 to multiple by 7 he calculated 3X7 is 21, wrote 1 under the bar, and 2 lightly on the left besides 13, and concluded the answer is 91	

	56:10 56:20 56:22	**”I don’t get it “that’s not possible” “nobody else gonna fit anyway”**	designer checking the dimensions with ruler scale again designer decided he finished the job, though a little confused about if his dimension conversion was correct, but checking with scale seemed to assure him that things work	

Captured Still from Video	Elapsed Time	Designer Actions & **Verbal Transcripts**	Observer's Notes & Observation	Details of Actual Drawing
	56:29	reached to the program next page, which is the questionnaire pages		
	56:34 56:42 56:55	**"okay now, okay, what's my id number"** designer started circling on the questionnaire responded "clearly" for all tasks description **"how difficult? "not difficult"** and continued to circle on the not difficult end of the assessment scale of how difficult to perform the tasks	6:07:36 observer responded, "forget about the ID number, just go on"	
	57:02	turned to next page		
	57:05 57:14	**"do you have any personalized symbol or short hands"** **"of course"** wrote "of course"	designer read out the questions	
	57:19 57:22 57:29 57:30 57:35 57:37	**"for example" "what should I put"?"** **"solid walls"** wrote solid **"chairs"** wrote chairs **"persons"** wrote "persons" **"dogs"** wrote "dogs" **"plants"** wrote "plants"	observer didn't recognize the writing of "solid" first and so at 6:09:06 asked what is that word	

	57:42 57:47 57:55 58:05	**"kitchen"** wrote "kitchen" **"windows"** wrote windows" **"computers"** wrote "computers" **"roof"** wrote "roof"		
	58:10	wrote "solid"	6:09:06 observer asked "what's that word" pointing to the scribble word beside "wall"	
	58:11 58:12 58:20	observer questioned **"I didn't do it here"**, reached for floor plan drew shapes on the tracing paper (page #4)	6:09:11 observer asked: "what's the symbol for roof" because the designer wrote down "roof" but in the design process that symbol was not used so the designer explained that he did not do it in the session, and so he went back to tracing paper to draw his symbol for the roof to show the observer	
	58:34 58:41 58:42 58:45 58:56 58:58	**"would like a computer to have freehand la la la?"** wrote **"I don't know"** **"well, I really don't know"** wrote "none, OK" (none in scribble form) observer asked **"none"** wrote "None"	 6:09:41 observer laughed 6:09:56 observer asked "what's that"	

	59:01 59:22	**"actually, well "place the camera in a more comfortable place "okay, we finished?""**	In the beginning of the design session, the video camera was placed on tripod right behind the designer's chair by the observer. The designer looked at the setup and suggested the observer to move the tripod higher and mount it directly on top of the table where he would be drawing. During the whole design session, the designer sat behind the table while camera tripod is high and above his drawing area. The tripod was mounted on top of the conference table of a 3.5 ft width.	

REFERENCES

Akin, O. (1986). *Psychology of Architectural Design*. London: Pion.

Akin, O., & Lin, C. (1995). Design Protocol data and novel design decisions. Design Studies, 16(#2, April), 211-236.

Asimow, M. (1962). *Introduction to Design*. Englewood: Prentice Hall.

Baudel, T. (1994). A Mark-Based Interaction Paradigm for Free-Hand Drawing. Paper presented at the ACM Symposium on User Interface Software and Technology, Marina del Rey, CA.

Belkin, N. J., Brooks, H. M., & Daniels, P. J. (1987). Knowledge elicitation using discourse analysis. International Journal of Man-Machine Studies, 27, 127-144.

Bijl, A. (1989). Computer Discipline and Design Practice - shaping our future. Edinburgh, UK: Edinburgh University Press.

Birkerts, G. (1994). Process and Expression in Architectural Form. Norman and London: University of Oklahoma Press.

Blackwell, A. F. (1997). Diagrams about Thoughts about Thoughts about Diagrams. In M. Anderson (Ed.), Reasoning with Diagrammatic Representations II: AAAI 1997 Fall Symposium, (pp. 77-84). Menlo Park, California: AAAI Press.

Boden, M. A. (1987). Artificial Intelligence and Natural Man. New York: Basic Books, Inc.

Ching, F. (1979). *Architecture : form, space & order*. New York: Van Nostrand Reinhold.

Cross, N., Christiaans, H., & Dorst, K. (Eds.). (1996). Analyzing Design Activity. New York: John Wiley & Sons.

Do, E. Y.-L., & Zacherl, A. (1993). Requirements for a Bubble Diagram Editor for Use by Architects during Conceptual Design (Archie Project Tech Report): Georgia Institute of Technology, College of Computing.

Do, E. Y.-L. (1993). Imaging the Concepts of Isovist Field in Design Process -- Topdown Isoview -- a tool for spatial analysis (Special Topics, independent study ARCH8183E). Atlanta: Georgia Institute of Technology.

Do, E. Y.-L. (1993). Transmitting Knowledge through Architectural Drawing: Phases of Conceptual Design and the Components of Diagrams for the Carpenter Center for the Visual Arts (Unpublished manualscript): Georgia Institute of Technology, College of Architecture.

Do, E. Y.-L., Or, S.-W. D., Carson, D. M., Chang, C.-s., & Hacker, W. C. (1994). *Usability Study of A Case-based Design Aid Archie* (Working Paper, GIT-CS-6752-94 GIT-CS-6752-94): Georgia Institute of Technology.

Do, E. Y.-L. (1995a). Feasibility Study of Using a Diagram Interface to Access an Architectural Knowledge Base (PhD Qualifying Paper): Georgia Institute of Technology, College of Architecture.

Do, E. Y.-L. (1995b). What's in a diagram that a computer should understand. In M. Tan & R. Teh (Eds.), CAAD Futures '95: The Global Design Studio, Sixth International Conference on Computer Aided Architectural Design Futures, (pp. 469-482). Singapore: National University of Singapore.

Do, E. Y.-L. (1995c). Visual Analysis through Isovist -- building a computation tool (working paper and computer program): Georgia Institute of Technology & Sundance Lab for Computing in Design and Planning, University of Colorado, Boulder.

Do, E. Y.-L. (1997). Computability of Design Diagrams -- an empirical study of diagram conventions in design. In R. Jungs (Ed.), CAAD Futures 97, (pp. 171-176). Munich: Kluwer.

Do, E. Y.-L., & Gross, M. D. (1997). Inferring Design Intention from Sketches -- an investigation of freehand drawing conventions in design. In Y. L. L. J. Y. Tsou (Ed.), CAADRIA, Computer Aided Design Research in Asia 97, (pp. 211-221). Taipei: Hu's Publishing.

Do, E. Y.-L., & Gross, M. D. (1995a,). Drawing Analogies -- supporting creative architectural design with visual references. Paper presented at the Computational Model of Creative Design, Heron Island, Queensland, AU.

Do, E. Y.-L., & Gross, M. D. (1995b,). Drawing Analogies: Finding Visual References by Sketching. Paper presented at the ACADIA 95, Computing in Design, enabling, capturing and sharing ideas, Seattle.

Do, E. Y.-L., & Gross, M. D. (1995c). Shape based reminding as an aid to creative design. In M. Tan & R. Teh (Eds.), The Global Design Studio, Proceedings of the Sixth International Conference on Computer Aided Architectural Design Futures, (pp. 79-90). Singapore: Centre for Advanced Studies in Architecture, National University of Singapore.

Do, E. Y.-L., & Gross, M. D. (1997). Tools for Visual and Spatial Analysis of CAD Models -- implementing computer tools as a means to thinking about architecture. In R. Junge (Ed.), CAAD Futures 97, (pp. 189-202). Munich: Kluwer.

Domeshek, E., & Kolodner, J. (1992). A case-based design aid for architecture. In J. S. Gero (Ed.), Artificial Intelligence in Design '92, . Dordrecht, Netherlands: Kluwer Academic Publishers.

Dorst, K., & Cross, N. (1995). Protocol Analysis as a Research Technique for Analysing Design Activity. Design Engineering Technical Conferences, 2(DE-83), 563-570.

Eastman, C. M. (1968). On the Analysis of Intuitive Design. In G. T. Moore (Ed.), Emerging Methods in Environmental Design and Planning, (pp. 21-37). Cambridge: MIT Press.

Edwards, B. (1979). *Drawing on the Right Side of the Brain : a course in enhancing creativity and artistic confidence*. Los Angles: J. P. Tarcher.

Eisenman, P. (1987). *Peter Eisenman House of Cards*. New York: Oxford University Press.

Ericsson, K. A., & Simon, H. A. (1984). Protocol Analysis. Cambridge: MIT Press.

Fish, J. C. (1996). How Sketches Work - A Cognitive Theory for Improved System Design. Unpublished PhD, Loughborough University of Technology.

Fish, J. C., & Scrivener, S. (1990). Amplifying the Mind's Eye: Sketching and Visual Cognition. Leonardo, 23(1), 117-126.

Flemming, U. (1987a). More than the sum of parts: the grammar of Queen Anne houses. Environment and Planning B Planning and Design, 14, 323-350.

Flemming, U. (1987b). The role of shape grammars in the analysis and creation of designs. In Y. E. Kalay (Ed.), Computability of design, (pp. 245-272). New York: Wiley.

Fraser, I., & Henmi, R. (1994). *Envisioning Architecture - an analysis of drawing*. New York: Van Nostrand Reinhold.

Gehry, F. O., & Associates; IDOM associate Architect (1997). Project Diary: Guggenheim Museum Bilbao, Bilbao, Spain - An architect's dream project in Spain's Basque Country brings worldwide attention. Architectural record., 185(Number 10), 74.

Genau, A., & Kramer, A. (1995). Translucent History. Human Factors in Computing Systems, CHI 95 Conference Companion, 250-251.

Giovannini, J. (1993). Beyond Convention. Architecture: the AIA journal., 82 (5), 52.

Goel, V. (1995). Sketches of Thought. Cambridge MA: MIT Press.

Goldschmidt, G. (1989). Architectural sketching, seeing as and seeing that : Unpublished manuscript submitted to the National Science Foundation.

Goldschmidt, G. (1991). The Dialectics of Sketching. *Creativity Research Journal*, v.4(# 2), 123-143.

Graves, M. (1977). The necessity for drawing: tangible speculation. Architectural Design, 6(77), 384-394.

Grosky, W. I., Neo, P., & Mehrotra, R. (1992). A pictorial index mechanism for model-based matching. Data and Knowledge Engineering, 8, 309-327.

Gross, M. D. (1994). The Fat Pencil, the Cocktail Napkin, and the Slide Library. In M. Fraser & A. Harfmann (Eds.), ACADIA '94, (pp. 103-113). St. Louis, MO.

Gross, M. D. (1996). The Electronic Cocktail Napkin - working with diagrams. Design Studies, 17(1), 53-69.

Gross, M. D., Zimring, C., & Do, E. Y.-L. (1994). Using Diagrams to Access a Case Base of Architectural Designs. In J. Gero (Ed.), *Artificial Intelligence in Design '94*, (pp. 129-144). Lausanne: Kluwer Academic.

Hammond, K. (1989). Case-Based Planning: Viewing Planning as a Memory Task: Academic Press.

Herbert, D. M. (1993). Architectural Study Drawings. New York: Van Nostrand Reinhold.

Hinrichs, T. R. (1992). Problem Solving in Open Worlds: A Case Study in Design. Northvale, NJ: Erlbaum.

Hix, D., & Hartson, H. R. (1993). Developing User Interfaces - Ensuring Usability Through Product & Process. New York: John Wiley & Sons.

Kahn, L. I. (1931). The Value and Aim in Sketching. T-Square Club, 1(6), 19.

Knight, T. W. (1981). The Forty-one Steps: the Language of Japanese Tea-room Designs. Environment and Planning B, 8, 213-238.

Kolodner, J. L. (1991). Improving human decision-making through case-based decision aiding. *AI Magazine*, 12(2), 52-68.

Kolodner, J. L. (1993). *Case-Based Reasoning*. San Mateo, CA.: Morgan Kaufmann.

Kramer, A. (1994,). Translucent Patches - dissolving windows. Paper presented at the ACM Symposium on User Interface Software and Technology, Marina del Rey, CA.

Kramer, A. (1995). Supporting Design Activities in the Written Medium. Human Factors in Computing Systems, CHI 95 Conference Companion, 61-62.

Landay, J. A. (1996). SILK: Sketching Interfaces Like Krazy. In T. V. Program (Ed.), Human Factors in Computing, CHI 96, .

Landay, J. A., & Myers, B. A. (1995). Interactive Sketching for the Early Stages of Interface Design, CHI '95 - Human Factors in Computing Systems, (pp. 43-50). Denver, Colorado: ACM Press.

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

Laseau, P. (1980). *Graphic Thinking for Architects and Designers*. New York: Van Nostrand Reinhold.

Laseau, P. (1991). *Architectural drawing: options for design*. New York: Design Press.

Laurel, B. (Ed.). (1990). The Art of human-computer interface design. Reading, Mass.: Addison-Wesley Pub. Co.

Lawson, B. (1994). Design in Mind. Oxford: Butterworth.

Lloyd, P., Lawson, B., & Scott, P. (1995). Can concurrent verbalization reveal design cognition? Design Studies, 16(#2, April), 237-259.

Lockard, W. K. (1973). *Design Drawing Experience*. Tucson: Pepper Publishing.

Maher, M. L. (1985). Hi-Rise and beyond: directions for expert systems in design, Computer Aided Design, (pp. 420-426): Butterworth & Co.

Maher, M. L., & Zhang, D. M. (1991). CADSYN: Using case and decomposition knowledge for design synthesis. In J. S. Gero (Ed.), Artificial Intelligence in Design 1991, Oxford: Butterworth-Heineman.

Markus, T. A. (1969). The role of building performance measurement and appraisal in design method. In G. Broadbent & A. Ward (Eds.), Design Methods in Architecture, (pp. 109-117). New York: George Wittenborn.

Matthews, K. (1994). The Great Buildings Collection. New York: Van Nostrand Reinhold.

McCall, R. J., Bennett, P., & Johnson, E. (1994,). An Overview of the Phidias II HyperCAD System. Paper presented at the ACADIA (Association for Computer Aided Design in Architecture), St Louis, MO.

McCall, R. J., Fischer, G., & Morch, A. (1990). Supporting Reflection-in-Action in the Janus Design Environment. In M. McCullough, W. J. Mitchell, & P. Purcell (Eds.), The Electronic Design Studio, (pp. 247-260). Cambridge, MA: MIT Press.

McKim, R. H. (1972). *Experiences in Visual Thinking*. Monterey, Calif.: Brooks/Cole Pub. Co.

Nakakoji, K., Reeves, B., Aoki, A., Suzuki, H., & Mizushima, K. (1995,). eMMaC: Knowledge-Based Color Critiquing Support for Novice Multimedia Authors. Paper presented at the Proceedings of ACM Multimedia '95, San Francisco.

Newell, A. (1968). On the analysis of human problem solving protocols. In J. C. Gardin & B. Jaulin (Eds.), Calcul et formalisation dans les sciences de l'homme, (pp. 146-185). Paris: Centre National de la Recherche Scientific.

Newell, A., & Simon, H. A. (1963). GPS: a program that simulates human thought. In E. A. Feigenbaum & J. Feldman (Eds.), Computers and Thought, (pp. 279-296).

Newell, A., & Simon, H. A. (1972). Human Problem Solving. Englewood Cliffs, NJ: Prentice Hall.

Oxman, R. (1993a). PRECEDENTS: Memory structure in design case libraries, *CAAD Futures '93*, (pp. 273-287): Elsevier Science Publishers.

Oxman, R. E. (1993b). Case-based design support: Supporting architectural composition through precedent libraries. Journal of Architectural Planning Research.

Peponis, J., & Wineman, J. (1995). Design Decision Support Systems (DDSS): Developing Second Generation Tools for the Analysis of Building Layout and Configuration. (Current Project). Atlanta, GA: College of Architecture, Georgia Institute of Technology.

Pohl, J., Chapman, A., Chirica, L., & Myers, L. (1988). ICADS - toward an intelligent computer-aided design system (Design Institute Report CADRU-01-88): CAD Research Unit, School of Architecture and Environmental Design, California Polytechnic State University, San Luis Obispo, CA 93407.

Preiser, W. F. E., Rabinowitz, H. Z., & White, E. T. (Eds.). (1988). Post-occupancy evaluation. New York: Van Nostrand Reinhold.

Rittel, W., & Kunz, W. (1970). Issues as elements of information systems (Working Paper 131): Center for Planning and Development Research, University of California, Berkeley.

Robbins, E. (1994). Why Architects Draw. Cambridge: MIT Press.

Rowe, P. (1987). *Design Thinking*. Cambridge: MIT Press.

Rubine, D. (1991). Specifying Gestures by Example. Computer Graphics, ACM SIGGRAPH '91 Conference Proceedings, 19(3), 225-234.

Sanoff, H. (1977). *Methods of Architectural Programming*. Stroudsburg, Pennsylvania: Dowden Hutchington & Ross, Inc.

Saund, E., & Moran, T. P. (1994,). A Perceptually-Supported Sketch Editor. Paper presented at the ACM Symposium on User Interface Software and Technology, Marina del Rey, CA.

Schodek, D. L. (1994). Critical Issues in Automation Based Creative Design Education. In A. Tzonis & I. White (Eds.), Automation Based Creative Design, (pp. 447-458). Amsterdam: Elsevier.

Schön, D. A. (1985). *The Design Studio*. London: RIBA Publications.

Schön, D. A. (1988). Designing: Rules, Types and Worlds. *Design Studies*, 9(#3, July), 181-190.

Schön, D. A., & Wiggins, G. (1992). Kinds of Seeing and their functions in designing. *Design Studies*, 13(#2), 135-156.

Schooler, J. W., & Engstler-Schooler, T. Y. (1990). Verbal Overshadowing of Visual Memories: Some Things Are Better Left Unsaid. Cognitive Psychology, 22, 36-71.

Simons, G. L. (1984). Introducing Artificial Intelligence. Manchester, England: NCC, National Computing Centre Ltd.

Sivaloganathan, S. (1991). Sketching Input for Computer Aided Engineering. Unpublished PhD, City University, Nothampton Square, London.

Stahovich, T. F. (1996). SketchIT: a Sketch Interpretation Tool for Conceptual Mechanical Design. Unpublished PhD, Massachusetts Institute of Technology, Cambridge, MA.

Stein, F., & Medioni, G. (1992). Structural Indexing: Efficient 2-D Object Recognition. IEEE Transitions on Pattern Analysis & Machine Intelligence,, 14(12), 1198-1204;.

Stiny, G. (1985). Computing with form and meaning in architecture. Journal of Architectural Education, 39(1), 7-19.

Stiny, G., & Mitchell, W. J. (1978). The Palladian Grammar. Environment and Planning B, 5, 189-198.

Sutherland, I. E. (1963). *Sketchpad: a man-machine graphical communication system.* Lexington, Mass.: M.I.T. Lincoln Laboratory.

Sutherland, I. E. (1975). Structure in Drawings and the Hidden-Surface Problem. In N. Negroponte (Ed.), Reflections on Computer Aids to Design and Architecture, (pp. 73-85). New York: Petrocelli/Charter.

Suwa, M., & Tversky, B. (1996a, 24-27 June 1996). What Architects See in Their Sketches: A Protocol Analysis. Paper presented at the Artificial Intelligence in Design ‘96, Stanford University.

Suwa, M., & Tversky, B. (1996b). What Architects See in Their Sketches: Implications for Design Tools. Proceedings of ACM Human Factors in Design '96 (CHI '96), 191-192.

Taggart, J. (1975). Sketching, An Informal Dialogue between Designer and Computer. In N. Negroponte (Ed.), Reflections on Computer Aids to Design and Architecture, (pp. 147-162). New York: Petrocelli Charter.

Thiel, P. (1961). A sequence Experience Notation for Architectural and Urban Space. Town Planning Review, 32(April), 33-52.

Ullman, D. G., Wood, S., & Craig, D. (1990). The Importance of Drawing in the Mechanical Design Process. Computer Graphics, 14(2), 263-274.

Van Sommers, P. (1984). Drawing and Cognition - descriptive and experimental studies of graphic production processes. Cambridge, England: Cambridge University Press.

Verstijnen, I. M. (1997). Sketches of Creative Discovery - A Psychological Inquiry into the Role of Imagery and Sketching in Creative Discovery. Unpublished PhD, Technische Universiteit Delft.

Wilson, T. D. (1994). The Proper Protocol: Validity and Completeness of Verbal Reports. Psychological Science: a journal of the American Psychological Society / APS., 5(5 (Sep 01)), 249-252.

Wineman, J. D. (Ed.). (1986). Behavioral Issues in Office Design. New York, NY: Van Nostrand Reinhold.

Wineman, J., & Hodges, L. (1995). Virtual Zoo through IsoVist (Current Project). Atlanta, GA: Georgia Institute of Technology.

Winograd, T., & Flores, F. (1986). Understanding Computers and Cognition - A New Foundation for Design. Norwood, NJ: Ablex Publishing Corporation.

Zeleznik, R. C., Herndon, K. P., & Hughes, J. F. (1996). Sketch: An Interface for Sketching 3D Scenes. SIGGRAPH '96, 163-170.

Zimring, C., Do, E. Y.-L., Domeshek, E., & Kolodner, J. (1994). Using post-occupancy evaluation to aid reflection in conceptual design: Creating a case-based design aid for architecture. In H. Dimitripoulos (Ed.), Design Decision Support System, (Vol. II,). Vaals, Netherlands.

Zimring, C., Do, E. Y.-L., Domeshek, E., & Kolodner, J. (1995). Supporting Case-Study Use in Design Education: A Computational Case-Based Design Aid for Architecture. In J. P. Mohsen (Ed.), Computing in Civil Engineering, A/E/C Systems '95, (Vol. II, pp. 1635-1642). Atlanta, GA: American Society of Civil Engineers.

Zimring, C. M., Wineman, J., & Carpman, J. R. (1988). The New Demand-Driven Post-Occupancy Evaluation. Journal of Architectural and Planning Research, 5: winter 88(4), 273-283.

VITA

Ellen Yi-Luen Do was born on April 7, 1965 in Taipei, Taiwan, Republic of China. She entered National Cheng Kung University (N.C.K.U.) in 1983 and obtained her Bachelor's degree of Architecture (Honors, Thesis Design Award) with a minor in Urban Planning in 1988. She received a Master's degree in Design Studies from Harvard Graduate School of Design in 1991.

Ellen received the National Outstanding Youth Award (Taiwan, R.O.C.) in 1988. She worked as a free-lance architect for design firms, as a Research Assistant for the Graduate Institute of Building and Planning in National Taiwan University, and on a special project commissioned by the Tourism Bureau of the Ministry of Communication. She received a Harvard Club of the Republic of China Scholarship in 1990. While at Harvard from 1990 to 1992, she was a Technical and Applications Assistant in the Computer Resources Department. In 1992, she entered the Ph.D. Program in Architecture at the Georgia Institute of Technology, where she taught graduate CAD and Multimedia Design courses for the Industrial Design program, using Alias, an advanced visualization and animation software package. In 1994, she won the College of Architecture Outstanding Teaching Award.

Since September 1994, Ellen has been associated with the Sundance Laboratory for Computing in Design at the University of Colorado as Professional Research Assistant and Instructor in the Environmental Design program at Boulder. She conducted research on pen based computer drawing environments and taught computer courses on such topics as graphics programming and a seminar on computer-aided design and knowledge-based design systems. She is currently a Research Associate at the Sundance Lab, working on the "Back of An Envelope" project with Principal Investigator Mark D. Gross.

Ellen is currently an elected member of the Steering Committee for the Association of Computer Aided Design in Architecture (ACADIA). Her research interests include cognitive science, human-computer interactions, drawing, sketching, diagramming, visual note taking, analogical thinking and creativity. When not hacking innovative design software, Ellen enjoys singing, free hand sketching (of course), buying books, skiing, hiking, taking photographs, traveling, watching movies, playing tuba, creating origami, eating noodles and sushi.

Ellen is also a member of AIA (American Institue of Architects), ACM (Association for Computing Machinery), IEEE (Institute of Electrical and Electronics Engineers) and AAUW (American Association of University Women).

ellen
@arch.
gatech.edu

Georgia Tech College of Architecture

Ellen Yi-Luen Do

currently valid email addresses:
ellendo@cc.gatech.edu
ellen@arch.gatech.edu
ellendo@gsd.harvard.edu
ellendo@nederland.colorado.edu
ellendo@colorado.edu
ellendo@hotmail.com

currently valid web pages:
http://www.arch.gatech.edu/~ellen
http://www.cc.gatech.edu/gvu/people/student/Ellen.Do
http://wallstreet.colorado.edu/Napkin