Application of dynamic graphics techniques to the appraisal of domestic activity spaces.

HAMMOND, B.G.

1985

To Joyce

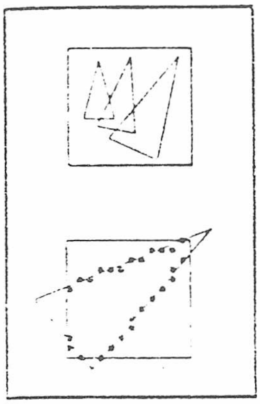
APPLICATION OF DYNAMIC GRAPHICS TECHNIQUES TO THE APPRAISAL OF DOMESTIC ACTIVITY SPACES

Brian G.Hammond; B.Sc.(Hons.) Arch.

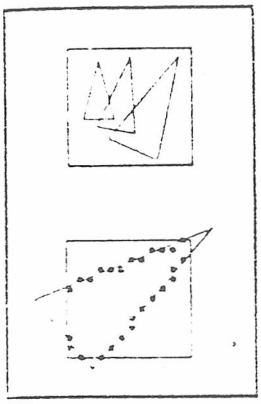
A thesis presented in partial fulfilment
of the requirements of the CNAA
for the Degree of Doctor Of Philosophy

October 1985

Awarding Establishment	:	Council for National Academic Awards
Sponsoring Establishment	:	(Scott Sutherland School of Architecture), Robert Gordon's Institute of Technology
Collaborating Establishment	:	(Department of Architecture) University of Sheffield.



CONTENTS



LIST OF FIGURES

SUPERVISION AND FUNDING

SUPERVISION

Director of Studies.

Dr. Lamond W.W.Laing, BA, M.Sc., Ph.D., ARIAS, Lecturer, Scott Sutherland School of Architecture, Robert Gordons Institute of Technology, Aberdeen.

Supervisor.

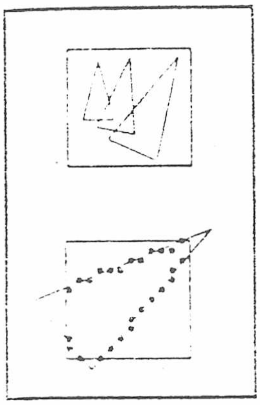
Dr. Alan H. Bridges, Dip Arch, M.Sc., Ph.D., RIBA, Deputy Director (Research), ABACUS, Research Unit, Department of Architecture and Building Science, University of Strathclyde, Glasgow.

Advisor.

Dr. Bryan R.Lawson, Reader in Architecture, Department of Architecture, University of Sheffield, Sheffield.

FUNDING

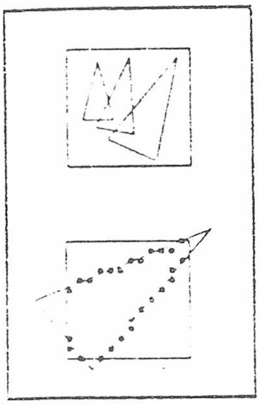
The research project was supported by a Science and Engineering Reserch Council Studentship; Ref. No. B/82301583.



ACKNOWLEDGEMENTS

My main thanks go to my supervisors for their help during this work; to Dr. Bryan Lawson who gave useful advise in the early stages; to Dr. Alan Bridges for his clear and creative criticism in his capacity as Supervisor; and most especially to Dr. Lamond Laing, who as Director of Studies, maintained daily contact with the project and whose insight and enthusiasm has permeated the whole work.

Thanks are also given to the following people and organisations; to students and staff of Scott Sutherland School of Architecture for helping in the subjective room evaluation tests; to Mrs. Sheena Falconer and the students of the School of Home Economics for their help in those same tests; to Mr. Duncan of Kitchen Studio and Mr. Tulloch of In-Toto Ltd., for their advise and assistance; and lastly to the many kitchen designers, listed in Appendix G, for their kindness in providing kitchen layouts.



DECLARATIONS

CONTENT OF WORK.

1. The candidate has not, while registered for this CNAA Ph.D. submission, been registered for another award of the CNAA or of a University during the research program.

2. None of the original material contained in this thesis has been used in any other submission for an academic award. Acknowledgements for assistance received are given under the heading acknowledgements, and any excerpt from other work has been acknowledged by its source and author.

ADVANCED STUDIES.

These included attendance at a Basic computing and programming course at the Mathematics Department of Robert Gordons Institute of Technology in the Autumn term of session 1981/82,as well as an advanced course of reading directed by the project supervisors.

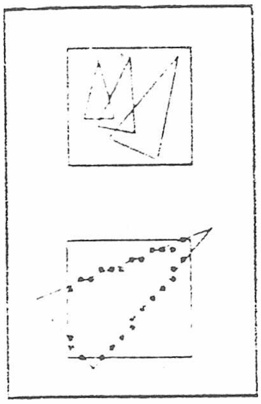
CONFERENCES AND SITE VISITS

Below are listed conferences, demonstrations, seminars and sites visited during the course of this research project:-

Dec '82	Attendance at Tektronix workshops in Livingston for 'hands on' familiarisation with 4054 terminal.
Jan '83	Attendance at ABACUS for preliminary discussions with supervisor.
Feb '83	Attendance at GABLE unit, Department of Architecture, Sheffield (collaborating establishment) to investigate capabilities of 4050's series terminals.
Mar '83	Attendance at Tektronix seminar in Aberdeen on their 4100 series terminals.
Mar '83	Attendance at internal lecture by Martin Garden of the Scottish Development Department.
Jun '83	Attendance at ROBOGRAPHICS demonstration, Aberdeen.
Jun '83	Attendance at internal lecture by Ian Hamilton of the CICA.
Oct '83	Presentation of paper at PARC 83, International Conference on Computers in Architecture.
Oct '83	Visit to Kirby Adair Newson Partnership to view their COMPUGRAPHIC DRAUGHTING system.
Nov '83	Visit to Reiach and Hall, Edinburgh to gain 'hands on' experience of their RUCAPS 8 system.
Nov '83	Attendance at GMWC Seminar to view RUCAPS 9.

May '84	Attendance at CAD 84, 6th. International Conference on Computers in Design Engineering.
Apr '85	Presentation of paper at ASSA research seminar, Edinburgh.
May '85	Attendance at ASSA Research Symposium, Edinburgh.

COPYRIGHT AND PERMISSION TO COPY THE THESIS.



NOTES ON REFERENCES, ABBREVIATIONS AND TERMINOLOGY

REFERENCES

1 Bibliographical references are placed within round brackets at appropriate positions in the text, eg. (1 1), (3.4) etc., where the first numeral refers to the chapter number and the second numeral to the reference number. These sectionalised references can be found at the end of each chapter under the heading References.

2 References to other parts of the thesis or to figures are also placed within round brackets, eg. (See Chapter 5), (See fig. 7.1) unless the direction is more aptly expressed as part of the text.

3. Where a figure from another source has been used, whether in its original or in an altered form, the word 'after' appears following the figure caption and is followed by the authors name, eg. (After Jones, E.), if need be the appropriate reference entry is given also, eg.

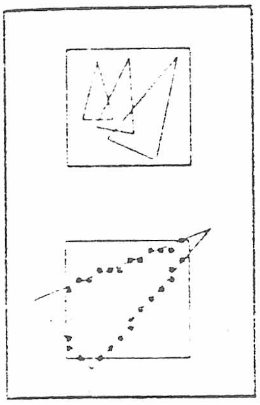
(After Jones, E. (14.5)).

ABBREVIATIONS

A number of frequently used or long named organisations etc., have been abbreviated in the text, references or in the bibliography, and these are fully described in Sub-section R.2.0, headed Abbreviations. Other abbreviations are those in common usage.

TERMINOLOGY

Novel nomenclature is defined within the text where the new concept occurs.



Brian G. Hammond
CNAA Ph.D.
October 1985

THESIS ABSTRACT

"Application of Dynamic Graphics Techniques to the Appraisal of Domestic Activity Space

The work contained in this thesis can be conveniently considered as falling into three distinct sections.

The first section investigates existing methods for man/machine communication in terms of geometry input. Utilising the results of this investigation the key possibilities for improved user interfacing using dynamic graphics were identified as:-

1. Dynamic numerical feedback for improved locational accuracy.

2. Line rubber banding for location of edges and/or vertices during primative object creation and manipulation.

3. Object translation in the x,y plane.

4. Increased use of alphanumeric feedback as a prompt and confirmation mechanism.

The second section investigates existing domestic activity space models and determines ergonomic data for a particular space activity instance (the domestic kitchen). A novel domestic activity space model is then presented based on four penalty factors:-

1. A penalty factor based on the economy of the bounding area of the activity space.

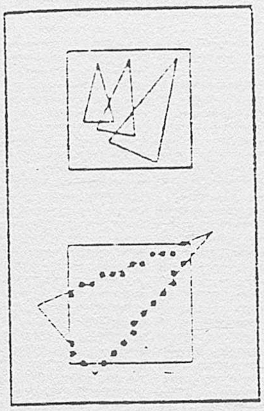
2. A penalty factor based on the economy of the enclosed volume of the activity space.

3. A weighted penalty factor based on the assessment of three dimensional overlapping of space categories (furniture and associated user areas).

4. A weighted penalty factor based on the assessment of the association distance between pairs of space category elements.

The third section represents an integration of the two sections above. A computer based implementation of the activity space model, using dynamic graphics, is presented. On the basis of comparison between subjective evaluations of a real activity space situation and a computer appraisal, the model is validated.

The system provides the architect/designer with a novel and flexible design and appraisal technique, which increases the speed and more importantly the quality of his work in designing adaptable, marginal layouts. Furthermore the system lends itself to a number of other space utilisation applications as well as forming an excellent basis for design participation.



CHAPTER 1

INTRODUCTION

CONTENTS

1.1.0 INTRODUCTION

The work contained in this thesis is concerned with the advancement of a novel model for the evaluation of domestic layouts and the consequential implementation of this model as a package of computer programs utilising refresh graphics to assist in man/machine communication.

The background to the project and its scope are as follows.

1.1.1 Background to the research project

The work was conceived as a result of the authors interest in Computer Aided Architectural Design (CAAD) applications in his days as an undergraduate studying architecture, and particularly as a result of his frustration at the benefits of CAAD being unrealised as a result of poor man/machine communication.(1.1)

As a result, the initial work of this research project consisted of a critical assessment of the extent to which existing methods of inputting building geometry (within CAAD) imposed inhibiting controls on the designers use of the available software. (1.2, 1.3, 1.4, 1.5) This assessment revealed several shortcomings. (1.6, 1.7)

1. Requirement of CAAD aids for more or different information to be input by the designer than would be required by manual methods.

2 A whole design concept is often a pre-requisite of inputting a design model.

3 Machine orientation of existing methodologies for geometry data capture.

4. Constraints imposed on the designer as to the order in which building geometry elements are created.

5. As a consequence of 1-4 above, the design process is interrupted by the computer. The designer cannot concentrate on designing since much of his attention is devoted to driving the machine.

Subsequently, input techniques were developed which made particular reference to the opportunities afforded by the use of vector refresh graphics display terminals in the alleviation of some of the shortcomings outlined above. (1.8 1.9)

1. Dynamic numerical feedback for improved locational accuracy.

2. Line rubber banding for location of edges and/or vertices during primative object creation and editing.

3. Object translation in the x,y plane.

These techniques were incorporated into a suitable methodology for the use of vector refresh graphics as an input medium for CAAD; comprising building geometry creation at three distinct levels (1.10)

1. Line manipulation.
2. Primative composition.
3. Compositional manipulations.

This methodology incorporated a mechanism which automatically created geometry primatives from a lower order of geometry elements (see Appendix E).

1.1.2 Scope of the research project

Having identified those aspects of geometry data creation and manipulation which were made easier by using dynamic graphics techniques, a particularly obvious application to emerge from the early work of this project refers to the creation and subsequent (re)location of primative geometry elements.

One area of design where this facility proved to be particularly beneficial was in the layout of furniture and equipment. Such an aid becomes especially more powerful when it is linked to rapid feedback or appraisal of the consequences of design decisions eg. whether a relocation of element(s) leads to an improvement in overall layout efficiency.

A general logical model for analysing layout efficiency was developed at Scott Sutherland School of Architecture by Langskog and successfully presented for Ph.D. examination in June 1981 (1.11, 1.12). It was proposed to develop the methodology of this appraisal logic further (the Langskog model omits consideration of certin variables which may on occassion be critical) by utilising the opportunities afforded by refresh graphics.

Thus the development of the research as a basis of a submission for Ph.D. was clarified to be as follows:-

1 To use the products of the research into man/ machine communication to extend and further explore previous research work undertaken at this school by Langskog which dealt with the numerical evaluation of the efficiency of activity spaces (rooms) utilising a methodology based on:-

   1. A weighted penalty factor based on the assessment of two dimensional overlapping areas of space categories (furniture and associated user spaces).

   2. A penalty factor based on the economy of the perimeter length of the activity space.

   3. A penalty factor based on the economy of the area of the activity space.

2. To use the concept of weighted penalty factors to revise the algorithms for activity space efficiency to overcome one or more of the following shortcomings:-

   1. Inability to assess three dimensional space category overlaps.

2. Inability to assess non-orthogonal activity spaces and space categories.

3. Inability to assess positive associations of space categories.

3. To enhance these novel algorithms by incorporating vector refresh graphics techniques to allow the rapid appraisal of the efficiency of a sample activity space eg. domestic kitchen layouts.

4 To validate the above design system utilising a carefully controlled test-bed problem.

1.1 3 Results and impact of the reserch project

The outcome of the research project is a novel CAAD system, which is relatively easy to use and understand and by combining the activities of design and appraisal within one model facilitates rapid feedback during the design process.

The program package has, as yet, had minimal use outwith this research project (1.13)

Within the project, the program validations have shown that the main objective of improvement in the evaluation of design proposals has been met. Given the high quality of man/machine communication acheived through the use of refresh graphics significant improvement in the synthesis of design alternatives should be possible.

The refinements discussed in Chapter 8, Section 8.2.0 would allow commercial application of the model in practice as a novel design aid.

1.2.0 GUIDE TO THIS THESIS

This section gives an indication of the contents of this thesis and is intended as a guide to the reader who wishes to be selective.

Chapter one (this chapter) gives a brief insight to the background of the research project, as well as indicating the duality of its nature. It also contains this guide and gives a brief summary of the results and impact of the research project.

Chapter two examines the data base for the test-bed situation of the domestic kitchen which will be used experimentally in the validation of the model. Not only does it consider ergonomic and associative data but it demonstrates how this data was converted to a form suitable for use by the computer model.

Chapter three considers the reverse side of the coin as regards this research work, ie. user interfacing. It appraises some existing hardware/software configurations primarily as regards geometry manipulation.

Chapter four returns to the model. It examines, briefly, some existing facilities planning techniques both manual and computerised before examining two domestic space appraisal systems in more depth. It then presents the mathematical basis of the new model's constituent parts.

Chapter five begins where chapter three leaves off. It integrates the two halves of this project. It examines alternative modes of display and the possibilities afforded by refresh graphics to the user interface. It then indicates how the computer implementation benefited from some of these possibilities.

Chapter six describes the course and outcome of three seperate validation experiments and presents the conclusions thereof.

Chapter seven presents the computer implementation of the model and its associated graphical manipulation segments in the form of a mini user manual. It also identifies some improvements in the implementation that are possible and desirable within the current configuration.

Chapter eight is in three sections. The first section examines the possibilities for improving the program implementation. The second section discusses possible applications for the algorithm in its present form. Whilst the third section is concerned with the possible usefulness of the algorithm in conjunction with the emergent fields of expert systems and artificial intelligence.

Section R contains a short bibliography of particularly useful sources, together with a list of abbreviations used in this thesis.

Appendix A is an unpublished paper concerning the user interface.

Appendix B is a published paper (Proceedings of PARC '83) which explores the use of refresh graphics within an 'intelligent' computer environment with regard to geometry manipulation.

Appendix C is an unpublished paper which expands on the above theme.

Appendix D was presented at an ASSA CAAD seminar, Edinburgh 1985 and represents a position report on this research project at that time.

Appendix E examines in detail an algorithm devised by the author for extracting enclosed areas from within a lattice of line segments.

Appendix F examines in detail an algorithm devised by the author for determining an ordered data set describing the bounding edges of the area of overlap between two enclosed polygons.

Appendix G acts as supplementary text to Chapter 6.

Appendix H presents a manual check on the numerical accuracy of the computer implementation of the model.

Appendix I is a pointer to further items of interest concerning this research project but not included within the body of this thesis.

1.3.0 REFERENCES

(1.1) Laing, L.W.W., et al., 'Enigma; an Enhanced Interpretive Graphics Module for Architects', Occassional Paper No. 10, Scott Sutherland School of Architecture, RGIT, Aberdeen, 1983, p.1

(1.2) Hammond, B.G., Techniques for the Input of Geometry Data in Computer Aided Architectural Design: Report No. 1 (unpublished), Scott Sutherland School of Architecture, RGIT, Aberdeen, March 1983.

(1.3) Hammond, B.G., Techniques for the Input of Geometry Data in Computer Aided Architectural Design: Report No. 2, (unpublished), Scott Sutherland School of Architecture, RGIT, Aberdeen, July 1983.

(1.4) Hammond, B.G., 'Techniques for the Input of Geometry Data in Computer Aided Architectural Design: Report No. 3', Occassional Paper No. 11, (unpublished), Scott Sutherland School of Architecture, RGIT, Aberdeen, October 1983.

(1.5) Hammond, B.G., Leifer, D., 'A Graphics Interface to Complement Traditional Input Techniques', PARC 83 Proceedings, Online Conferences Ltd., Pinner, 1983.

(1.6) Reynolds, R.A., Computer Methods for Architects, Butterworth, London, 1980.

(1.7) Newman, W., 'Some Notes on User Interface Design', Methodolody of interaction - IFIP Workshop, (Ed.) Guedj, R.A., North Holland Publishing Co., Amsterdam, 1980.

(1.8) Tektronix, 4054 Options 30 and 31 Dynamic Graphics Operators Manual, Tektronix Inc, Oregon, USA, October 1981.

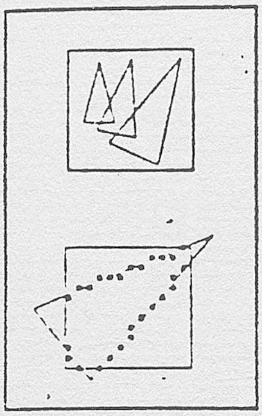
(1.9) Newman, W., Sproull, R., Principles of Interactive Graphics, McGraw Hill, New york, 2nd Edition, 1979.

(1.10) Hammond, B.G., 'Feedback Aids to Geometry Data Input', Occassional Paper No. 4, (unpublished), Scott Sutherland School of Architecture, RGIT, Aberdeen, March 1983

(1.11) Langskog, K., Laing, L.W.W.,'A Computer Aided Architectural Design Technique for the Appraisal of Domestic Activity Spaces', CAD 84 Proceedings, Butterworths, Guilford, 1984.

(1.12) Langskog, K., A Computer Aided Architectural Design Technique for the Analysis, Synthesis and Appraisal of Domestic Activity Spaces and House Plans, (Ph.D. Thesis), Scott Sutherland School of Architecture, RGIT, Aberdeen, 1981.

(1.13) Elliot, T.P., Space Requirements and Functional Constraints Affecting the Design of Medical Installations on Offshore Drilling Rigs, ongoing Ph.D. research, Greys School of Art jointly with Scott Sutherland School of Architecture, RGIT, Aberdeen, registered 1984.



CHAPTER 2

INVESTIGATION OF FUNCTIONAL ASPECTS OF DOMESTIC KITCHEN LAYOUTS

CONTENTS

2 1 0 INTRODUCTION

The investigation of functional aspects of domestic layouts is of central importance to this research project.

No attempt was made to determine, by experiment, the space requirements for individual furniture elements since this information has been published by others better funded and equipped for this type of work.(2.1)(2.2)(2.3)(2.4)

Since the scope of this project restricted investigation to that of a single test-bed situation, that of the domestic kitchen, only information relating to that situation is presented in the following sections.

2.2.0 KITCHEN ACTIVITY SPACES AND FUNCTIONS

This section summarises the findings of the literature search relating to kitchen activity spaces and functions.

2.2 1 Effect of household type on kitchen requirements

It is a trueism that different users have different kitchen requirements. Each individual is unique, and an optimal solution for one user may be totally unsuitable for another. For example, compare the requirements of a left-handed as opposed to a right-handed user.

Obviously such considerations are beyond the scope of the general room analysis model presented later in this thesis. However, it is important to be aware of some of the major variables subject to change dependant on household size, location and/or character.

1. Food storage.
   Large households, households remote from shopping facilities, households buying in bulk or making large quantities of jams and preserves and households who share kitchens but not housekeeping, such as is found in non-selfcontained flats, are likely to require more food storage than other households.

2. China and glass storage.
   The quantity of this type of storage depends not only on household size but on other factors such as the degree of entertaining and whether

seperate glass and china is retained for special occassions. Also, older households have had time to accumulate large amounts of china.

3. Utensil storage.
   Basic cooking equipment is very much the same irrespective of household size. Larger households have larger pots and pans but not necessarily more of them. However, there is a great deal of extra equipment available for those who are interested and can afford it.

The designer using the space activity model should keep these 'user specific' requirements in mind since the computer model is only capable of determining a basic efficiency.

2.2 2 The room as a whole

Varying degrees of separation between working and dining areas produce different types of kitchen.

1. Working kitchen with seperate dining area.
2. Working kitchen with associated seperable dining space (distinct seperable areas for working and dining within a single space).
3. Dining kitchen.

This is illustrated graphically below:-

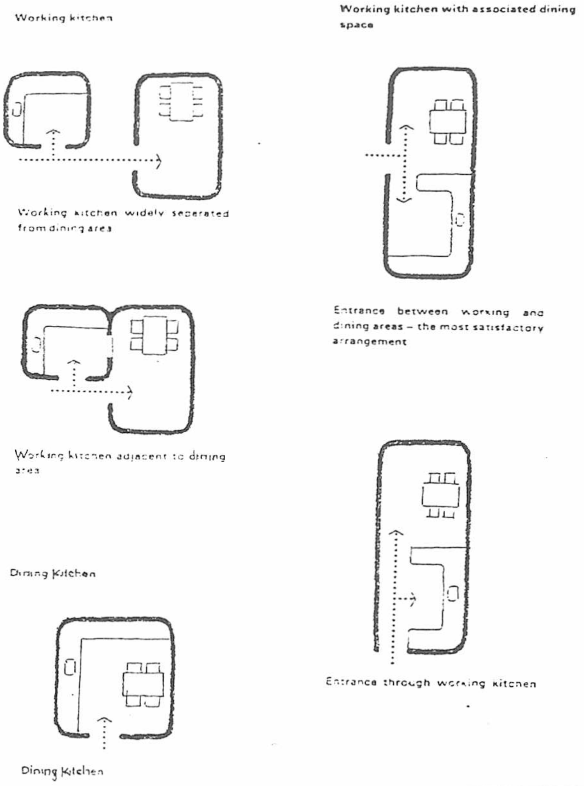


Fig.2.1 Kitchen Classification (after DOE)

> 'None of these kitchen types is better than any other. They merely represent different degrees of seperation between meal preparation and other activities.' (2.5)

Since the room analysis model as implemented is restricted to consideration of a single room, kitchens selected for testing (see chapter 6, appendix G) were either working kitchens or working kitchens with associated seperable dining spaces. In the latter case, the associated seperable dining areas were discarded leaving only the working kitchen element. The method for achieving this is described in the next section - 2.3.0.

Again some aspects of designing kitchens in relation to the remainder of the house escape the scope of the computer model.

1. Natural light
   Although sufficient light for working can be provided artificially, there is evidence that housewives prefer some natural daylight. (2.6)(2.7)

2. Access to the main entrance door.

3. Access, not through a living room, to a store for refuse not more than 6m away.

4. Access to private open space for supervision of children and/or clothes drying.

5. A pleasant view out?

This type of design factor might be incorporated into a wider building appraisal model. This is speculated upon in a later chapter of the thesis.

2.2.3 The meal preparation process

This sub-section discusses the design implications of the meal preparation process.

The kitchen houses a series of activities which are closely interrelated. A crude representation of these activities is given in the figure 2.2 over.

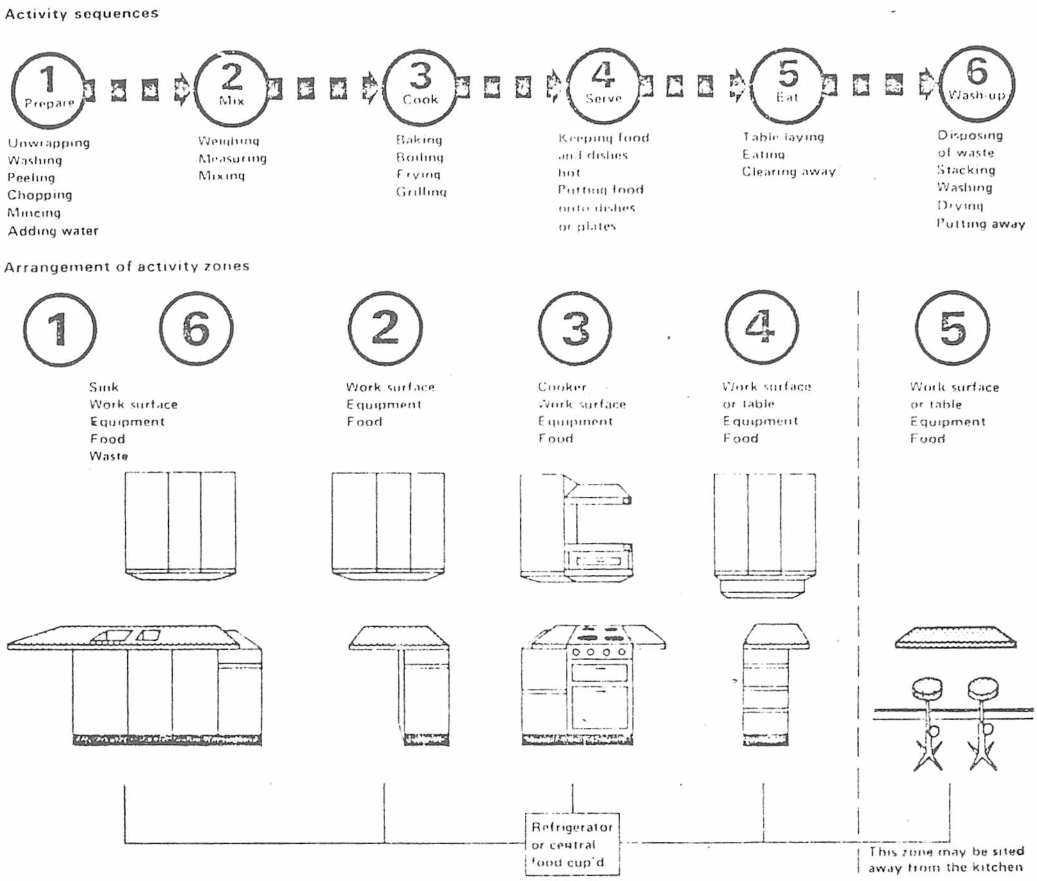
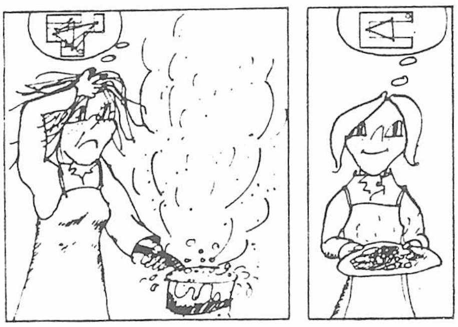


Fig.2.2 Activity Sequences(after DOE)

Without going into the details of each individual activity, it is clear that each activity zone bears some relationship to the other activity zones. It is also noticeable that certain activities are centred around particular furniture elements. For example, washing up and preparation are centred around the sink, whilst cooking is centred on the cooker.

One way of assessing how efficiently a kitchen has been planned is to measure the distance the housewife travels around the kitchen in a given time or for a particular task. It has been found that a quick way to assess the general efficiency of a kichen is find out how far the housewife has to walk between the sink, cooker and the refrigerator or central food store - the 'work triangle'.(2.8)


"... how far the housewife has to walk..."

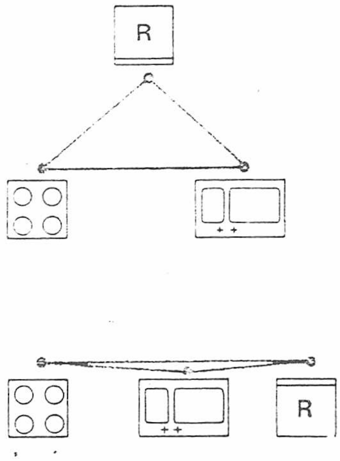


Fig.2.3 Work Triangle (after DOE)

The total distance between these three elements (the length of the sides of the work triangle) should be between 3600mm and 6600mm (2.9) to give adequate working space and yet remain reasonably compact. In addition the distance between cooker and sink should be between 1200mm and 1800mm long.

The major appliances (cooker and fridge) together with the sink are usually planned with regard to this work triangle. In addition certain simple rules apply to each of them.(2.10)

1. Sinks

   1. Do not place a sink in the corner or beside a tall unit.

2. Allow at least 100mm, preferably 300mm, from a corner with other units.

3. There should be space for a person to dry the dishes at the drainer by the sink.

4. If there is a single drainer, it should be on the side away from the cooker.

5. The surface on the cooker side of the sink should be about 900mm long.

2 Cookers

1. Do not put a cooker at a window.

2. Do not put a cooker in a corner because at least 300mm is needed to allow the cook to stand comfortably while cooking.

3. Keep the cooker away from the end of a run of units, circulation routes, doorways and where a swinging door could hit the cook or the cooker.

4. Do not put the hob under shelves or a cupboard, it would be a fire risk to someone reaching over the hob.

5. Allow 300mm on both sides of the hob for setting down hot pans.

6. Gas cookers should not be in a draught that might blow out the flame.

3. Fridges

1. Units above worktop level should not interrupt the basic sink-cooker sequence.

2. Upright fridge and freezer doors must be opened 130 degrees to remove shelves or pull out baskets, so allow 100mm on the hinge side for this.

4. Other appliances

1. Plumbing and hot water requirements will mean that washing machines are usually near the sink.

Many of these rules find expression in the room analysis model, within the remit of the association penalty. This is explained more fully in the next section.

2.2.4 Meal preparation areas

This sub-section examines the space requirements of each individual activity particularly as they relate to individual fitments.

Fixtures are covered by dimensional coordination standards (2.11), and by limitations in the manufacturers' ranges.

Figure 2.4 over illustrates the dimensional limitations of most kitchen fixtures. In general, most appliances, cookers, fridges, etc., are designed to coordinate with the remaining kitchen fitments, either by approximating the dimensions of a base unit or by being housed in a housing unit which is dimensionally coordinated.

In the following pages, a series of figures and diagrams demonstrate the space required for the use of each item of furniture; but before that, there is a diagram explaining the draughting conventions that have been used, see fig. 2.5.

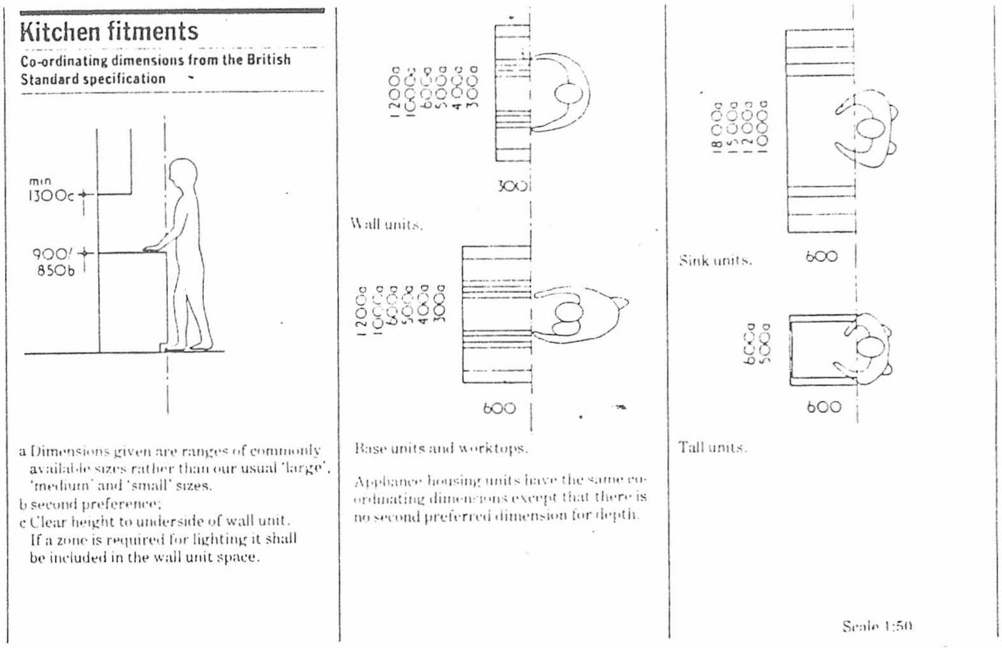
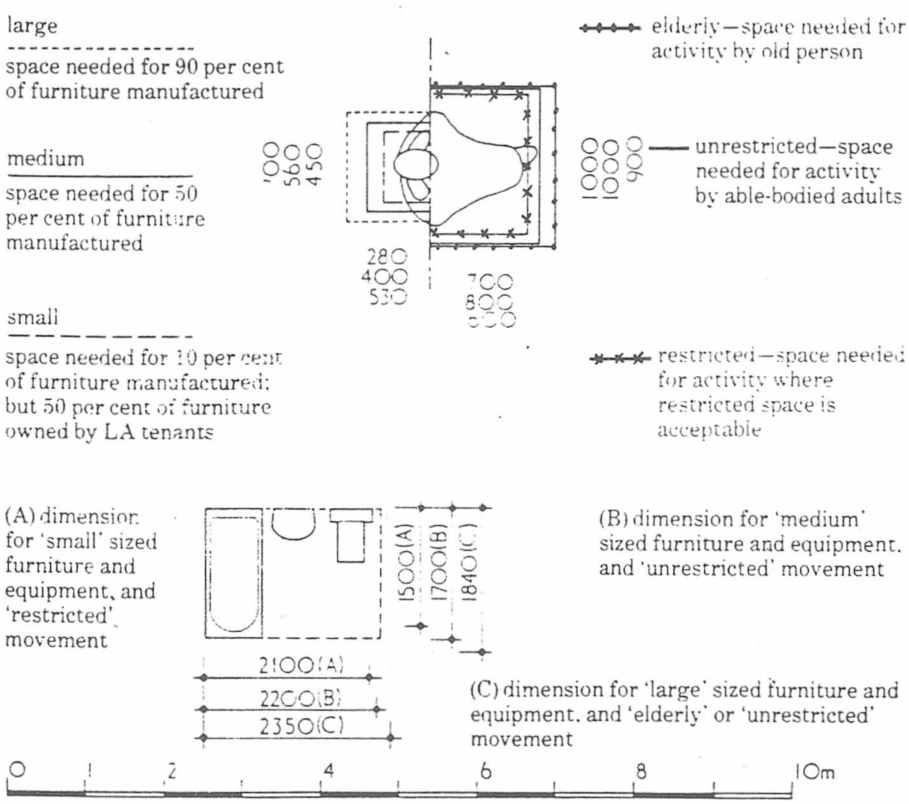
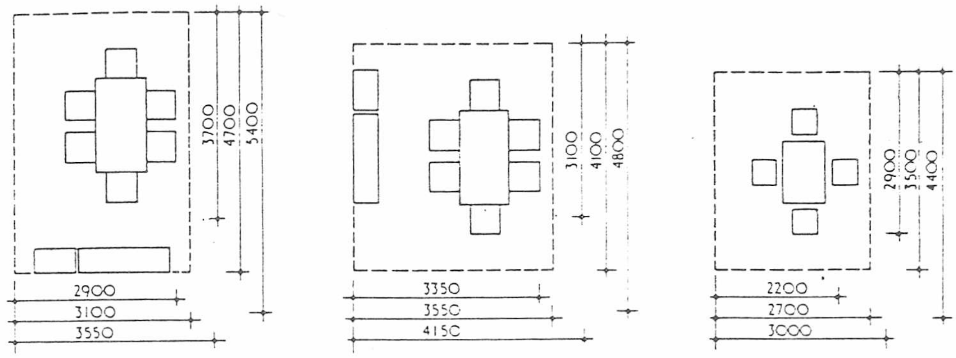


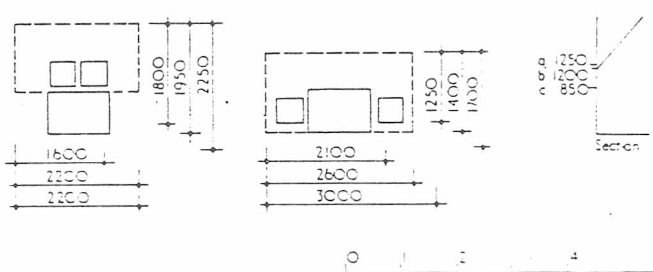
Fig.2.4 Kitchen Fitments (after Architects Journal)

Fig.2.5 Drafting Conventions (after Architects Journal)



Dining areas





a Maximum sideboard.
b Minimum headroom at back of 600 mm wide kitchen table placed against wall. 50° min ceiling angle.
c Maximum kitchen table.

Typical layouts of dining areas. Space for access to sideboards has been derived from the kitchen storage access data. Space has been allowed for circulation around tables.

Dining and kitchen chairs

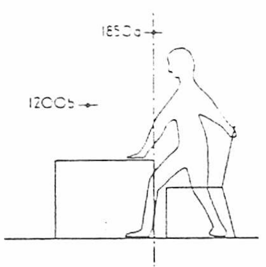
FIRA survey data: 100 items—dining chairs

Per cent	*Width (mm)*	*Depth (mm)*	*Height (mm)*
100	670	610	
90	610	560	
75	550	560	760-1290
50	490	540	
10	460	470	

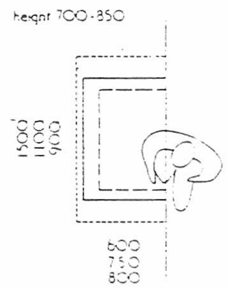
FIRA survey data: 14 items—kitchen chairs

Per cent	*Width (mm)*	*Depth (mm)*	*Height (mm)*
100	500	480	
90	460	460	
75	430	430	770-940
50	430	430	
10	360	430	

Dining and kitchen tables



a Headroom at edge of table.
b Headroom 600 mm from nearside of table.

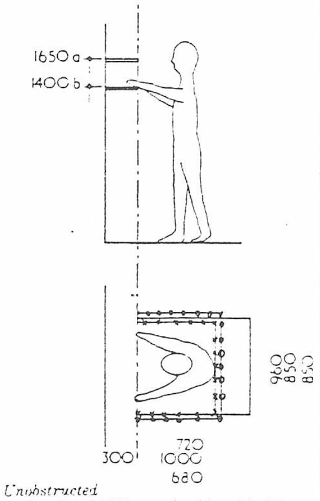


ACTIVITIES AND SPACES

Elderly
Unrestricted
Restricted
Scale 1:50

Fig.2.6 User Requirements (after Architects Journal)

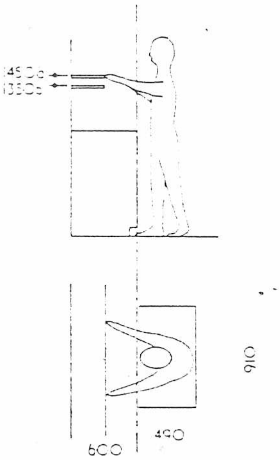
High shelves and cupboards



Unobstructed

a A height of 1650 mm should enable 95 per cent of the general population to reach items at the front of the shelf.

b A height of 1400 mm should enable 95 per cent of the elderly population to reach with two hands items at the back of a 300 mm deep shelf.

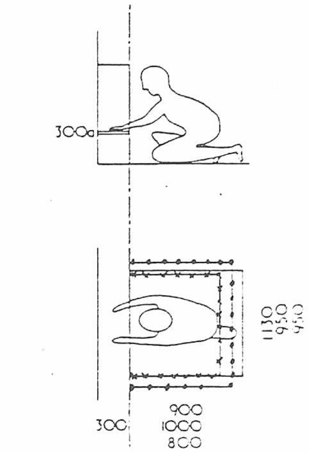


Obstructed

a A height of 1450 mm should enable 95 per cent of the general population to reach items at the front of the shelf.

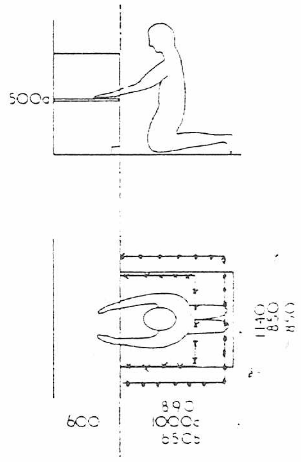
b A height of 1350 mm should enable 95 per cent of the elderly population to reach with two hands items at the back of a 300 mm deep shelf.

Low level cupboards



Shallow cupboards

a For the elderly this shelf should be not less than 300 mm above the floor level to allow for full use with both hands.



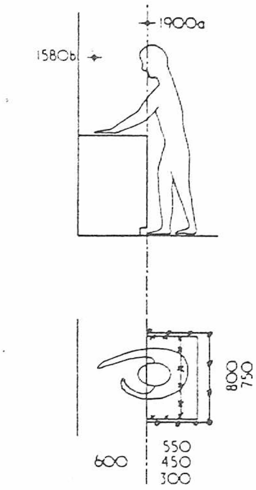
Deep cupboards

a Deep shelves at low level should be avoided for the elderly whenever possible. When necessary the shelf should be not less than 500 mm above floor level. (Even then most elderly people will only be able to retrieve items with one hand.)

b For pull-out racks the restricted depth would be 750 mm and the width 1000 mm.

c Suits pull-out racks but width would be 1000 mm.

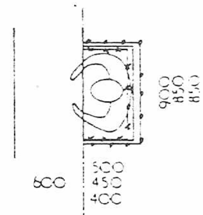
Worktops



Standing

a At front edge of unit.

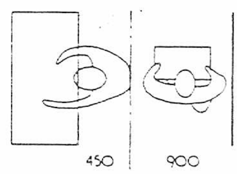
b 490 mm from front of unit—when reaching to back of worktop.

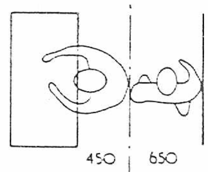


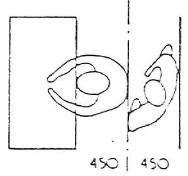
Sitting

See 'Dining and kitchen tables' for space required to get up and sit down.

Circulation in the kitchen

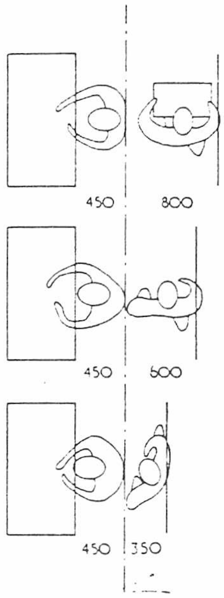






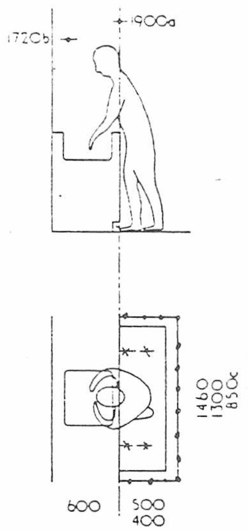
Distance between a person standing or sitting at a worktop and a wall or tall unit.

Fig.2.6 Cont.



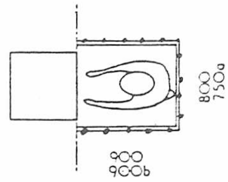
Distance between a person standing or sitting at a worktop and a table or base unit.

Sinks



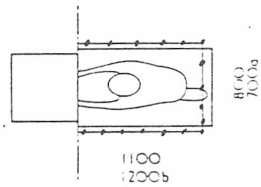
a At front edge of unit.
b 450 mm from front of unit when reaching to clean back of sink.
c Using only the sink bowl.

High level ovens



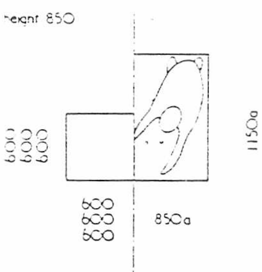
a An overall width of 1000-1200 mm is needed to use both the oven and adjoining setting down surface.
b The same depth is required for both side- and bottom-hung doors.

Low level ovens



a An overall width of 1000-1200 mm is needed to use both the oven and adjoining setting down surface.
b A depth of 1000 mm would be adequate for ovens with side-hung doors.

Dishwashers

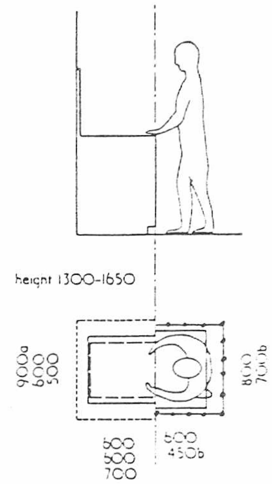


a Dimensions are for side-loading, which most people prefer. The space for front-loading is 775 × 1200 mm.

FIRA survey data: 10 items (floorstanding)

All models would fit within the 600 × 600 mm space.
All were 850 mm high to fit under worktops.
See notes on cookers regarding BS recommended dimensions for appliances, (right).

Cookers



The British Standard specification[illegible] for domestic kitchen equipment recommends co-ordinating dimensions for spaces to accommodate appliances and appliance-housing units.
A plan space 600 × 600 mm is the preferred dimension, but the depth excludes any space that may be required for horizontal services. It is assumed that in future most freestanding appliances and built-in units will fit within this space.
Built-in ovens range in width from 550-700 mm and in depth from 550-630 mm.
a This width would allow for models with two ovens side by side.
b When using a cooker hob a depth of 650 mm and a width of 750 mm are needed in front of the cooker in order to use a high level grill at a height of between 1250 and 1350 mm.

FIRA survey data: 22 items—freestanding*

Per cent	*Width (mm)*	*Depth (mm)*	*H[illegible]*
100	900†	690	
90	610	690	
75	550	690	1300-16[illegible]
50	510	600	
10	500	560	

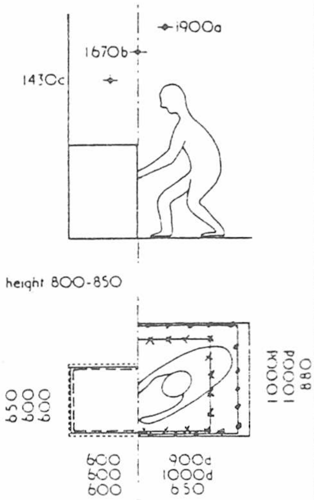
* 10 gas and 12 electric.
† This width would allow for models with two ovens side by side.
‡ The highest are those with eye-level grill or top ovens.

ACTIVITIES AND SPACES

- Elderly
- Unrestricted
- Restricted
- Scale 1:50

Fig.2.6 Cont.

Washing machines



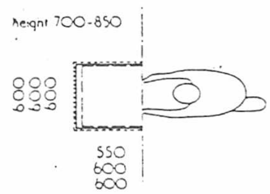
A space 800 mm wide × 500 mm deep × 750-800 mm high should accommodate most twin tubs: in a FIRA sample of five models, the largest dimensions were 830 mm wide × 521 mm deep × 830 mm high.
A space 420 mm wide × 420 mm deep × 750-800 mm high should accommodate most single tubs: in a FIRA sample of three models the largest dimensions were 420 mm wide × 425 mm deep × 757 mm high.
a 250 mm in front of machine.
b At front edge of machine.
c 240 mm from front of machine.
d 1200 mm width × 1100 mm depth would be needed with a laundry basket placed on the floor in front.

FIRA survey data: 20 items—front-loading automatics*

Per cent	Width (mm)	Depth (mm)	Height (mm)
100	800	650	
90	650	600	
75	600	570	760-850
50	590	560	
10	450	420	

* Top-loading automatics plan dimensions were similar to front-loading automatics. The tallest was 1030 mm.
See notes on cookers regarding BS recommended dimensions for appliances.

Tumble driers

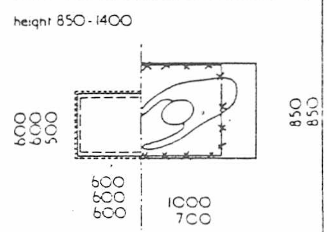


FIRA survey data: 13 items—front-loading

Per cent	Width (mm)	Depth (mm)	Height (mm)
100	610	570	
90	610	560	
75	600	560	670-850
50	590	550	
10	490	490	

See notes on cookers regarding BS recommended dimensions for appliances, previous page.

Refrigerators

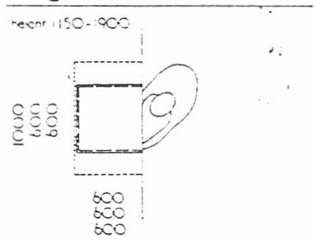


FIRA survey data: 24 items—freestanding*

Per cent	Width (mm)	Depth (mm)	Height (mm)
100	610	710	
90	600	610	
75	560	600	840-1370
50	560	600	
10	460	510	

* Built-in refrigerators range in width from 590-610 mm and in depth from 570-710 mm.
See notes on cookers regarding BS recommended dimensions for appliances, previous page.

Fridge freezers

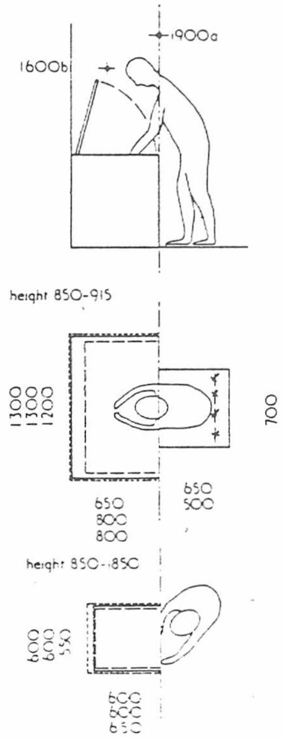


LA survey data: 29 items—freestanding

Per cent	Width (mm)	Depth (mm)	Height (mm)
100	1010*	670	
90	610	610	
75	610	600	1140-2050†
50	600	600	
10	500	600	

* This accommodates double door items with fridge and freezer side by side.
† Only one item was 2050 mm high. Others were less than 1890 mm.
See notes on cookers regarding BS recommended dimensions for appliances, previous page.

Freezers



Most FIRA data is for upright freezers, shown in the bottom diagram.
a At front edge of freezer.
b 450 mm from front of freezer.

FIRA survey data: 18 items—upright freestanding*

Per cent	Width (mm)	Depth (mm)	Height (mm)
100	760	720	
90	610	650	
75	600	650	850-1840
50	550	600	
10	500	580	

* In a FIRA sample of four built-in freezers all widths were 600 mm and depths ranged from 570-580 mm.
In a FIRA sample of 15 chest freezers the largest width was 1600 mm and the greatest depth was 735 mm. Heights ranged from 850-915 mm.
See notes on cookers regarding BS recommended dimensions for appliances, previous page.

Fig.2.6 Cont.

2.3.0 IMPLICATIONS FOR THE EXPERIMENTAL DATA BASE

This section represents the interface between the input information presented in the previous section (2.2.0) and the room layout appraisal model presented in chapter four. Here, the input data is translated into a form suitable for the computer software.

2.3.1 The room geometry

In section 2.2.2 it was noted that there were three broad classifications of kitchen. It was also noted that the room layout appraisal model was limited to discrete room analysis. As a result only working kitchens or kitchens with distinct seperable dining areas were used within the experiments, described in chapter six.

This sub-section demonstrates how kitchen geometries (donated to the author by professional kitchen designers - see acknowledgements and appendix G, section G.2.1) were converted to a format compatible with the capabilities of the computer model. This is done with the aid of the figure below.

A

B

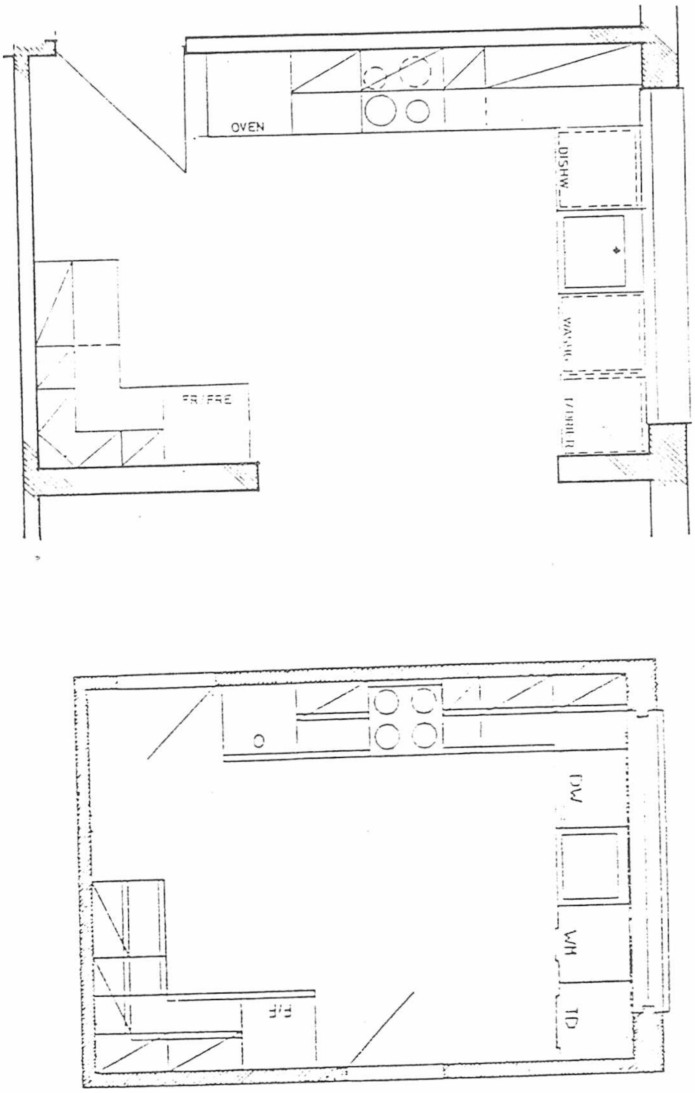


Fig.2.7 Room Geometry Conversion

Diagram A is a representation of a kitchen geometry given to the author by a professional kitchen designer. Diagram B is a representation of the same room as input to the modelling system. Note that both drawings are not to the same scale.

Together they demonstrate how the dining area was seperated from the working kitchen element of the combined kitchen. In this case, a partition and a 900mm door were inserted in the circulation space.

Other changes to the original fitment layout were forced by the capabilities of the modelling system as currently implemented, i.e. the restrictions imposed on menu size.

Examples of this in figure 2.7 are:-

1. Conversion of 1200mm long high level unit into two 600mm long high level units.
2. Omission of high level cooker hood.
3. Conversion of 500mm base and high level units into 600mm long units.
4. Conversion of 900mm corner base unit to 1200mm and 300mm base units in a corner assembly.
5. Conversion of 600mm corner high level unit to 600mm corner high level unit and 600mm high level unit.

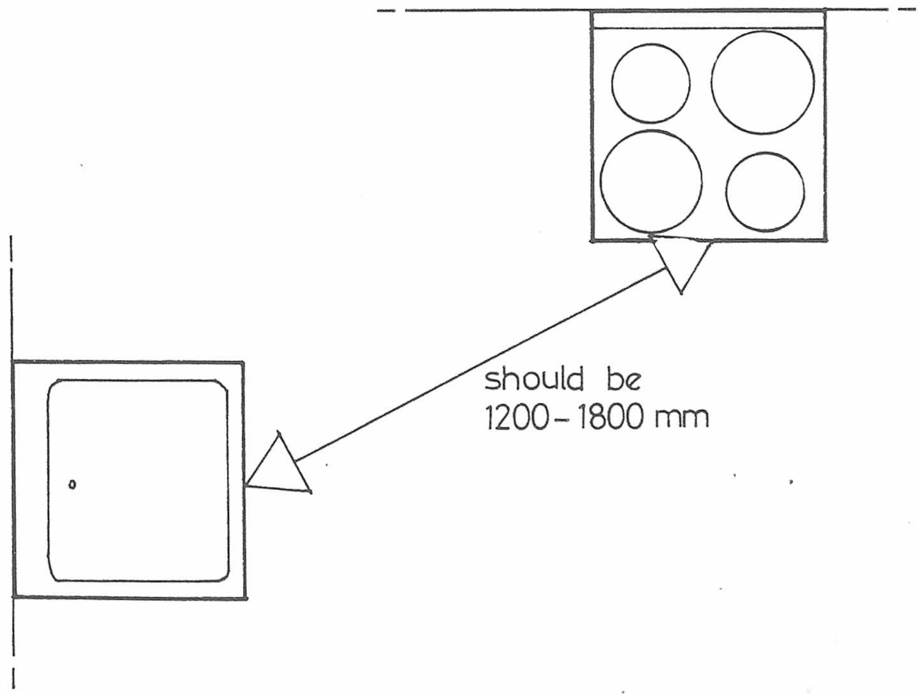
However, it is clear that both A and B represent the same room with an identical (virtually) layout. Overall room dimensions are constant betweem A and B as is the relative position of the appliances and the sink.

2.3.2 Implications of the meal preparation process.

This sub-section examines the consequences of some of the rules governing the location of furniture elements described in section 2.2.2 with regard to the association penalty element of the room analysis model.

The association penalty is discussed more fully in chapter four but for now, it is enough to know that it seeks to assess the relationship between one furniture element and another by comparing the distance between them to a predefined max/min desired distance. This concept is understood more easily with the aid of a diagram, see figure 2.8 below.

Fig.2.8 Cooker/Sink Association



The rule for the association of sink and cooker was that they should be between 1200mm and 1·800mm apart. This rule is mimicked in the association test, see figure 2.9 below.

OBJECT NAME	OBJ No.	MIN. DIST	MIN PEN	MAX. DIST	MAX PEN	OBJ No.	MIN. DIST	MIN PEN	MAX. DIST	MAX PEN	OBJ No.	MIN. DIST	MIN PEN	MAX. DIST	MAX PEN	OBJ No.	MIN. DIST	MIN PEN	MAX. DIST	MAX PEN	OBJ No.	MIN. DIST	MIN PEN	MAX. DIST	MAX PEN
1 COOKR	21	1800	7	4500	1	22	1175	7	4500	1	23	1175	7	4500	1	0	0	0	0	0	0	0	0	0	0
2 SINK6	1	1220	3	2230	5	4	1200	4	2400	5	21	10	1	2160	4	0	0	0	0	0	0	0	0	0	0
3	1	1220	3	2230	5	4	1200	4	2400	5	21	10	1	2160	4	0	0	0	0	0	0	0	0	0	0
4 FRIDG	1	1220	4	2830	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5 WASHC	2	10	1	1520	4	3	10	1	1520	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6 TUMBD	5	10	1	1220	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7 DISHW	2	10	1	1520	4	3	10	1	1520	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8 WT600	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9 WTRC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10 WTLC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11 WT500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12 WT300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14 HLC60	1	1100	3	4500	1	2	1365	3	4500	1	3	1350	3	4500	1	0	0	0	0	0	0	0	0	0	0
15 HL500	1	1140	3	4500	1	2	1350	3	4500	1	3	1330	3	4500	1	0	0	0	0	0	0	0	0	0	0
16 HL600	1	1100	3	4500	1	2	1365	3	4500	1	3	1350	3	4500	1	0	0	0	0	0	0	0	0	0	0
17 HL300	1	1120	3	4500	1	2	1305	3	4500	1	3	1290	3	4500	1	0	0	0	0	0	0	0	0	0	0
18 FRFRE	1	1280	4	2860	5	2	1320	4	2880	5	3	1320	4	2880	5	0	0	0	0	0	0	0	0	0	0
19 TD600	1	960	3	4500	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig.2.9 Table of Association Penalties

As can be seen from the figure, element 2 (sink6) is related to element 1 (cookr) in that the minimum distance apart, with no penalty application, is 1220mm and the maximum distance apart is 2230mm.

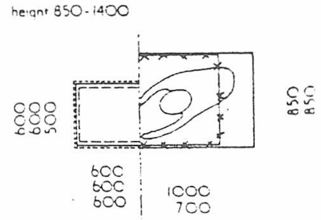
The discrepancy in the numbers between those of the literature search and those of the model is easily explained. The literature search rule refers to face/face dimensions whilst the model refers to centre/centre dimensions in three dimensions.

The model adopts this approach for computational simplicity. Since centre points are used as reference points the computer need not 'know' or calculate the orientation of each fitment. This saves considerably in computer memory and processing time.

2.3.3 Implications of meal preparation areas

Since user space requirements play a major role in the formulation of the room layout model, meal preparation areas, as they apply to each individual fitment, are of importance particularly in the experimental validation of the computer model described in chapter six.

(A) **Refrigerators**



FIRA survey data: 24 items—freestanding*

Per cent	Width (mm)	Depth (mm)	Height (mm)
100	610	710	
90	600	610	
75	560	600	840-1370
50	500	600	
10	460	510	

*Built-in refrigerators range in width from 590-610 mm and in depth from 570-710 mm.

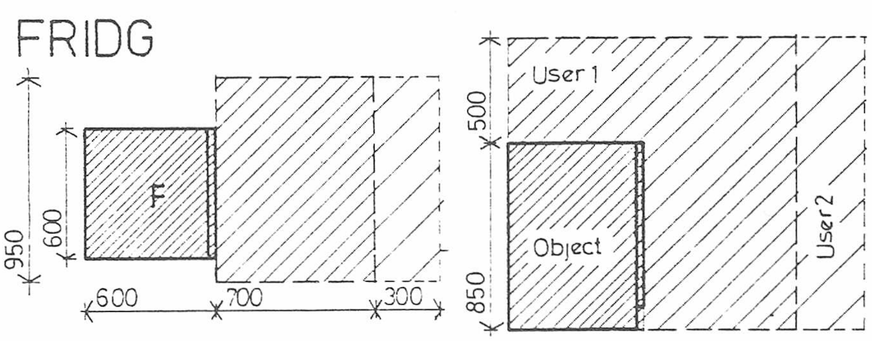


Fig.2.10 Fridge as a User Area

Figure 2.10 above shows the translation of user areas as defined in the literature search into user areas as defined for the purposes of the computer model.

User1 area is defined by the restricted access area of (A) combined with an additional 100mm wide area on the hinge side of the fridge, so as to allow the fridge door to fully open, as required by section 2.2.3. User2 area is equivalent to the unrestricted area plus the same 100mm slice.

The following pages present a 'menu' of possible kitchen fixtures and fitments as defined for use by the layout analysis model. This menu is based on the table of user requirements presented in figure 2.6.

TABLE

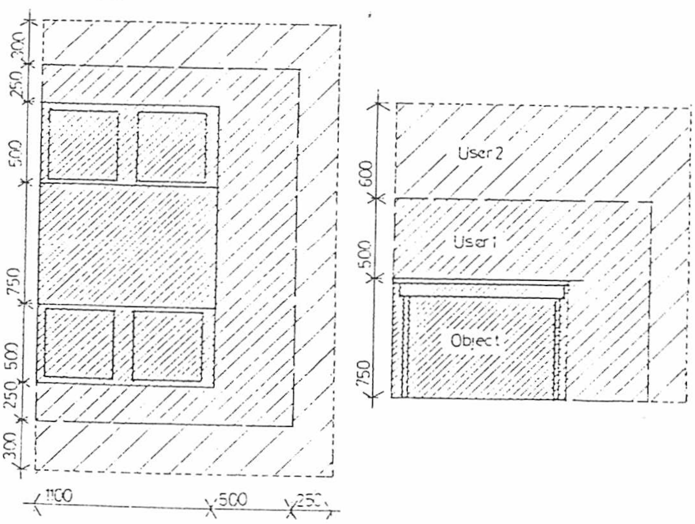
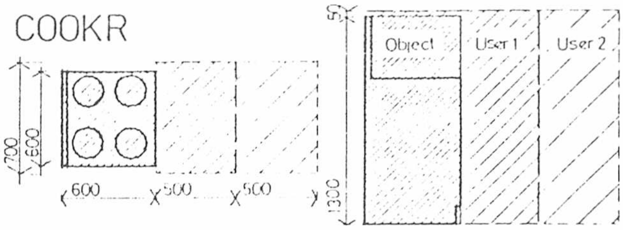


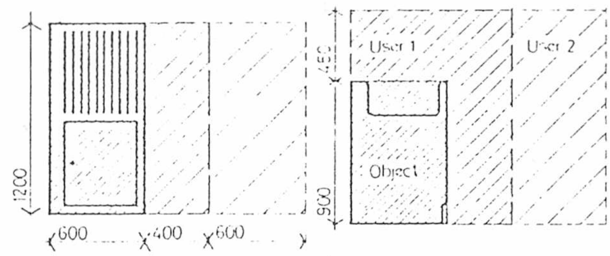
Fig.2.11 Menu of Kitchen Fitments

Fig.2.11 Cont.

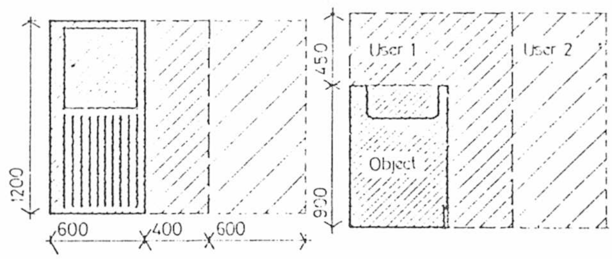
COOKR



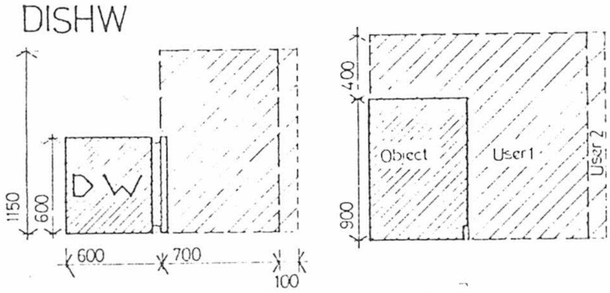
SINKL



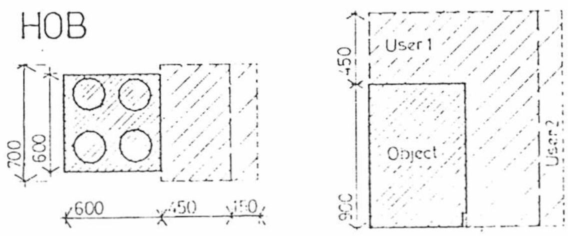
SINKR



DISHW



HOB



OVEN

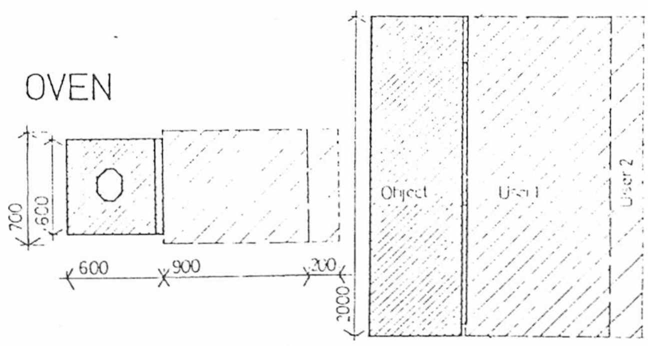


Fig.2.11 Cont.

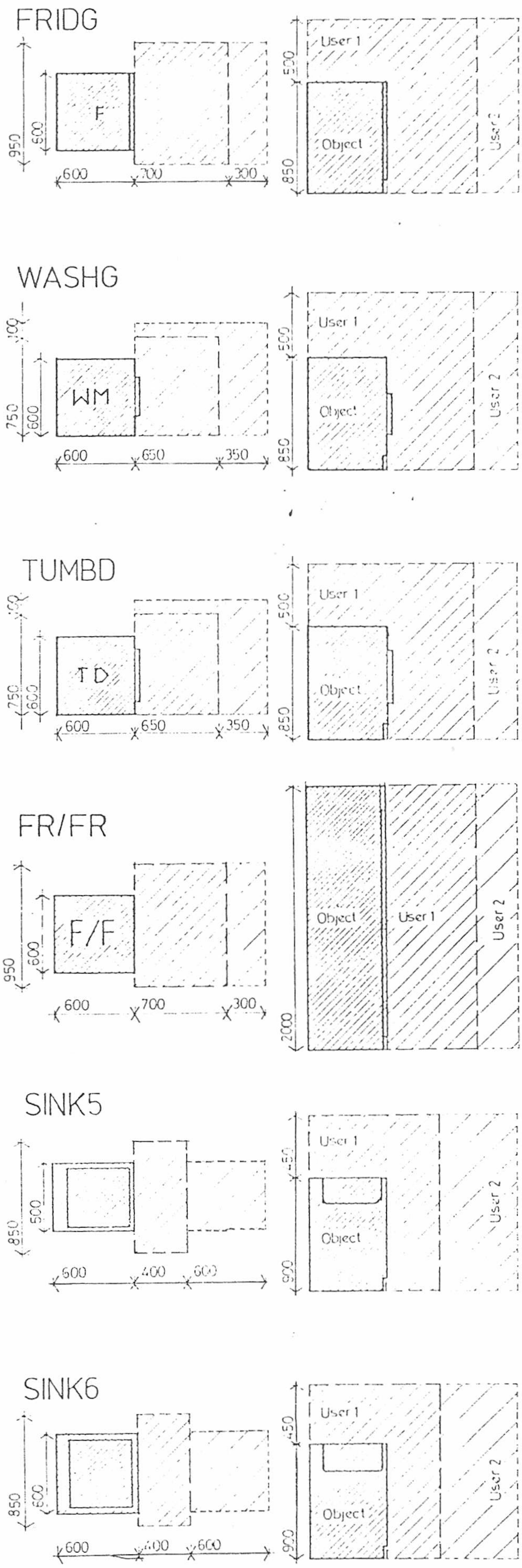
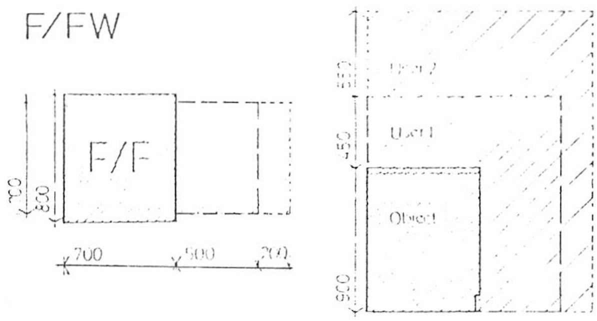
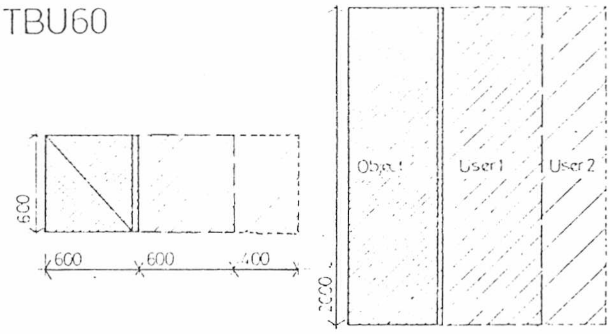

FRIDG
F
User 1
Object
User 2
950
600
600
700
300
500
850
WASHG
WM
User 1
Object
User 2
100
750
600
600
650
350
500
850
TUMBD
TD
User 1
Object
100
750
600
600
650
350
500
850
FR/FR
F/F
Object
User 1
User 2
950
600
600
700
300
2000
SINK5
User 1
Object
User 2
850
500
600
400
600
450
900
SINK6
User 1
Object
User 2
850
600
600
400
600
450
900

Fig.2.11 Cont.

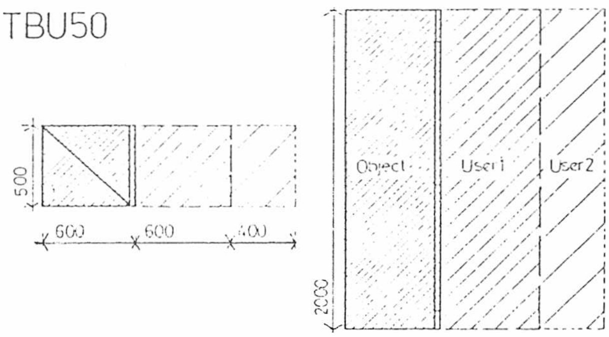
F/FW



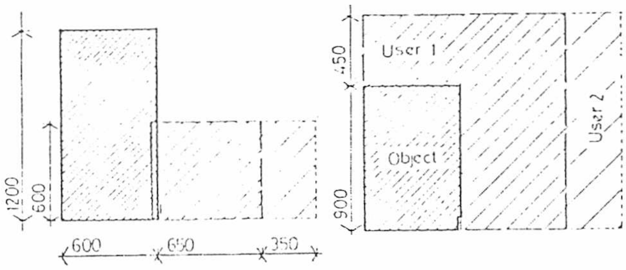
TBU60



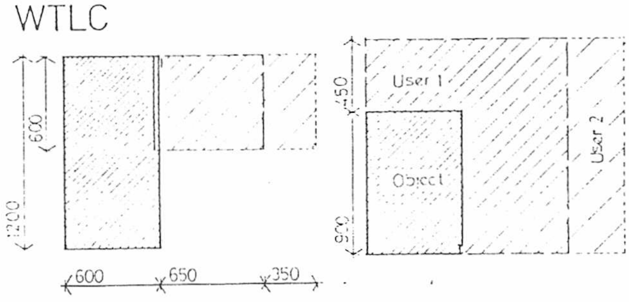
TBU50



WTRC



WTLC



WT5RC

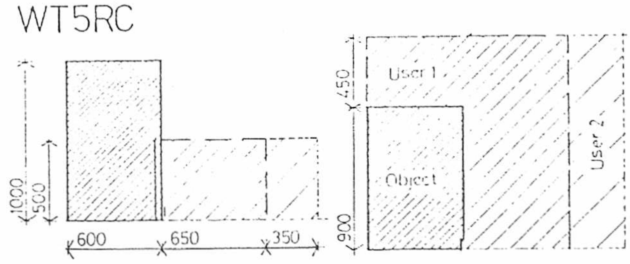


Fig.2.11 Cont.

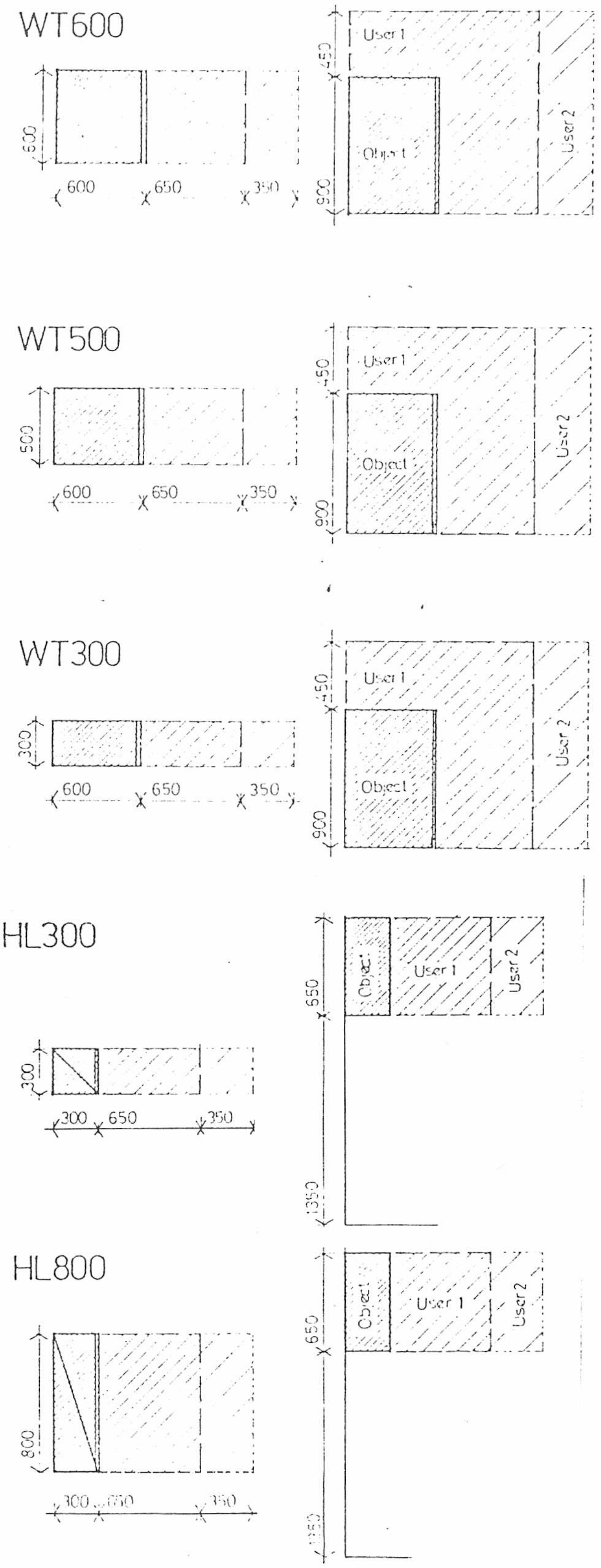
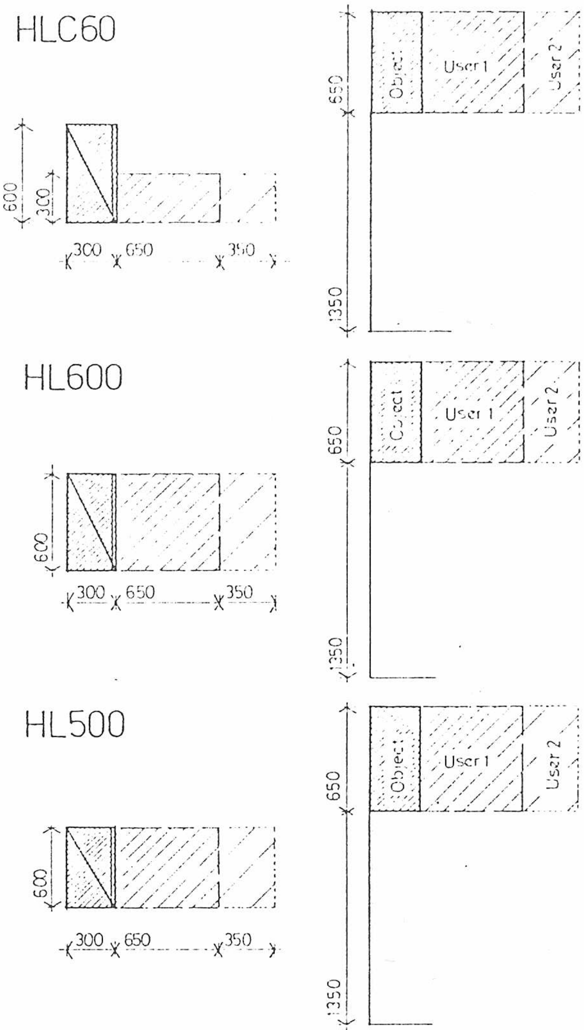

WT600
600
650
350
450
900
User 1
Object 1
User 2
WT500
500
600
650
350
450
900
User 1
Object
User 2
WT300
300
600
650
350
450
900
User 1
Object 1
User 2
HL300
300
300
650
350
650
1350
Object
User 1
User 2
HL800
800
300
650
350
650
1350
Object
User 1
User 2

Fig.2.11 Cont.


HLC60
600
300
300
650
350
Object
User 1
User 2
650
1350
HL600
600
300
650
350
Object
User 1
User 2
650
1350
HL500
600
300
650
350
Object
User 1
User 2
650
1350

2 4.0 SUMMARY

This chapter presents information pertaining to the test-bed situation of the domestic kitchen as it relates to a more general room layout analysis model.

Although the mechanisms of the model are not fully detailed until chapter four, this chapter has explained how ergonomic data gathered by previous researchers has been translated into a form suitable for use by that model.

2.5.0 REFERENCES

(2.1) Department of the Environment, Design Bulletin No.6 Space in the Home: Metric Edition, London, HMSO, 1968.

(2.2) Wilson, J.R.,Cooper, S.E., and Ward, J.S.A., Manual of Domestic Activity Space Requirements, compiled by Institute for Consumer Ergonomics for DOE, Loughborough University, unpublished.

(2.3) The British Standards Institution, BS 6222 Part 1 1982, Domestic Kitchen Equipment: Specification for Coordinating Dimensions, London, BSI, amended 1985.

(2.4) DOE, Design Bulletin No.24 Part 2, Spaces in the Home: Kitchens and Laundering Areas, London, HMSO, 1972, pp2 56

(2.5) DOE, op.cit., p29.

(2.6) Danish National Institute of Building Research, Report 60, User Study in the housing estate Vaerebo Park, Copenhagen, The Institute, 1969.

(2.7) Krantzer, K.R., 'Kitchens with a frontal siting of a diner space', Wirtschaftlich Bauen, October 1968, pp171,175.

(2.8) DOE, op.cit., p21.

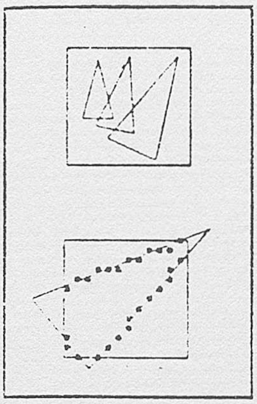
(2.9) Electricity Council, Guidelines to Kitchen Planning, Electricity Council, Free.

(2.10) Ed. Carolin, P., 'Update: Domestic Kitchen Design', Architects Journal, No.40, Vol.180, 3 Oct. 1984, p69,78.

(2.11) The British Standards Institution, op.cit

(2.12) The Marketing Consultancy Group, Furniture Industries Research Association, A Study of Furniture and Appliances Used in the Home, 1979, Prepared by MAS Survey Research Ltd.

(2.13) HDD Occasional Paper 2/81, A Survey of Tenants Attitudes to Recently Completed Estates, London, HMSO, 1981.



CHAPTER 3

INVESTIGATION OF EXISTING HARDWARE, SOFTWARE CONFIGURATIONS

CONTENTS

3.1.0 INTRODUCTION

This section notes the comparatively rapid development of computer graphics hardware and the relatively slow introduction of CAAD aids into architectural practice.

It then examines some existing draughting and modelling systems in terms, primarily, of their geometry input methodology; before drawing general conclusions about the effectiveness of existing man/machine interfacing techniques.

3.2.0 RAPID HARDWARE DEVELOPMENT

A brief review of the history of computer graphics shows that digital plotters were introduced by the Benson-Lehner corporation about 1953. By 1956 the Stromberg-Carlson corporation introduced the first graphical Computer Output Microfilm (COM) recorder. It wasn't until the 1960's that interactive graphics appeared on the screen when Sutherland and Johnston announced the work they were doing with a system called SKETCHPAD at the Massachusetts Institute of Technology (3.1).

By 1970 cheaper computer graphics, of the storage tube variety, were on offer by the Tektronix corporation. By the mid/late seventies new forms of graphic display such as vector scan and raster scan refresh displays were becoming available.

3.2.1 Introduction of CAAD aids into practice

Computer Aided Architectural Design has been around almost as long as there has been hardware to support it. During the early 1960's CAAD enjoyed a brief burst of popular (in terms of the architectural press at least) acclaim; however, few practices became actively involved with CAAD at that time.

It wasn't until around 1978, when the rigours of the 1974 economic recession had to some extent abated, that CAAD sustained a revival in interest (3.2). This is evident from the continuing prominence of the topic within the architectural and technical press.

However as the table below shows (fig. 3 1), the number of companies in the construction industry actually utilising CAAD systems is quite low in comparison to the total number of said companies.

System Supplier	System Name	Number of Installations UKci	UK	EUR	WOR	TOT	Launch Date
Applicon	Image	7	51	200	1200	1400	
Autotrol	GS-1000		10	20	350	400	
ARC Ltd	GDS	20	23	3	23	49	1980
Admel/Bruning	Easidraf2	3	3		125	150	81/82
Cadam Inc	Cadam	5	20	60	200	300	71/78
Calcomp	IGS 500	3	4	25	90	120	1969
Carbs Ltd	Carbs	11	13	1	2	16	1973
Calma	Cadec		60	20	900	1000	71/72
CIS	Medusa	4	28	53	29	130	1980
Computervision	CAE	7	100	600	2400	3000	1973
Genesys*	Gable	6	8		2	10	1981
Intergraph	IGDS	10	40	90	350	500	73/81
GMW Computers	Rucaps	46	46	9	12	67	1977
McAuto	Unigraphics	1	17	18	170	200	1978
Oasys Ltd	Cadraw	4	6			6	80/81
Olivetti	IGS	3	15	60	30	100	1980
PAFEC Ltd	Dogs	3	27		7	34	1980
Summagraphics	Datagrid	6	10	15	120	150	79/80
Scott Wilson K	Gipsys	1	1			1	1979
Approximate Totals		140	500	1200	6000	7700	

Key: UKci construction industry installations in the UK
UK total installations in the UK
EUR remaining installations in Europe excluding UK
WOR remaining installations worldwide excluding Europe

Fig.3.1 CICA survey showing the adoption of CAAD systems (after Hamilton,I.)

This lack of involvement is even more evident when the architectural element is to some extent isolated. (See fig. 3.2). This diagram is based on data provided by the AJ Computing Club membership list, relating to architectural practices or departments, public and private. The map serves as an indication of the geographical location of each practice user. Although the clubs' membership cannot be considered comprehensive; it does give a large representative sample of the architectural practices utilising CAAD aids.

COMPUTERS

ACT Sirius	5
Apple	55
BBC Micro.	19
DEC	12
Hewlett Packard	5
Olivetti	5
Pet	40
Sinclair ZX81	11
Spectrum	7
Superbrain	4
Video Genie	3

DRAUGHTING SYSTEMS

Applicon	1
ARC II	2
Calcomp	2
Gable	2
GDS draughting	5
Intergraph	4
Rucaps	12
Scribe	4

Note other software widely utilised is:

NBS (Specifications) 10
Visicalc (spreadsheet) 24
Wordstar (Wordprocessing) 16

Note these figures are based on information supplied by 209 AJ Computing club members

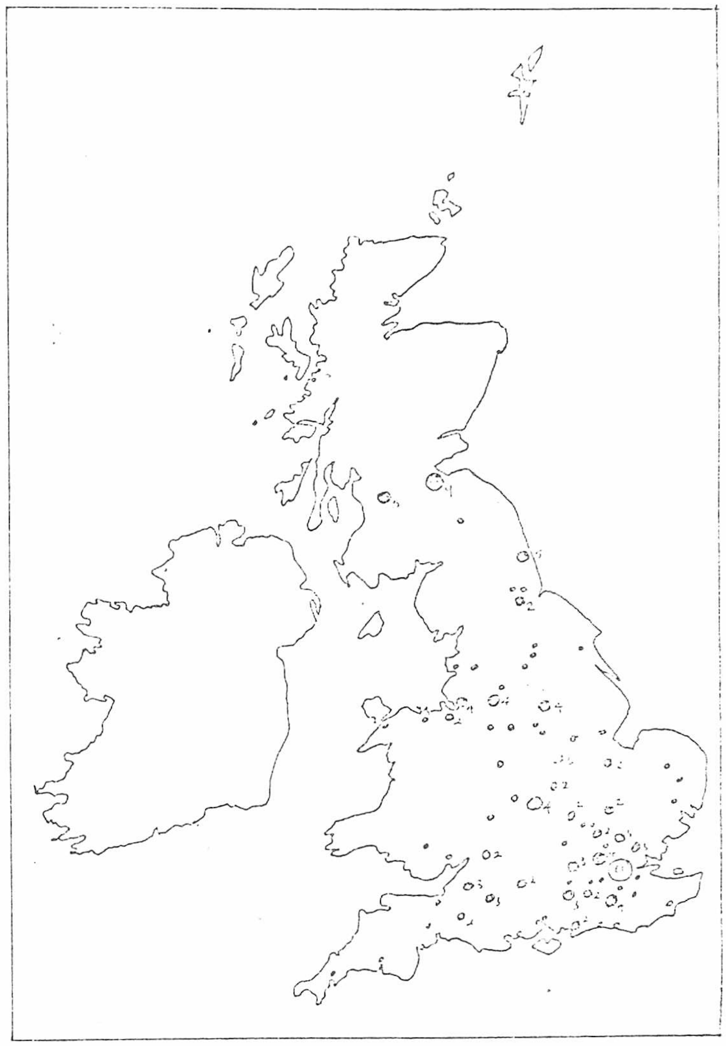


Fig.3.2 AJ computer club membership

One of the most important points to note is that the majority of the club members have opted to use small micro's such as the APPLE or PET, some even use home computers like the SINCLAIR, rather than larger machines such as the DEC PDP.

This is not surprising given that larger computers (and larger computer systems) require a greater capital investment than the smaller micro and that the pattern of current architectural practice does not lend itself to large scale investment in new technology (3 3).

According to Bijl et al., (3.4) those practices which have implemented large CAAD systems, exhibit four common characteristics. Firstly they are usually involved in a single building type (eg.hospitals, schools, mass housing, airports etc.,); secondly the construction methods they use are restricted; thirdly they anticipate long term building projects and finally they operate relatively stable protocols for the the organisation of their design practices. Significantly these characteristics are more likely to be maintained by a large and financially stable practice.

The majority of registered architects (some 85%) work in practices employing 10 or less architectural staff - ie. the smaller practice (3 5).

To the smaller practice, as was noted earlier, the high level of capital investment required to implement a large CAAD system is a major disincentive. However other disincentives are also evident such as:-

1. Lack of expertise in utilising CAAD aids (and the expense of 'buying in' such expertise).

2 Loss of control over the design process by team leaders and principals.

3 Disruption of idiosyncratic methods of working.

4. Lack of flexibility exhibited by the CAAD aids themselves.

This last factor is very important. If the lack of flexibility in existing systems can be overcome, both in terms of the building that can be described and the manner of its description, then comprehensive CAAD systems would become a more practicable proposition to more of the smaller practices, since the financial risks would be reduced due to the increased variety of jobs that could economically be computerised.

3 3.0 EXISTING SYSTEMS

The investigation of existing systems takes the following course:-

1. Definition of type of system to be investigated.

2 Enumeration of hardware devices generally available for geometry input.

3. System by system analysis, in terms of geometry input methodology.

3.3.1 Definition of systems

All the CAAD aids that have been investigated have at least the following two properties in common.

1. The designer must describe the building geometry to the computer.

2. All the systems,with the exception of ROBOGRAPHICS, were used by the author over a period of several days. Only a four hour demonstration of ROBOGRAPHICS could be arranged.

This, of course, means that a fairly representative sample of both draughting and modelling systems have been considered. Implementations of GABLE and GRAMP are available within SSSA and were examined in August 1983. RUCAPS 8 was used by myself at the office of Reiach and Hall, Edinburgh, in November 1983, whilst ROBOGRAPHICS was demonstrated in June 1983 at the Schoolhill premises of RGIT.

The comments made in later sections about these systems pertain to the implementations available to me at that time and place. However it should be noted that as a result of commercial pressures many of these systems are under a continuous process of updating and improvement. As a result some of the criticisms made may no longer be valid for newer versions of the system.

3 3.2 Graphical input devices

There are three principal means of effecting geometry input and consideration of the variety of means available will concentrate on these areas:-

1 Keyboard.

2. Screen.

3. Tablet.

Note that in terms of CAAD only two dimensional input devices need be considered.

3.3.3 Input via the keyboard

Most if not all graphic terminals are provided with a QUERTY keyboard. Obviously this can be used for direct keying of absolute cartesian coordiantes describing the building's geometry.

Sophistication in the modelling software would allow relative or polar coordinate systems to be used.

This method of geometry input is the most machine-friendly manner of creating geometry descriptions.

3.3.4 Input via the screen

Input devices using the screen can be broadly classified into two groups, positioning devices and pointing devices. All positioning devices make use of a cross-hair cursor so as to feedback to the user the current position of his input.

(A) Stepping keys

To a certain extent this might be viewed as an example of sophisticated keyboard entry. The software assigns certain keys move/draw commands in a relative coordinate environment. An image is created by repeated operation of sequences of keys; the cursor indicating the current start point of the next line (3.6). See figure 3.3 below.

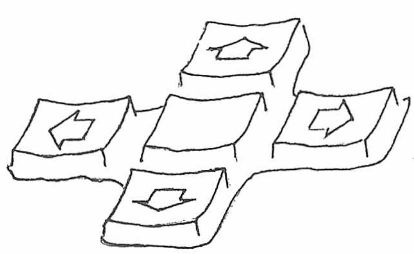
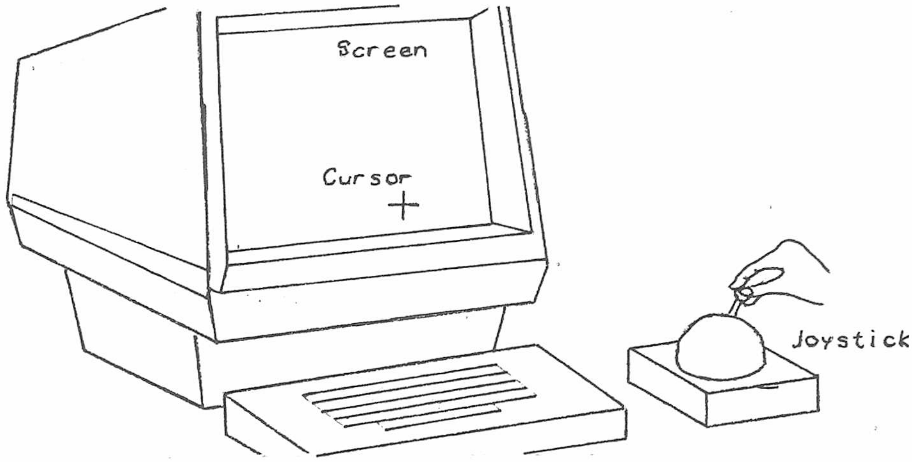


Fig.3.3 Stepping keys (after Newman & Sproull)

(B) Joystick

In as much as the joystick controls a cursor, the input operations are similar to that of using stepping keys. However there are differences, the user is not restricted to step movements, nor is he restricted to orthogonal movements - that is the X and Y coordinate of the cursor can be changed simultaneously (3.7). See figure 3.4 below.

Fig.3.4 The joystick (after Reynolds)



(C) Trackerball

With this input device cursor movement on the screen is very similar to that of the joystick, although the ergonomics of the device are quite different (3.8). See figure 3.5 below.

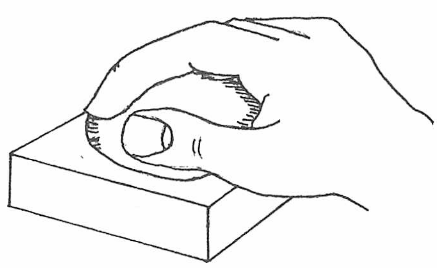


Fig.3.5 Trackerball (after Newman & Sproull)

(D) Thumbwheels

Some graphics display terminals, such as the Tektronix 4054 provide this input device on the keyboard. In philosophy, it is mid-way between the joystick and stepping keys in that although there is no stepped movement in the cursor, X and Y translations in the cursor position are independant in each other.

(E) The mouse

As the diagram shows, the mouse consists of two thumbwheels mounted at right angles to each other on the bottom of a box. This makes the action of the cross-hair similar to that of the joystick driven cross-hair. However the mouse is different from the joystick in that when the cross-hair gets to one edge of the screen it will 'wrap round' and appear at the opposite edge (3.9). See diagram 3.6.

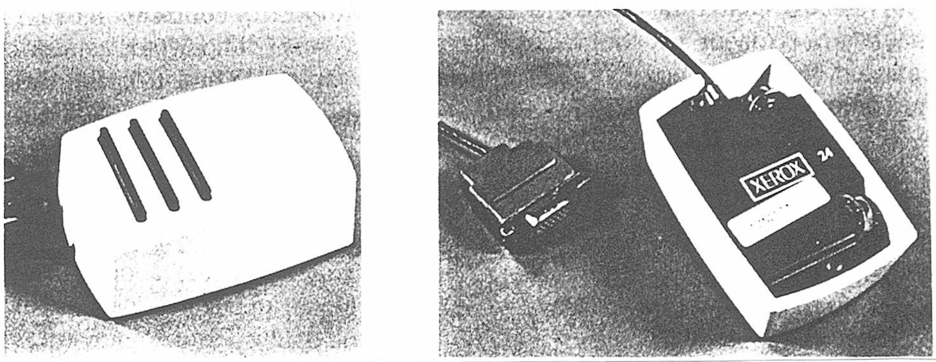


Fig 3.6 The mouse (after Newman & Sproull)

(F) Lightpen

So far we have discussed positioning devices. The lightpen is a device for pointing at the screen. If it is pointed at the screen then it generates information from which the item being pointed at can be identified by the program (3.10). See figure 3 7 for an illustration. However, as Reynolds has said, the lightpen is not very precise, and is actually poor ergonomically, as the user usually has to hold the pen perpendicular to the screen as he draws.

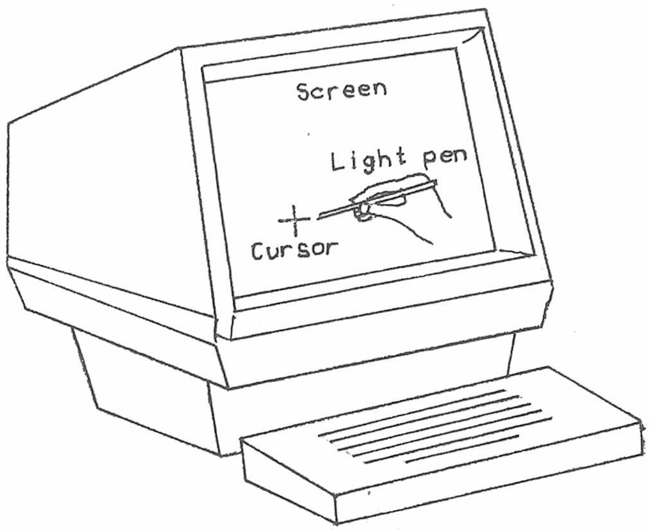


Fig.3.7 The lightpen (after Reynolds)

3 3.5 Input via the tablet

Also known as a digitiser or pencil follower. In itself, it is not an input device. The input is derived from the position of a stylus or puck placed on top of it.

A tablet can conventionally work in two modes; point digitising and continuous digitising.

In point digitising the pen is pointed at the tablet, the coordinates derived from this operation can be converted into the appropriate screen location by the application software.

In continuous digitising the pen is moved over the surface of the tablet, and the computer, at fixed intervals of time, interrogates the stylus to establish its current position. This allows complex shapes such as the contours of a map to be traced accurately (3.11). See figure 3 8 for an illustration.

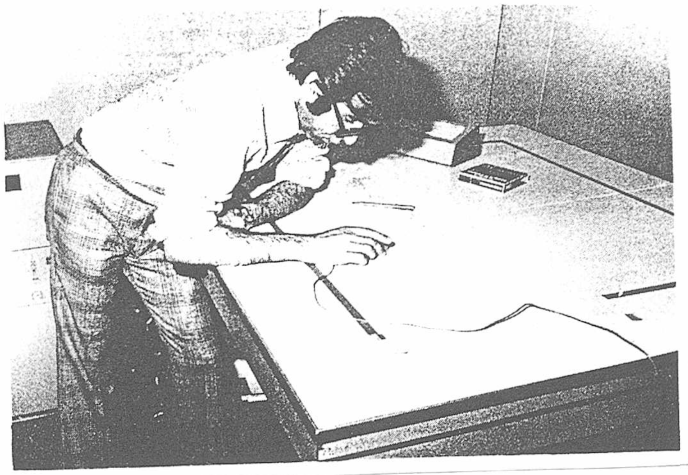


Fig.3.8 The digitising tablet (after Reynolds)

3 4.0 SYSTEM BY SYSTEM ANALYSIS

The system by system analysis will follow the following pattern in each case.

1 Introduction of the system and its conceptual background.

2 Identification of hardware configurations used in terms of input devices.

3. Discussion of the quality of the user interface.

4. System description and personal comment on the extent to which input methodology imposes constraints on the designers use of the system with particular regard to geometry input.

Item (3) will be discussed in terms of Newmans concept of the four components of the user interface.

1. "'A users model' ie. the users conceptual perception of the information he manipulates and of the processes he applies to this information.

2. A COMMAND LANGUAGE in which the user expresses his commands to the program.

3. FEEDBACK provided by the computer to assist the user in operating the program.

4. Information display showing the user the state of the information he is manipulating." (3 12)

3 4.1 Gable 9

(A) Introduction

GABLE: General Aid to Building Layout and Evaluation. This is a modelling and draughting package developed at Sheffield University by architects and for architects. It became commercially available in 1981 (See figure 3.1).

GABLE is marketed in two linked sections. The modelling package and the draughting package. The draughting package (at least in this early implementation) is heavily dependant on the modelling package for its efficiency and so has been excluded from further consideration.

The modelling package is in three major parts; input, where the geometry description is assembled in two dimensional form; interpretation, where the geometry description is assembled into a building model (3.13), (3.14) and output, where the assembled building model can be appraised visually and numerically.

Conceptually GABLE is an attempt to bring modelling CAAD aids to the architectural practice at a price closer to that which they can afford (approx. £50,000 including hardware and staff training) so that the quality of design can be improved as well as the speed of drawing production. All in a manner suited to the architects method of working ie. designing at the drawing board (terminal).

(B) Hardware configurations

Tektronix 4052 or 4054 Graphic Computer System containing 56k user workspace memory and an 11" or 19 " high resolution storage tube display.

Tektronix 4952 Joystick in 4052 configurations (the 4054 has inbuilt thumbwheels).

Tektronix 4907 file manager and dual floppy disk drive.

Tektronix 4663 A2 2-pen flatbed plotter/digitiser.

It should be noted that the Tektronix 4050 series terminals are powerful desk top computers combined with a high resolution storage tube screen suitable for architectural work. They are also simple to operate, cheap (relatively) and compact.

(C) The user interface

(i) The users model

The basis of the users model is the system manual describing what he can do, how he does it, and what happens when he does do it.

Unfortunately the GABLE manual is comparatively poor. It offers little information on how data is captured and stored.

It does outline each of the system capabilities and their method of operation, especially those aspects relating to plan input - where the bulk of geometry input lies.

However the sections of the manual dealing with the more automated functions of the software such as creating roof plans or elevations are inadequate. The user finds it difficult to conceptualise what he's doing.

(ii) Command language

GABLE is driven by the use of function keys and overlay cards. Each program segment has its own overlay. This is an effective way of communicating to the machine since it reduces the amount of keyboard typing required. Furthermore keys that do the same or similar functions in different parts of the program suite are located in the same location on the overlay in each part of the suite.

(iii) Feedback

Bearing in mind that a storage tube is the only medium available for display the quality of the feedback is quite good. Bells prompt all inputs and errors are noted by a long bell sound. The maximum possible alphanumeric feedback is given.

(iv) Information display

As with all storage tube displays, the system suffers from the need to redraw the image that the user is working on. This is particularly marked when the user is deleting or modifying information already created, or alternatively changing scale or viewing area.

A particularly irritating feature is that if an image is redrawn, then the background grid is not redrawn at the same time.

(D) System description

The GABLE system attempts to allow the designer to design at the computer terminal. However it does not really succeed. This is best illustrated by a step-by-step description of how a typical floor plan might be generated.

Firstly the designer needs to know the shape of his enclosing perimeter wall. If he doesn't know this then he can do no more since GABLE needs this envelope to be enclosed before it can interpret the model, and trying to add the external wall at a later date can be a little difficult.

Once the designer has defined the external wall geometry he should define the wall construction. GABLE uses the construction specification to decide how to show and dimension the external walls.

The designer is now ready to start; he draws in the enclosing envelope, specifying the wall type, using thumbwheels and cursor, to point to the ends of lines. After he's formed a closed polygon the computer comes back with a display showing each wall element with a thickness.

Right, ready to start defining room layouts? Wrong. First it's advisable to make sure that the external walls have the correct dimensional aspect to each other. If this is not done now, then it becomes an extremely laborious and error prone task later on.

The designer finishes dimensioning and aligning the enclosing envelope; he's ready to start determining room layouts but first he'd best work out what the partitions are made of, to establish their thickness, otherwise he will have to be very careful when doing this for every line at a later stage.

This step by step approach gives some idea of the complexity of trying to design a building using GABLE. The same philosophy of placing followed by dimensioning is evident throughout the rest of the PLAN INPUT sections such as input door, window, floor and staircase (multiple floor segments).

However it should be said that if the user is unconcerned about exact placing and dimensioning of elements, the input process is much speeded up and approximate analysis of outline proposals would be possible.

In terms of the input methodology used, placing of elements is achieved by cursor positioning using thumbwheels or joystick, followed by a single keyboard entry to confirm the location. All dimensional data is input via the keyboard.

It is possible to use a digitising tablet or the plotter/digitiser instead of the joystick for positioning, however single keyboard entry is still needed. This makes the operation very user unfriendly in that the users attention is constantly wavering between the screen and the digitiser.

To conclude GABLE is user friendly in terms of inputting building geometry in that the designer draws on the screen in a completely unresticted manner. There is no locking on or off grids nor typing in of dimensions or coordinates. GABLE becomes unfriendly however when dimensional accuracy is needed.

3.4.2 Rucaps 8

(A) Introduction

RUCAPS: Really Universal Computer Aided Production System. (3.14). This is a two and a half dimensional draughting system. It was developed at the office of Gollins Melvin Ward Partnership, and is now marketed through a wholly owned subsiduary GMW Computers Ltd.

It was written by architects for architects. It became commercially available in 1977. (See figure 3.1).

Conceptually RUCAPS concentrates on the rapid production of drawings to justify itself. Few supplementary results in the form of thermal or accoustic analysis etc. are available, although simple scheduling in the form of no. of componants used is available. The modelling capability is used only to derive sections and elevations from the components making up the plan form.

(B) Hardware configurations

DEC PDP or Prime based dedicated mini-computer including 4 hard disk drives.

A0 size digitiser and puck.

Imlac Dynagraphic 19" high resolution vector refresh display including lightpen.

Benson A0 drum plotter.

RUCAPS is sold as a turnkey draughting system at a cost ranging from £60,000 for effectively two dimensional draughting to £140,000 for two workstations and full modelling capability.

(C) The user interface

(i) The users model

The RUCAPS 8 manual is quite effective in giving the user a good idea of what happens to the data he inputs to the system. This may be because the data structure used by RUCAPS is considerably more simple than that for GABLE for example.

In addition the manual gives a detailed step-by-step description of what happens during each program segment and informs the user of what he should do if anything goes wrong!

(ii) Command language

RUCAPS 8 is a menu driven software package. The user selects from a menu of available choices, either by typing in a key letter, or by using the light pen. (This is apparently the only effective use that is made of the light pen.)

When using the A0 digitising board and puck a menu is attached to the board. However no feedback is given on the screen as to which of the many available options on the tablet menu is currently active.

(iii) Feedback

Despite the fact that a refresh graphics display is available, there is comparatively little feedback, as such, during the geometry creation parts of the RUCAPS 8 software. Errors are only notified by the sounding of a bell. There are no prompts for input, other than the display of menus and the display of current cursor position.

(iv) Information display

At the end of each user action, the information display is updated. Only minimal use is made of the refresh graphics capability during user actions. For example, when the designer is manipulating an existing component on the screen then the component is temporarily attached to the cursor and can be 'dragged' accross the screen.

(D) System description

RUCAPS 8 is not designed for, nor is it capable of allowing the designer to design at the computer terminal. The user must have worked out the geometry of his project in some detail before attempting to use the system. This is best explained by describing how the building geometry is stored.

The designer starts by defining a project name and building(s) name(s) and also the grids to which all geometry constructs are related.

The user then creates or calls up from a standard library a component (for example a rectangle corresponding to a column) and locates this component in one or more places within a building at a defined floor level.

Thus the component is stored as one item of information; and the building level has a record of all instances of that component used at that level.

In terms of the input methodology, there are three seperate and distinct aspects; creating the component, amalgamating it into a supercomponent and placing it on a building level. Specific drawings are created by defining which part of a building and which classes of components are to be drawn.

With RUCAPS 8 there are two methods of creating a component; by keyboard entry or via tablet and puck.

Using the keyboard the designer types in relative move/draw commands, specifying line type and thickness and calling special functions such as 'circle' as and when necessary.

Using the tablet and puck, the computer automatically derives the same kind of data file that is obtained by 'keying in' as described above. However, here the designer 'picks up' functions from the tablet menu and then points on the tablet. The input is echoed on the screen. In this way sketch components can be defined rapidly by tracing over a manually created drawing. This process is not dimensionally accurate.

Facilities are provided such as user defined grids, and construction lines, to enable dimensionally accurate components to be created on the digitising tablet. However, the users attention will waver between the screen, showing what the component looks like so far and the tablet menu as he selects new functions. In addition choosing which combination of functions should be activated to achieve a particular result might confuse the novice user.

To conclude; RUCAPS is user friendly to the extent that the user model is well defined and the information display is good. It is unfriendly to the extent that a large part of the geometry input is either by keyboard entry (both dimensions and component names) or by the tablet with echo on the screen, resulting in a large amount of head movement between screen and tablet.

3.4.3 Gramp

(A) Introduction

GRAMP: Graphical Manipulation Program is at the core of much of the ABACUS software. It was developed originally for SPACES by Harvey Sussock.

The package concerns itself primarily with the manipulation of existing graphic data, whilst maintaining the integrity of any associated geometry model.

(B) Hardware configuration

This program is designed to run on a time-sharing mainframe facility and uses a Tektronix 4010 compatible graphics display device for output and cursor/keyboard input.

(C) The user interface

(i) The user model

The user has a fairly clear idea of the editing facilities of the program since each individual manipulation is clear, and comparatively simple. The user is unaware of the data structuring of the geometry, other than that he is resticted to orthogonal shapes.

(ii) Command language

GRAMP has a menu driven command structure employing the usual ABACUS feature of cursor picking of menu commands or typing the initial letter of the command.

(iii) Feedback

Within GRAMP feedback plays a comparatively minor role. Errors are noted only by a bell sounding - there is no echo of a users menu selection.

Whilst editing individual picture elements, the user has no direct visualisation of what he is doing until the editing process has been completed.

(iv) Information display

As was stated earlier (Chapter 3, Section 3.4.1) a storage tube suffers from the need to redraw the image that the user is working on subsequent to data modification and manipulation.

However this disadvantage is perhaps more marked in GRAMP than with GABLE since GRAMP manipulates whole primative objects rather than individual lines and several manipulations of a single object may be needed to effect the desired geometric outcome.

(D) System description

The GRAMP program allows the designer to manipulate geometry objects on the screen of the computer terminal. It is not concerned with the creation of the geometry object; this is done by utilising other related software.

GRAMP suffers from two principal limitations. Firstly it suffers from the need to periodically redraw the screen image, and secondly it can only manipulate orthogonal and rectilinear geometry primatives.

In terms of the facilities given to the designer the main notable omission is that the designer has no quick and accurate way of aligning parallel edges of two geometry primatives so as to make them colinear.

Having said that, GRAMP undoubtedly succeeds on its own terms as the provision of a simple geometry manipulation facility to be used in conjunction with other more complex building model appraisal packages.

3.4.4 Robographics

(A) Introduction

As an opening note, unlike the previous systems examined, the author has not had 'hands on' experience of this system. The following comment is based on technical brochures and attendance at a software demonstration.

ROBOGRAPHICS is a general two dimensional graphics system, which has recently appeared on the market. An enhancement of the basic system called SCRIBE, which was not demonstrated, allowed for three dimensional sectional overlays.

(B) Hardware configurations

Apple II or IIe Personal Computer with 64k RAM.

Two disk drives DOS 3.3.

Monitor (colour or monochrome).

Bitstick 3-axis precision controller.

Calcomp plotter up to A0 size.

(C) The user interface

(i) The user model

Visualisation of the ROBOGRAPHIC's system of operation is deceptively simple. It has been likened to word processing with graphic images.

However, although the strategic 'modus operandi' is clear; detailed operation is less clear, involving as it does, cursor selection of options from a palette of possible operations spread round the edges of the monitor.

I have had no access to the manuals and so have no indication of their quality. Nor can I determine whether detailed operations within the system are made clear by the manual.

(ii) Command language

ROBOGRAPHICS appears to be a modular system driven by menu selection. However the menus are not in the conventional format in a table but spread around the edges of the display area.

All input, apart from text entry, is done with the bitstick controller which is in essence a multi-keyed joystick.

(iii) Feedback

Much of the feedback given in ROBOGRAPHICS concerns how the bitstick controller is operated and is very useful and powerful.

There is some alphanumeric feedback as to line length, grid size and lock and current cursor position. One form of feedback which is missing is that there is no visual feedback giving confirmation that a particular program segment has been activated.

(iv) Information display

ROBOGRAPHICS uses a raster scan display and suffers from low resolution (512 x 512) compared to that of the storage tube or dynamic graphics display terminal.

In addition the menu options are quite difficult to see since they are sited at the periphery of the monitor.

Furthermore some of the editing facilities, in particular deletion of an element, suffers from the fact that the object or line must be undrawn.

(D) System description

For its price (about £6,000) ROBOGRAPHICS is a powerful geometry creation and manipulation package.

It operates in a manner similar to RUCAPS 8 in that libraries of objects are created and then placed on a drawing (which in itself is another library object.)

It is not really possible to sit at the terminal and start designing from scratch. Some idea of what the designer wants to draw must be in his mind.

However apart from the objections raised earlier in this discussion one of the main disincentives to using this system must be the difficulty of creating library objects of complex, non-orthogonal geometry, quickly and accurately. Although the feedback available would be of some help in this task, there appears to be no mechanism for either locking the start/stop point of a line onto another line nor is there any method for locking a grid onto an existing line, ie. the grid seems to be drawn at all times from the origin.

However to conclude, the system should be relatively easy to understand and operate and should be able to accomodate most architectural drawing work, when in experienced hands. Its most obvious feature to this end is that all geometry is created dynamically using cursor input. There is no requirement to type in coordinates or line lengths etc.

3.5.0 CONCLUSIONS

The following conclusions are based on the investigation of the systems which have been described in the previous sub-sections together with the analysis, to greater or lesser extent, of other systems such as BDS, Sue, Designer 1, and Source, to which the author was unable to arrange access.

1. The designer has to provide more information about the building model than he might otherwise do when utilising manual methods. A simple example of this is that both BIBLE and GOAL require absolute coordinate descriptions in three dimensions of the building model.

2. Creating a building geometry is potentially a more lengthy process than manual methods in that additional information is required.

3. A complete design concept is a pre-requisite to inputting a design model. Only GABLE, of the systems reviewed, attempts to allow the user to design at the computer terminal.

4. Many systems use a very machine orientated methodology for capturing the geometry model characteristics. For example, in RUCAPS 8 one of the main ways of creating a component (the basic building block of that system) is to type numbers into the computer.

5. Systems, such as GABLE, utilise cursor input for building geometry creation. This allows graphical display during the input process. However, these systems may impose constraints on the order and means by which a building geometry can be created.

6. As a result of the above factors, the design process is interrupted by the computer. That is, the designer cannot concentrate on the design task since much of his attention is diverted to driving the machine.

7 Most systems offer only a limited mechanism for the modification and repetition of data already entered into the machine. However, all such mechanisms are superior to manual means.

8. The considerations of cost, reliability and control are really outwith the scope of this research project, but it could be said that the majority of systems have a high investment cost and require specialist operators. In general, systems that have been on the market for several years and utilise 'standard' hardware are reasonably reliable, although the quality of the 'backup' in terms of user manuals etc. may be suspect in some cases.

3.6.0 REFERENCES

(3.1) Humcke, D.J., "Interactive Computer Graphics in CAD", Computer Aided Design, Jones, G.W.(Ed.), Vol. 8, No. 4, IPC Business Press Ltd., Guilford, Oct. 1976, p.233.

(3.2) Hamilton, I., Computer Draughting in the Building Team, CICA, Cambridge, 1981, p.6.

(3.3) Hammond B.G., Leifer, D., "A Graphics Interface to Complement Traditional Techniques", PARC 83 Proceedings, Online Conferences, Pinner, 1983, p.322.

(3.4) Bijl, A., Stone, D., Rosenthal, D., "Integrated CAAD Systems", Final Report to DOE on funded research project DRG 470/12, EdCAAD, University of Edinburgh, March 1979.

(3.4) Leifer, D. The Machine Refinement of Raw Graphic Data for Translation into a Low Level Data Base for a CAAD Decision Matrix, (Ph.D Thesis), Aberdeen, Scott Sutherland School of Architecture, Robert Gordons Institute of Technology, 1984.

(3.5) Newman, W.N., Sproull, R.F., Principles of Interactive Graphics, 2nd. Edition, McGraw Hill, New York, 1979, p.162.

(3.7) Reynolds, R.A., Computer Methods for Architects, Butterworth, London, 1980, p.30.

(3.8) Newman, W.N., Sproull, R.F., op.cit., pp.148,149.

(3.9) ibid.

(3.10) Reynolds, R.A., op.cit., pp.29,31.

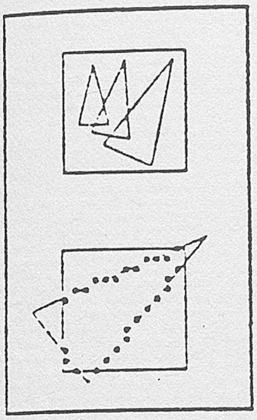
(3.11) ibid.

(3.12) Newman, W., "Some Notes on User Interface Design", Methodology of Interaction - IFIP Workshop, Guedj, R.A.(Ed.), North Holland Publishing Co., Amsterdam, pp.325,326.

(3.13) Lawson, B.R., Riley, J.P., "ISAAC A Technique for the Automatic Interpretation of Spaces from drawn floor plans", CAD 82 Conference Proceedings, Pipes, A.(Ed.), Butterworths, 1982, pp.663,666.

(3.14) Lawson, B.R., Riley, J.P., "RODIN A System of Modelling Three Dimensional Roof Forms", op.cit., pp.667,670.

(3.15) Reynolds, R.A., op.cit., p.78.



CHAPTER 4

A NEW LAYOUT APPRAISAL SYSTEM

CONTENTS

4.1.0 INTRODUCTION

"In its most general terms, layout planning can be described as the attainment or satisfaction of multiple objectives subject to a variety of constraints." (4.1)

The objectives typically include:-

1. Effective movement of materials and personnel.
2. Effective utilisation of space.
3. Adaptability to unforseen changes.
4. Safety.
5. Good appearance.

Common constraints might include:-

1. One or more fixed activities.
2. Activities which must be seperated.
3. Regulation restrictions.
4. Room size.
5. Budget.
6. Time.

Typically the designer manipulates these objectives and constraints in an intuitive manner to yield one or more acceptable design concepts which are then firmed up into detailed layouts.

4.2 0 EXISTING ACTIVITY SPACE MODELS

Over the years, some designers have found this intuitive approach too subjective. Several more rigourous approaches have been applied, either to assist in the construction of the design or to appraise a completed design.

1. Bubble diagramming.
2 Graphical layout techniques.
3. Scoring techniques.
4. Clustering techniques.
5. Layout algorithms. (4.2)

In the following sections, existing activity space models and proceedures will be examined in the light of the above classifications.

4.2.1 Bubble diagramming

This is not really a technique as such but rather a method of illustrating an intended arrangement of activities. It is popular among architects and interior designers. (4.3)

4.2.2 Graphical layout techniques

This approach could be regarded as a structured manual approach to layout design. Some examples of the variety of approaches are indicated in the following pages.

<u>The Svennar System</u>

The Svennar System is mostly the result of data given in the NBI(Norgesbyggforskningsinstitutt - Norwegian Building Research Institute) data sheets on spatial standards in the home.

Each activity element is represented by a user area - similar to those identified in section 2.3.3. Rooms are on a modular grid, and activity elements are manually placed in the appropriate rooms.

The system relies on conventional draughting techniques. Each activity element user area is hatched allowing a visual appreciation of congestion within a room layout to be made from the degree of cross-hatching. (4.4)

<u>The Alice Thiberg System</u>

This system (really more like a design guide in application), is very similar to the Svennar System above with hatched user areas. Though it appears that some method of calculating the furniture content required by each room space has been devised (4.5)

The Bjorko System

The NBI were increasingly confronted with the problem of assessing the usable value of a house plan, and the Bjorko System was devised as such an appraisal method.

The Bjorko System identifies five sub-analysis:-

1. Fixed installations.
2. "Furnishability".
3. User areas.
4. Circulation (between rooms).
5. Room relationships.

These analysis are mainly of a graphical nature. (4.6)

The Richard Muther System

This system is driven by two sets of input information. A schedule of accomodation areas and a matrix of the relationships between units of accomodation. See figure 4.1

The relationships are ranked from essential to unimportant (an additional relationship is X-Undesirable). These relationships are converted into diagramatic form using a number of lines code

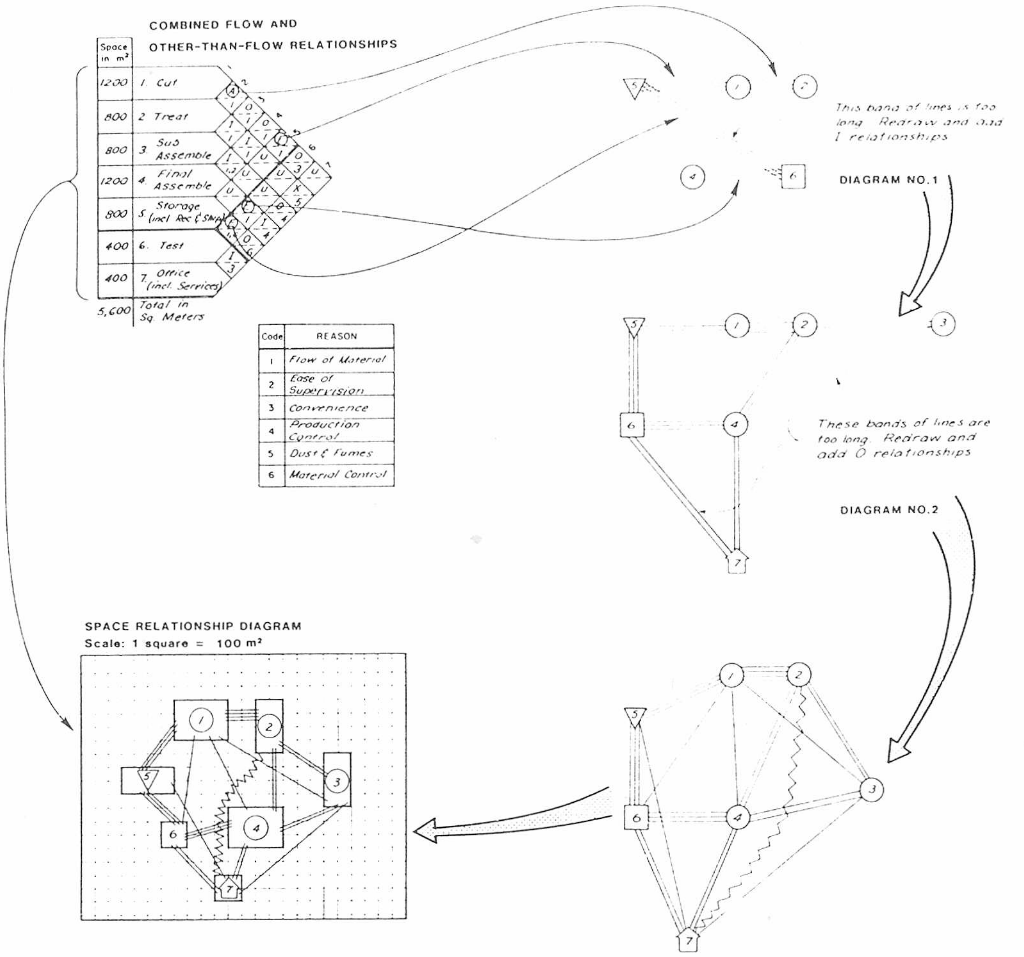


Fig.4.1 A Graphical Layout Technique (after Muther et al.)

and a length of line scale. Essential relationships are given the shortest line, with other relationships being scaled from that class. Essential relationships are drawn first.

The diagram is redrawn after each relationship class has been positioned, if the line of scale distance has been exceeded.

The finished diagram is 'exploded' and the symbols redrawn as the corresponding areas of each activity element. The final step being to fit the activity areas of each activity element into the room space available, adjusting the shapes of each element as required whilst retaining the integrity of the relationships identified earlier.

4.2.3 Scoring techniques

Closeness scoring techniques do not generate layouts; they evaluate how "good" a layout is.

Most of the scoring techniques assume the designer is seeking to maximise the closeness of highly interrelated activities. Well designed scoring techniques will highlight adjustments to a layout and guide the planner to a better design.

A simple method of utilising this technique would be to plot intensity vs. distance for each of the desired relationships. (Intensity being a measure of the desirability of a particular relationship - traditionally this is a measure of transport costs between nodes.)

The total area under and to the left of each point represents the measure of "goodness" of the design. Individual points with a large area 'behind' them indicate weaknesses in the design. See figure 4.2

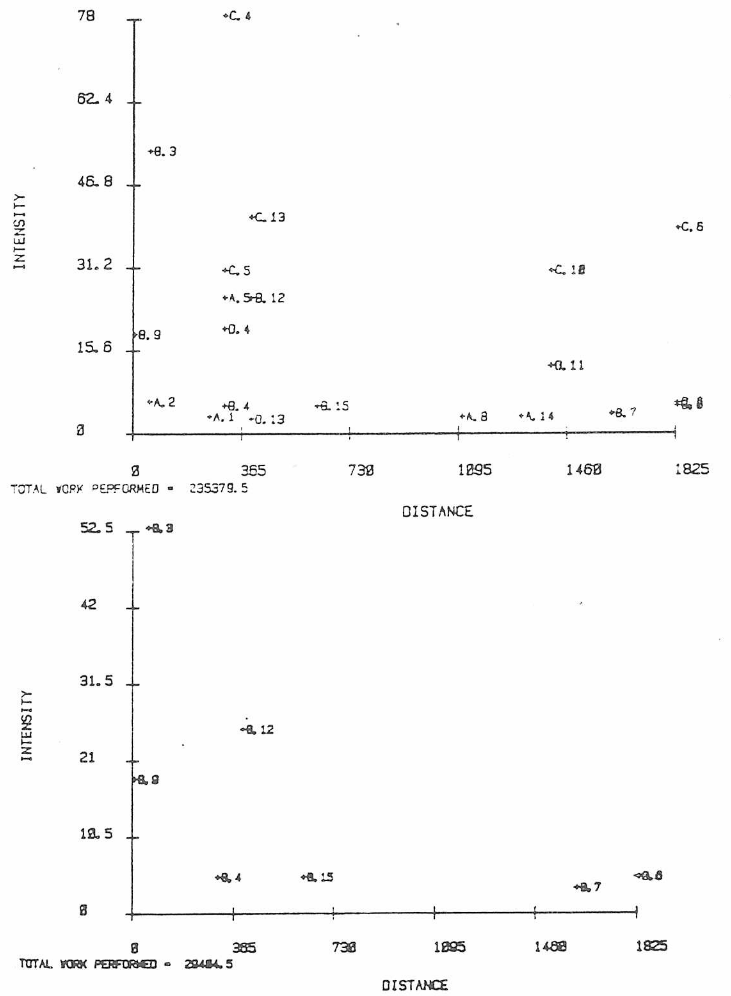


Fig.4.2 Intensity vs. Distance graph (after Hales, H.L)

This type of approach lends itself to computerisation. At its simplest this involves an algorithm to draw the graph from a set of input data. More sophisticated systems might automatically load relationship distances into the scoring routine from an interactive graphics layout program suite.

4.2.4 Clustering techniques

Clustering is an analytical tool which can be used in three ways:-

1. To define areas prior to layout planning.
2. To study relationships between activity areas.
3. To reveal the overall block sizes of a space for closely related activities - the cluster.

Input is usually in the form of results from questionnaires. Output is achieved with a mathematical technique known as hierarchical decomposition. The repetitive mathematics involved usually require the use of a computer for anything other than the simplest of situations. An example of this type of program is CLUSTR by ABACUS. (4.8)

4.2.5 Layout planning algorithms

"The field of computer aided layout planning began 20 years ago with the publication of the CRAFT layout algorithm by Elwood Buffa, Gordon Armour and Thomas Vollman." (4.9)

Traditionally layout planning algorithms such as STUNI (4.10) have worked by placing units of accomodation with the highest closeness rating first (ie. that element with the most and/or strongest relationships). The remaining activities are examined for their relationship to those already placed, with placement being in descending order of closeness desired to those activities already placed.

The layout can then be scored. Some sophisticated algorithms attempt to further improve on this layout by pairwise exchange of activity spaces which seek to achieve a better score.

4.3.0 EXISTING DOMESTIC ACTIVITY SPACE MODELS

So far in this chapter we have been discussing existing activity space models, as they apply to the wider field of facilities planning. Although these models have many principles which could be applied to the analysis of domestic activity spaces, in general they are geared for a more general form of layout planning. That is, deciding the area and location of say the typing pool, as opposed to the detailed layout of the 'pool' itself.

In the remainder of this section we will examine two domestic activity space models which will, to a certain extent, highlight this difference.

4.3.1 Genova system

At the time of writing (1985) the work of this group has not been published outside Italy (4.11) and the information presented here is dependent on translation from Italian.

Written in FORTRAN on a VAX 750 computer using a Tektronix 4012 as a display unit, their package is intended to aid the designer by allowing him to visually appraise the effects of his design decisions.

It adopts a whole house approach, breaking the layout design into rooms before allowing the designer to select from a standard menu items of furniture to be located within a single room.

A sample of the visual appraisal is presented below in figure 4.3. It can be seen that in many respects it represents a computerisation of the Svennar system.

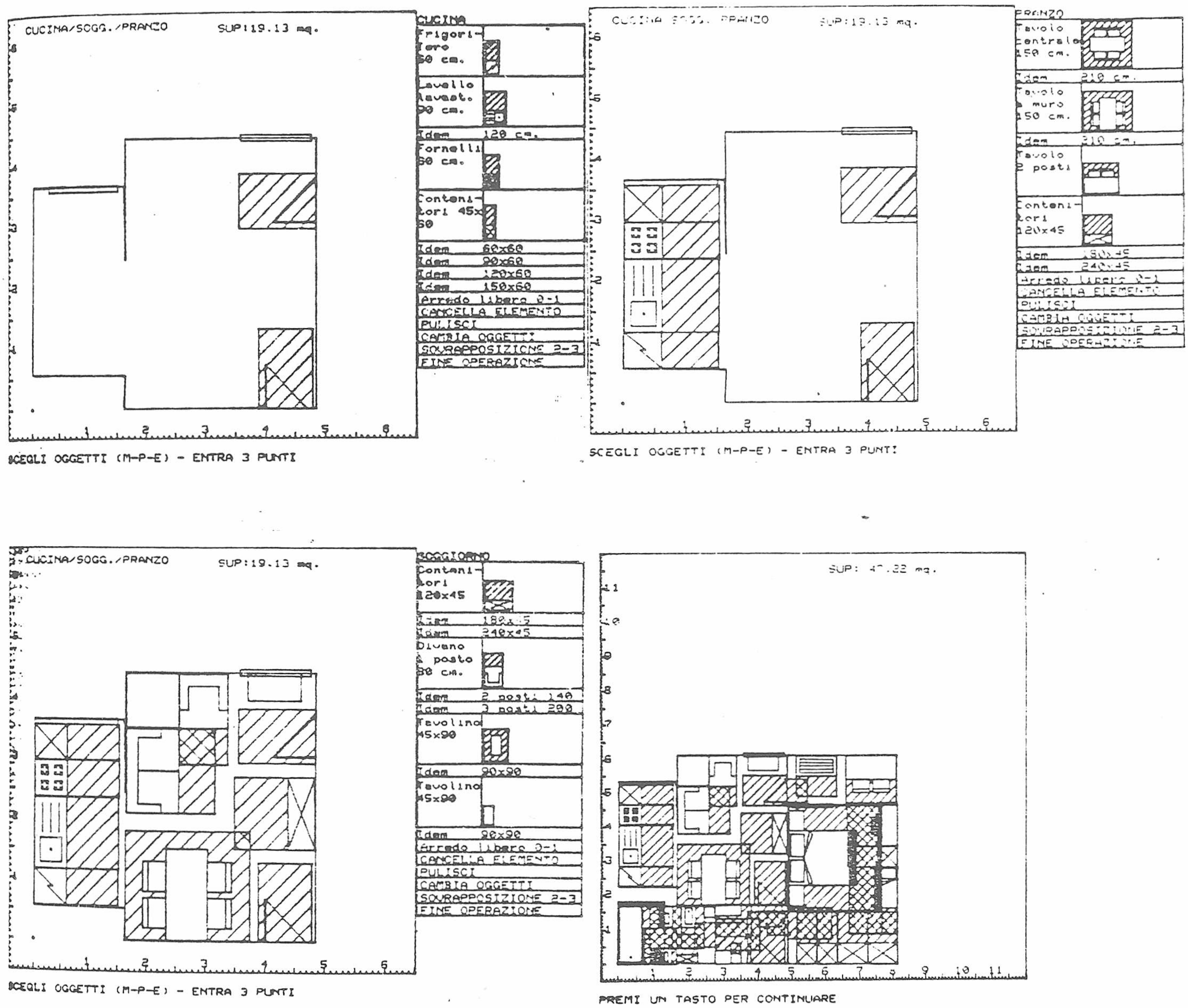


Fig.4.3 Genova system, an indication (after Gambro et al.)

4.3.2 Langskogs system

Langskog attempted to derive a model that was both bottom up, and top down, in as much as he saw the system that he developed within an overall framework of a building appraisal package. (4 12)

Although he defined this strategy, the only aspect of the model that was actually implemented on a computer (FORTRAN on a DEC 2050 using Tektronix 4010 for display) was his room appraisal package.

This analysed the layout of a room in terms of its two dimensional plan form. Furthermore all geometry was restricted to orthogonal, rectilinear forms.

At the core of his appraisal measures, were three penalty factors:-

1. A shape penalty.
2. An area utilisation penalty.
3 An overlap penalty.

An indication of the sort of appraisals possible with the Langskog system is given in figure 4.4 below.

These factors are more properly discussed in the following sections, since they are conceptually very similar to parts of the new model. The differences generally relate to the use of three dimensional,

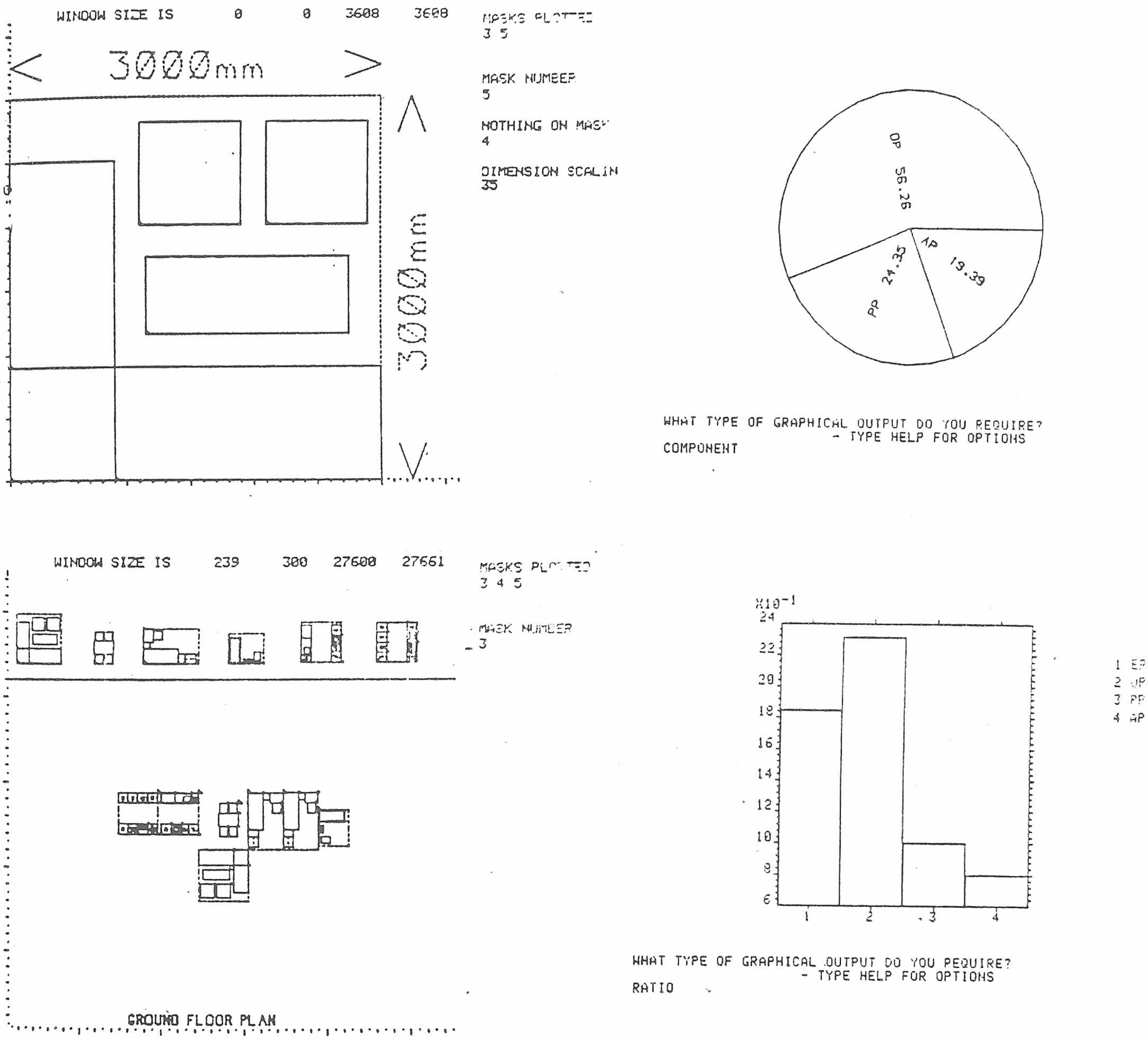


Fig.4.4 Langskog system, an indication (after Langskog)

non-orthogonal non-rectilinear geometry allowed by the new model. Futher details are given in Appendix D.

4.4.0 THE NEW MODEL

The last section described existing domestic activity space models. In this section we discuss a new model. In the next chapter we discuss its implementation on a computer and in the following chapter subjective experiments which seek a validation.

As was seen in the last chapter, existing domestic activity space models seek to appraise or score either numerically or subjectively, by graphic display, a given design.

The new model also follows this strategy for a variety of reasons. Firstly the personal design philosophy of the author would indicate that 'design' in the most creative sense of the word is not yet (and may never) be the province of the machine. Secondly, that the technology available today (in non-military establishments) is inadequate for the sophisticated model that would be required to encapsulate even the smallest part of the designers 'knowledge base'. These two reasons interact with the third, that appraisal or scoring of designs is one method of making design knowlege explicit in a form that might be useful in the future for sophisticated 'expert system' computer models. This concept is explored further in Chapter 8, Section 8.4.0.

Thus the philosophy for the new system was that of an automated layout scoring system backed onto an interactive graphics layout package, which would allow the designer to appraise any layout, or part layout of a room and tell the designer not only how "good" or "bad" the design was but would also indicate possible areas for improvement. It would then be up to the designer to accept or reject the computer's advice in order to "improve" the quality of the design.

It should be noted that "good" and "bad" are always in inverted commas because there are always factors (such as aesthetic appearance) outside the competence of the computer model which may make a "bad" layout good or vice versa.

4.4.1 The new model - an overview

The new model was constructed to accomodate three dimensional, non-orthogonal, non-rectilinear room and fixture geometries and to present an appraisal in terms of four measures of layout "goodness":-

1. A room space efficiency penalty (SP)
2. A volume utilisation penalty (VU)
3. An overlap penalty (OP)
4. An association penalty (AP)

As an overall measure these four penalties were averaged to give an overall efficiency ratio (ER).

These penalties are also discussed in Appendix D and in Appendix H.

4.4.2 Space efficiency penalty

This penalty is given by:-

$$\frac{P.H+2A}{4\sqrt{V/c}.c+2V/c}$$

where:-

P = perimeter length of room.
V = volume of room
c = a constant (set to optimal height for room - in this case 2300mm)
H = height of room.
A = area of room

As can be seen in figure 4.5, this penalty measures the efficiency of the room shape.
It is a variation on the volume compactness ratio, and allows differing room volumes to be compared.

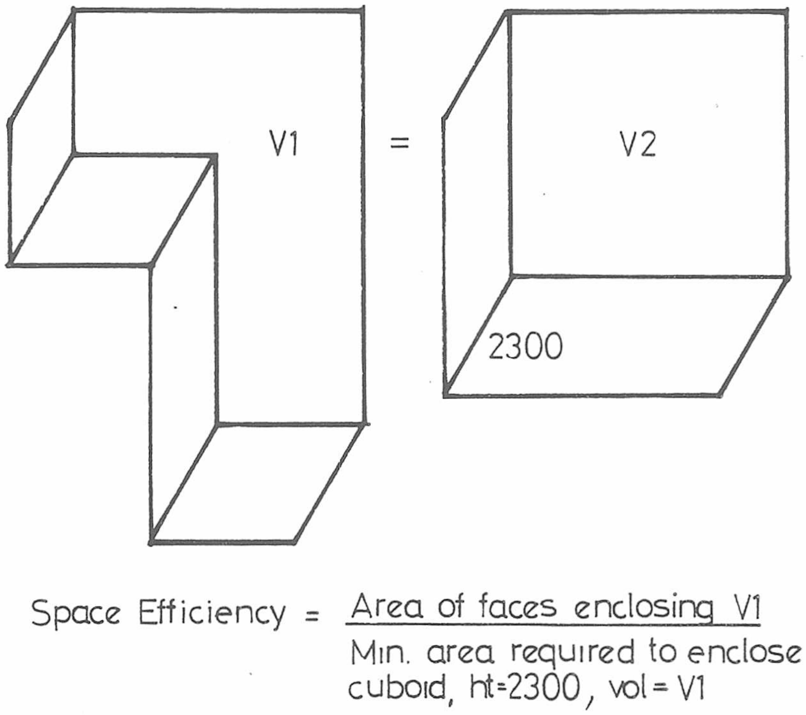


Fig.4.5 Space Efficiency Penalty

The penalty compares the designed room surface area to an ideal room surface area enclosing the same volume as that of the designed room.

In the traditional volume compactness ratio this would be a cube, but for large rooms this would mean an ideal with an unnecessarily high head height. Therefore, for the new model, the traditional ideal was distorted by fixing the ideal ceiling height at 2300mm, being a suitable height as regards the Scottish Building Regulations. (4.13) Naturally this height could be varied for other (non-domestic) applications of the model.

4.4.3 Volume utilisation penalty

This penalty is given by:-

$$\frac{TV}{\sum_{1}^{n}(REUmax) + \sum_{1}^{m}(FEUmax)}$$

where:-

TV = total volume of room
REUmax = maximum user vol. of a single room element
FEUmax = maximum user vol. of a single furniture element
n = no of room elements
m = no of furniture elements

This measure is intended to assess the compactness of the room layout, see figure 4.6

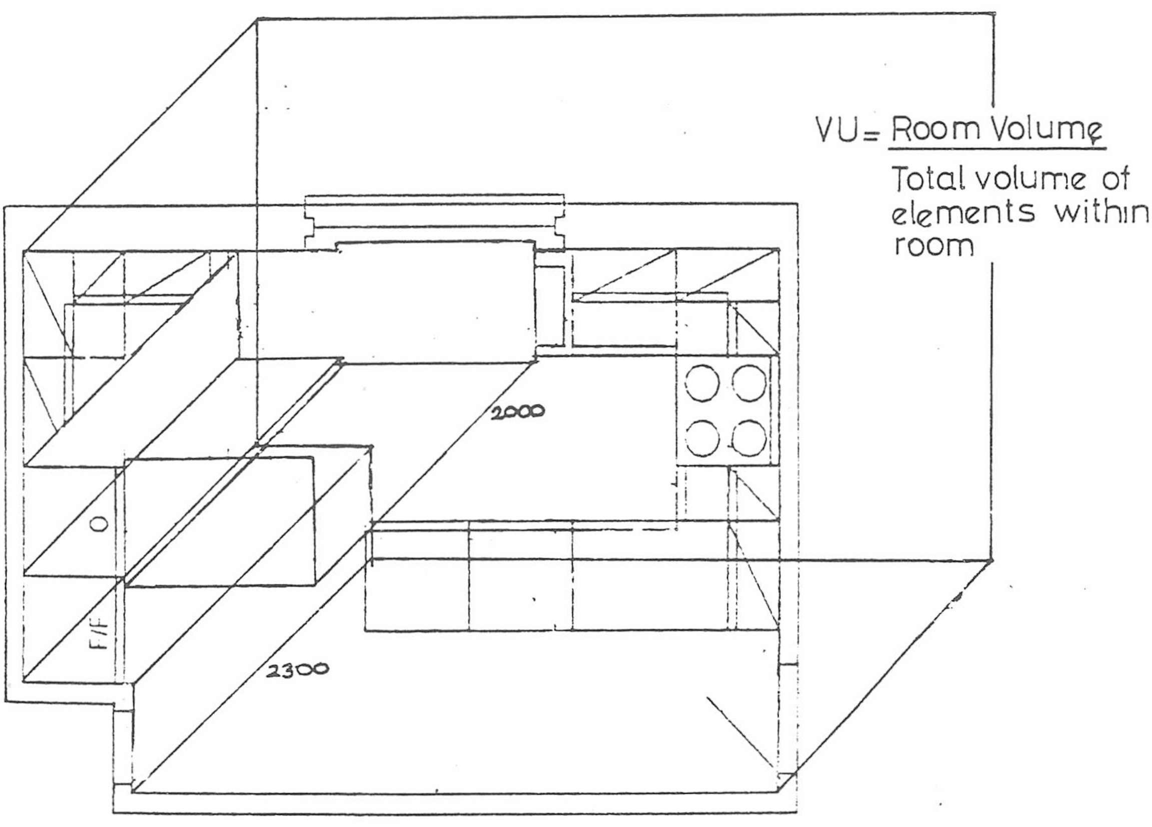


Fig.4.6 Volume Utilisation Penalty

The penalty assesses the ratio of minimum total enclosing volume required by the fixtures within the layout, to the actual volume available within the room. This means that rooms which are comparatively sparse would attract a greater penalty than those which are comparatively full.

4.4.4 An overlap penalty

This penalty is given by:-

$$\frac{\sum_{1}^{n}(OVr.Pr+\sum_{1}^{n-1}(OVns.Pns))+\sum_{1}^{m}(OVr.Pr+\sum_{1}^{m}(OVns.Pns)+\sum_{1}^{n-1}(OVms.Pms))}{\sum_{1}^{n}VR+\sum_{1}^{m}VF}$$

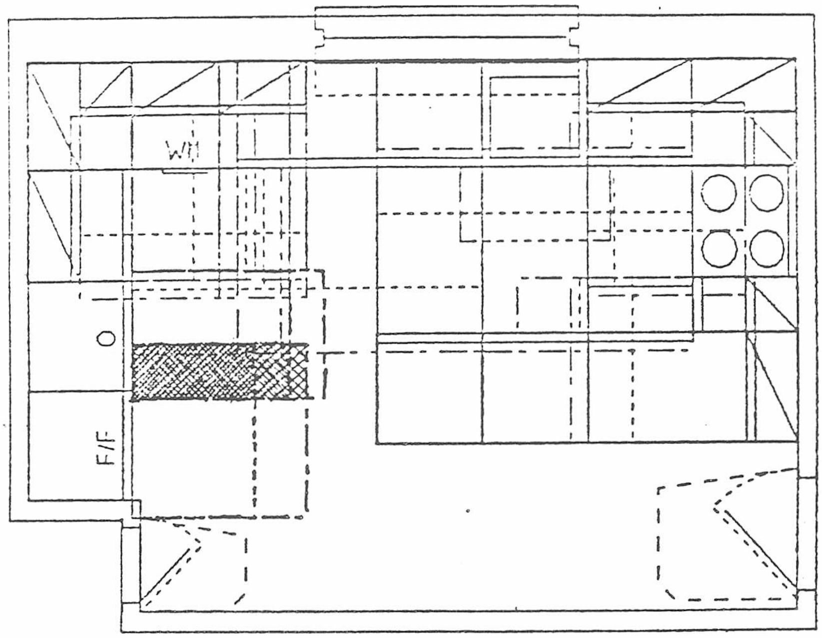
where:-

VR = volume of an individual room element
VF = volume of an individual furniture element
m = no of furniture elements
n = no of room elements
OV.P = overlap value for a discrete pair of conflicting elements multiplied by the weighting factor for those elements
r = subscript donoting room edge being used as one of the conflicting elements
ns = subscript donoting that a room element is one of the conflicting elements
ms = subscript donoting that a furniture element is one of the conflicting elements

See figure 4.7 for a diagram of the overlap penalty.

This measure is intended to measure the congestion within a room layout.

Each fixture has three layers of information associated with it and used within the numeric analysis of the model. Firstly, there is a description of the physical limits of the fixture. Secondly, there is a minimum user space requirement, ie. the amount of space surrounding and including the first order description, but also including the



$$\text{Overlap penalty} = \frac{\text{Total weighted volume of overlap}}{\text{Total fixture volume within room}}$$

Fig.4.7 Overlap Penalty

minimum amount of space required for use by a normal able-bodied adult. The third layer adds the additional volume required by elderly users. The third order volume would allow the able-bodied adult to use the fixture more easily.

These layers are demonstrated in Chapter 2, Section 2.3.3.

Obviously, first order overlaps are not physically possible. (Should the designer accidentally do this whilst designing a layout, the model will penalise it heavily and when interogated point out the error.) Second and third degree overlaps, including overlaps with the the room edge and room fixtures such as doors and windows, attract a penalty based on the volume of overlap and the order of overlap - the weighting factor. That is, two second order volumes interpenetrating attracts a higher penalty than two third order volumes.

The penalty measure totals all these weighted overlaps and forms a ratio of comparison between that figure and the total volume (ie. third order volume) of all fixtures, including room fixtures such as doors and windows, within the layout.

This means that room layouts with fewer or less serious overlaps, ie. less congested layouts, have a lower overlap penalty value.

For futher details of the mechanics of the overlap algorithm see Appendix F.

4.4.5 An association penalty

This penalty is given by:-

$$\frac{\sum_{1}^{n}(\Delta XLap.W)}{\sum_{1}^{m}(\Delta Lap/2)}$$

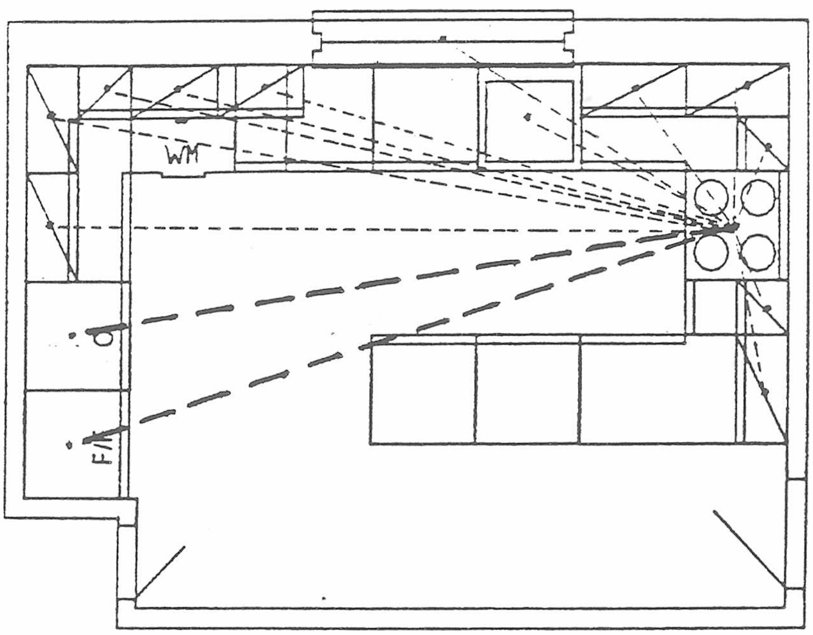
where:-

n = no of failed association pairs
m = no of possible association pairs
ΔXLap.W = weighted excess length of failed association pair
ΔLap = length permitted between minimum and

maximum distance apart for a possible association pair

See figure 4.8 for a diagram of the association penalty.

Fig.4.8 Association Penalty



$$\text{Association penalty} = \frac{\text{Total failed assoc. weighted distance}}{\text{Total possible assoc. mean distance}}$$

This measure is intended to assess the workability of a room layout based on the relationship between fixtures.

Any fixture which should relate to another has an association distance for that relationship pair fixed at the outset by the designer. (See Chapter 2, Section 2.3.2 for an example.)

This penalty measures the ratio of weighted failure distances over possible mean distances (ie. over the mid-range distance of each possible association pair).

This allows the value of the penalty to be adjusted according to the number of relationships to be satisfied. The lower the association value, the more relationships have been satisfied and presumably the more workable the layout.

4.4.6 The efficiency ratio

The four penalties described in Sections 4.4.2 to 4.4.5 are averaged to give an overall efficiency ratio.

This provides a convenient summary of the four previously described measures, but should never be used in isolation since by its nature it tends to mask particularly "good" or "bad" aspects of any layout.

It may be that a straight unbiased average of the four penalties is not suitable for deriving an overall efficiency ratio and that instead some form of weighting should be given to one or more of the penalties. This topic is considered further in

Chapter 6. The reader is particularly directed to page 149A

4.5.0 SUMMARY

In this chapter we have outlined some of the structured techniques, both manual and computerised, used to date to aid facilities planning.

We have examined briefly two domestic layout appraisal systems and presented a new volumetric and associative model. This model can be seen to have been formed around two distinct concepts.

Firstly, it has adopted the conflict of user areas concept, both as a visual design aid (after Svennar and others) but also as a means of numerically assessing congestion.

Secondly, it adopts the concept of association distances between related activities (in this case individual fixtures) proposed by Muther and others, not as a means of automated incremental design but as a means of numeric interpretation of the workability of a layout.

As far as the author has ascertained, these two concepts have never before been combined within one model.

This chapter has discussed the new model in abstract terms. Later chapters discuss the model either as a computer implementation or as it has been applied to the test-bed situation of the domestic kitchen.

4.6.0 REFERENCES

(4.1) Hales, H.L., Industrial Engineering 9, Computer Aided Facilities Planning, Marcel Dekker Inc., New York, USA, 1984, p.79

(4.2) Hales, H.L., op.cit, pp.80,81

(4.3) ibid.

(4.4) Langskog, K., A Computer Aided Architectural Design System for the Analysis, Synthesis and Appraisal of Domestic Activity Spaces, Ph.D. Thesis, Scott Sutherland School of Architecture, Aberdeen, 1981. pp.63,66

(4.5) Langskog, K., op.cit, pp.84,90

(4.6) Langskog, K., op.cit, pp.71,79

(4.7) Muther, R., Hales, H.L., Systematic Planning of Industrial Facilities, Vol I and II, Kansas City, M.O., Management and Industrial Research, 1980.

(4.8) Bridges, A.H., A New Approach Towards the Computer Aided Analysis of Architectural Design Problems, Proceedings of CAD 78, Science and Technology Press, 1978, pp.443,460.

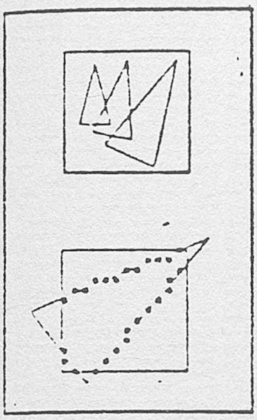
(4.9) Hales, H.L., op.cit, p.96

(4.10) Th'ng, R., Davies, M., ABACUS Occasional Paper 23, SPACES 2 A Program Package for Producing Sketch Layouts, University of Strathclyde, 1972.

(4.11) Gambro, G., Gambro, P., Pienovi, C., Un Sistema CAAD per la Verifica della Qualita' Residenziale, Istituto per la Matematica Applicata, Genova, N.163(84), 1984.

(4.12) Langskog, K., A Computer Aided Architectural Design System for the Analysis, Synthesis and Appraisal of Domestic Activity Spaces, Ph.D. Thesis, Scott Sutherland School of Architecture, Aberdeen, 1981.

(4.13) HMSO, The Building Standards (Scotland) Regulations 1981, HMSO, London, 1981, Amended 1982,1984.



CHAPTER 5

DYNAMIC GRAPHICS IMPLEMENTATION

CONTENTS

5.1 0 INTRODUCTION

This chapter describes the philosophy for the implementation of the activity space model described in chapter four. The philosophy takes cognisance of some of the problems identified in section 3.4.0.

5.2.0 EXISTING INPUT METHODOLOGIES

As has been noted earlier in this research project (5.1), existing input methodologies use three distinct techniques for entering geometry data.

1. Keyboard input
2. Screen input via crosshair digitising
3. Tablet digitising

Keyboard entry of coordinates to describe building geometry must be regarded as a user unfriendly methodology, since it involves the designer in a great amount of key punching as well as forcing him to think about geometry in an unnatural way, ie. in terms of numbers.

Discarding this option, leaves two possible methods for implementing the proposed activity space appraisal model:- screen digitising and tablet digitising.

The main advantage to the designer of using a digitising tablet is that existing drawings can be laid over the surface of the tablet and digitised relatively quickly. This means that the tablet is of minimal interest during the design process, since at that stage the designer's interest is to create the original drawing.

Other benefits of using the tablet such as its resemblance to a traditional drawing board, which may or may not reduce the psychological inhibitions which the designer may feel about using the computer, are more than offset by the need to transfer his attention between the input device, the tablet, and the output device, the screen.

It should be noted however, that this last disincentive may not hold true when more automated systems of describing building geometry are developed (5.2).

This narrows down the choice of input methodologies to screen digitising using either thumbwheels, joystick, mouse or trackerball.

Stepping keys can be discarded as an input device since they unnecessarily restrict the user's ability to select any line length, angle or position; whilst the light pen can be discarded due to its poor ergonomics for prolonged use.

As to which of the other digitising devices are used for input - it really is a matter of availability and personal preference, since each achieves the same end in a very similar manner.

5.3.0 REFRESH GRAPHICS VS. STORAGE TUBE

There are two types of display screen available for use, the storage tube display and refresh graphics display. The storage tube is generally cheaper and more precise whilst the refresh graphics is more flexible.

In the storage tube image vectors are drawn once by the computer on the screen. The screen then retains the image until the whole screen is cleared. With refresh graphics, the image is drawn and then redrawn on the screen.

This means that the essential advantage refresh graphics has over the storage tube display is the capability to have images that can be moved, modified or erased without erasing and updating the entire screen.(5.3) Another advantage, and in some applications a greater advantage, is that the storage tube is limited to monochrome display whilst refresh can be in colour.

At first sight this ability to move, modify or erase images without erasing and updating the entire screen seems comparatively insignificant. However, further examination reveals that it allows significant improvements in the quality of the man/machine interface.

The possibilities of refresh graphics improvement in the quality of the man/machine interface are identified as being in three key areas:-

1 Feedback

2. Line rubber banding

3. Object moving

These topics are discussed further below.

5.3.1 Feedback

Utilising refresh graphics capabilities it is possible to display a greater number, and more complex, feedback messages, so as to inform the user of what the computer expects; or of what the computer has done.

Although user messages are used, and in some cases, quite widely used with storage tube displays - it is only really possible to do this at a point in the application program when a screen redraw is necessitated anyway. This is not the case with refresh graphics, as will be made clear in section 5.4.0.

5.3.2 Line 'rubber banding'

With refresh graphics it is possible to employ a technique known as 'rubber banding'. To give an example of this technique; the start point of a line is fixed either alphanumerically or through digitisation, the end point is then indicated using a cursor to point on the screen. Meantime a refresh vector is drawn between the start point and the current cursor position (see figure 5.1 below). This means that the user has a direct visual appreciation of the line that is being drawn, rather as he would when using pencil and paper.(5.4)

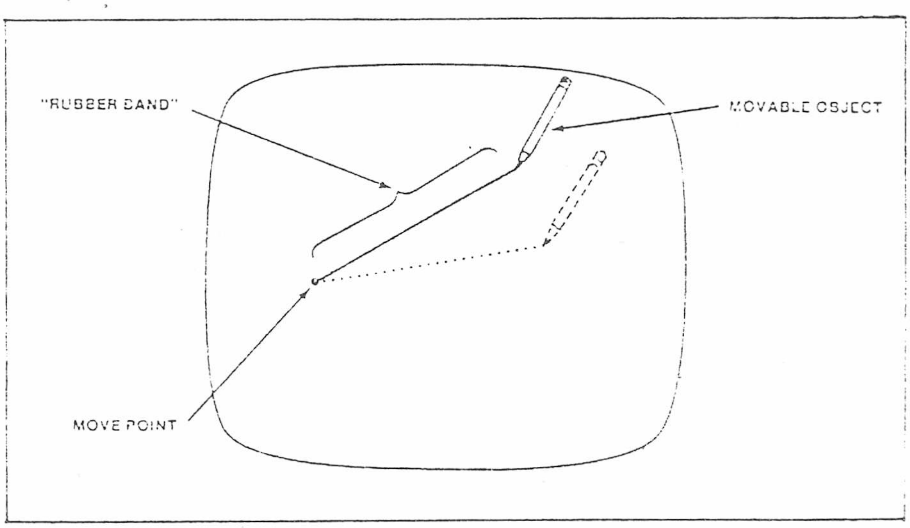


Fig. 5.1 Rubber Band Effect

5.3.3 Object moving

Essentially this means that it becomes possible with refresh graphics, to 'pick up' an existing displayed image and move it dynamically across the screen locating copies of the image as and where required. This is illustrated graphically in figure 5.2 below. (5.5)

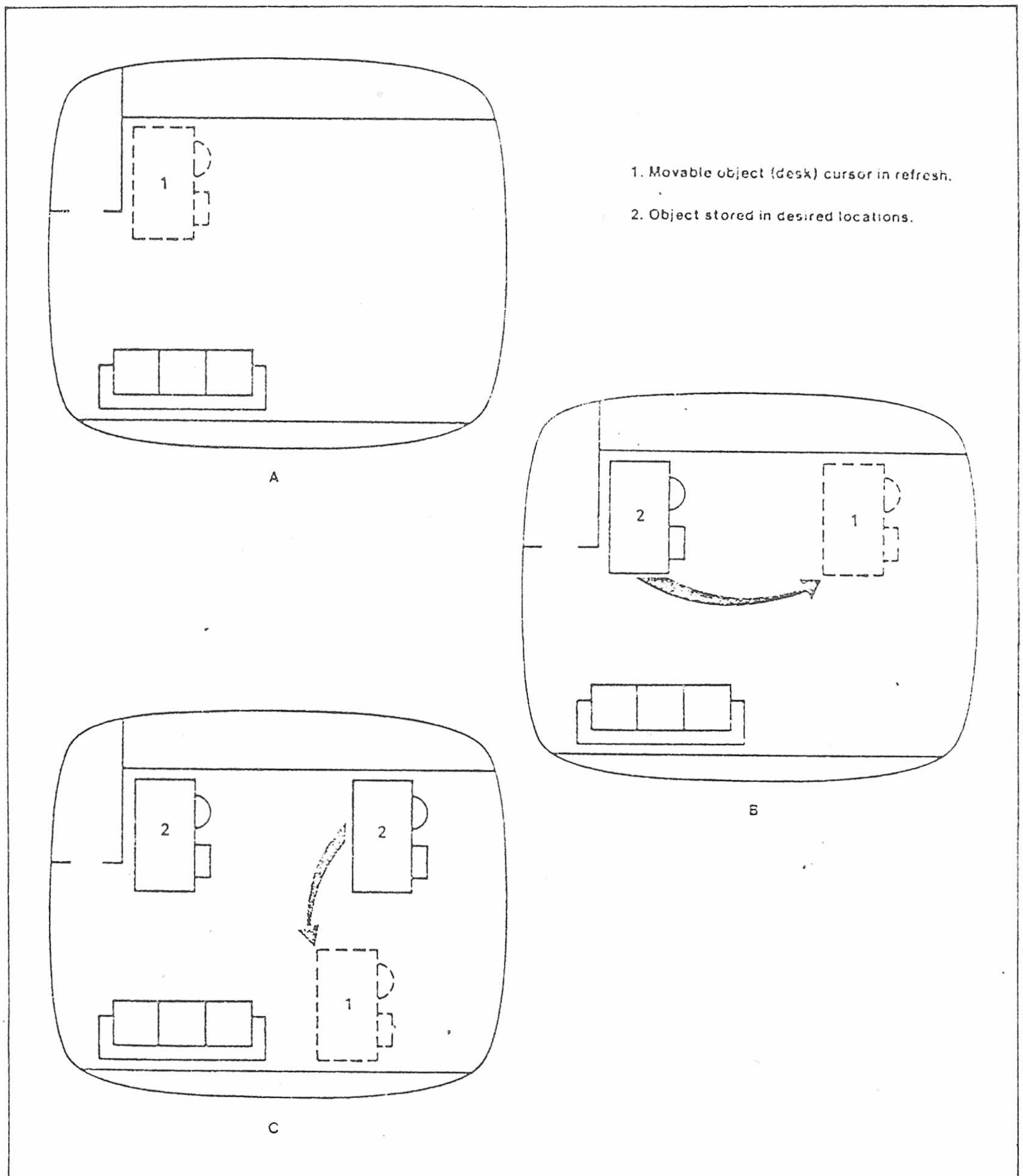


Fig. 5.2 Object Moving

5.3.4 Refresh graphics technology

Having established the flexibility and versatility of refresh graphics capabilities for improving man/machine communications; it is worth noting in passing that two types of refresh graphics screens are available. They are raster scan displays and vector draw displays. The essential difference between these two types of refresh display is exemplified in figure 5.3 below.(5.6)

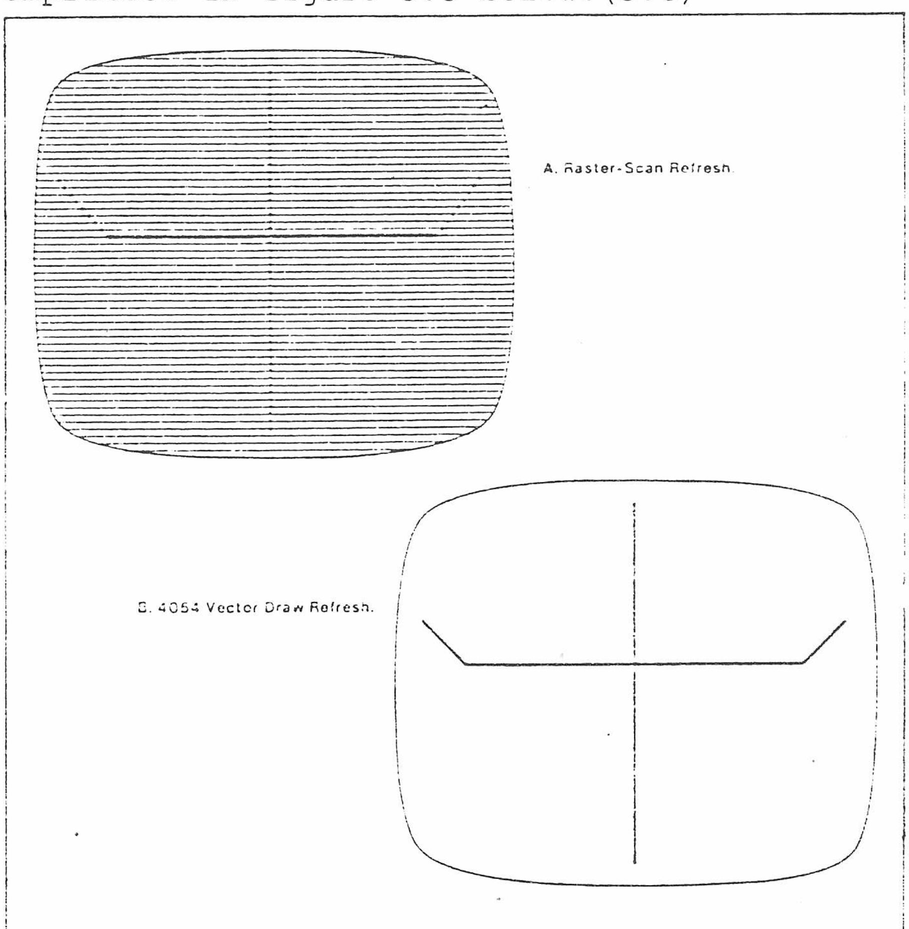


Fig. 5.3 Raster Scan vs. Vector Refresh

Raster scan employs a technique similar to that of the television screen and as can be seen above, the detail of the image can be limited by the gaps between horizontal scanning lines. This makes it less suitable for architectural work.

A much sharper and more detailed image is available utilising vector draw refresh. This is because vector draw refresh, as its name implies, draws each vector in a manner similar to that of the storage tube.

5.4.0 INTERFACING

Examination of the possibilities afforded by the use of refresh graphics determined that this type of hardware should be adopted. Fortunately, a Tektronix 4054 desk-top micro with refresh graphics was available at Scott Sutherland's. As a consequence of adopting this input hardware, a decision was made to make maximal use of the refresh capabilities, as outlined in section 5.3.0.

5.4.1 User feedback

Particular attention was paid to utilising the capability for feedback because as Newman & Sproull have said, "Feedback is often overlooked as a component of the user interface."(5.7)

An early decision was made to fully employ refresh graphics techniques to provide the following forms of feedback. (see fig.5.4)

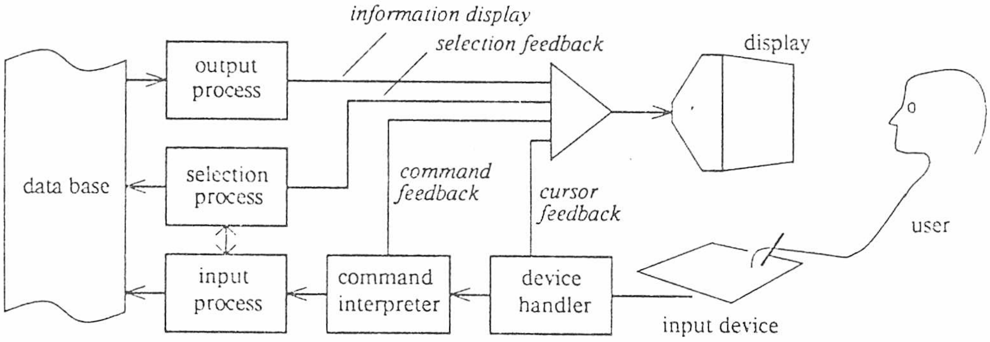


Fig. 5.4 User Feedback

> "1. Feedback from the command interpreting process, informing the user whether the command has been accepted, what stage of execution has been reached, and whether an error condition has arisen.
>
> 2. Feedback from the application data base - principally selection feedback.
>
> 3. Feedback unrelated to command interpretation or the data base: cursor feedback, character echoing etc." (5.8)

All the above forms of feedback can be found throughout the software that was developed (KAPABLE), particularly in the LAYOUT program suite as described in chapter seven.

5.4.2 Continuous feedback

This form of feedback is mentioned seperately because although it uses only those forms of feedback mentioned in the earlier sub-section, it combines them in such a way that the whole is greater than the sum of the parts. The concept is best explained with the aid of a diagram. See fig.5.5 below

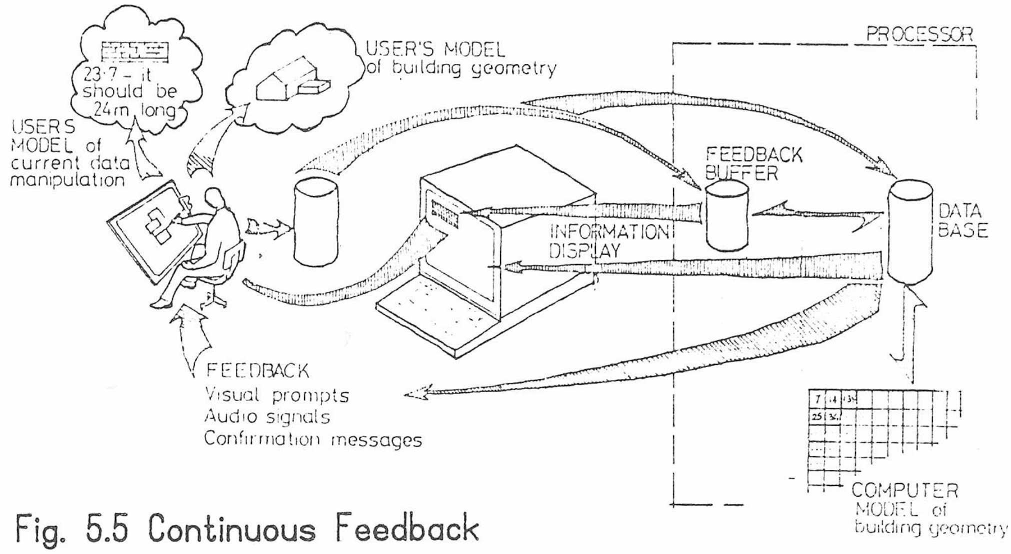


Fig. 5.5 Continuous Feedback

What the above diagram tries to show, is that whilst the user is inputing data to the data base he receives feedback as to what the computer 'thinks' he wishes to do. The user then uses this feedback to refine and redefine his input to the point where the computer is accurately interpreting his intentions.

To illustrate with a simple example, suppose the user intends to create a new line in his database. He starts by using the cursor to indicate the start point of the line. Before this point is fixed into the data base, feedback is given to the user indicating alphanumerically (or perhaps even graphically) the relationship between the point indicated and other existing lines within the data base. The user can then use this feedback as a basis for accurately positioning his input.

This process could then be repeated for an end point evaluation, except that additional computer generated information would be available describing the 'new' line.

Only when the user is entirely content with his input is the necessary data finally entered into the data base.

This is a very powerful form of feedback and requires considerable processing power in the computer. In addition, the 'intelligence' aspects require quite large amounts of decision making software. As a consequence, it was impossible to make full use of this form of feedback in the LAYOUT suite of programs, however a prototypical implementation can be found elsewhere in KAPABLE, notably in those program segments concerned with creating the room outline and with creating individual objects.

5.5.0 ADVANTAGES/DISADVANTAGES OF IMPLEMENTATION ON A MICRO-COMPUTER

This sub-section examines the consequences of implementing the activity space model on a 64k micro-computer.

5.5.1 Disadvantages

1. Capacity

   The most obvious disadvantage is the lack of core memory. With so little core memory available, software segments must be carefully structured, particularly when perhaps 10k is required for the dynamic graphics display and perhaps another 10-15k for variables.

2. Speed

   This has two aspects, processor speed and file handling speed. Taking processor speed first, a micro lacks sheer processing power. Calculation intensive tasks take much longer on a micro, than they would on say a mainframe. This disadvantage is particularly evident when the designer is interpreting his room layout.

   File handling between computer and disk drive is also comparatively slow, despite the fact that steps were taken to reduce file search time. In particular disk mounting seems to take an average of about one minute. This greatly slows down the process of copying files from one disk to another.

5 5.2 Advantages

I. Control

It is much easier for a computer illiterate to program using a micro since the operating system is generally much simpler than for a mainframe.

2. Reliability

Although this may not be generally true, the author found his micro to be less prone to breakdown and failure than the Institute mainframe.

3. Structuring

An important benefit of using a micro was that because of its limited memory the software designer is encouraged to stucture and order his software. A micro is likely to be less tolerant of inefficient programing than a mainframe.

5.6.0 CONCLUSIONS

Use of refresh graphics as outlined in this chapter proved to be an exceptionally powerful input mechanism, allowing greater 'intelligence' to be programmed into the input interface. It also considerably enhanced the screen display.

However for this application, lack of 'number crunching' power is a serious limitation, particularly when the designer is interpreting the design within the LAYOUT* segment (see Chapter 7). It becomes impossible to use the INTERPRETATION* segments interactively because of the excessive time overhead.

Other aspects of 'lack of speed' evidenced for example when copying files, could be significantly alleviated by the introduction of more disk drives.

The author attempted to overcome some of the worst effects of this lack of speed, for example, the QUICK HIGH OP* routines described in Chapter 7, Section 7.4.4 overcame the time overhead problem associated with the HIGH OP* routines, albeit with a degradation in the quality of design feedback given to the user.

Although it would certainly be possible to reduce run-time through the INTERPRETATION* routines by rewriting the algorithm to reduce even further unnecessary calculation; it is unlikely, in the authors opinion, to reduce the run time by the factor of ten required to make interactive use of the INTERPRETATION routines a possibility.

* Note: LAYOUT, INTERPRETATION and QUICK HIGH OP refer to segments of the developed software and are fully explained in CHAPTER 7.

Thus it has to be said that although the Tektronix 4054A has excellent graphics handling capabilities, supports a very good quality of BASIC and has excellent 'back up' in the form of manuals (allowing a non-computer specialist to deign and implement a significant system) it has insufficient 'power' for an application, such as the implementation of the overlap penalty as defined in Chapter 4 where large amounts of consecutive calculation are required.

It was, and is however, a very good development tool.

5.7.0 REFERENCES

(5.1) Hammond B., Techniques for the Input of Geometry Data in Computer Aided Architectural Design, Occasional Paper No 11, Scott Sutherland School of Architecture, Robert Gordons Institute of Technology, Aberdeen, 1983. pp 14-26.

(5.2) Leifer, D. The Machine Refinement of Raw Graphic Data for Translation into a Low Level Data Base for a CAAD Decision Matrix, (Ph.D Thesis), Aberdeen, Scott Sutherland School of Architecture, Robert Gordons' Institute of Technology, 1984.

(5.3) Tektronix _4054 Options 30 and 31 Dynamic Graphics Operators Manual_, Tektronix Inc. Oregon, USA, Oct. 1981, pp 1-3,1-14.

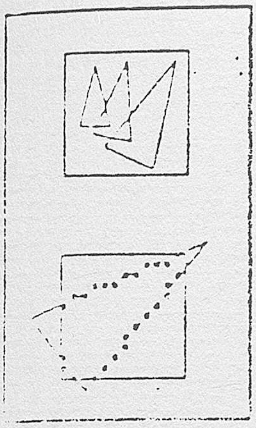
(5.4) _Ibid_

(5.5) _Ibid_

(5.6) _Ibid_

(5.7) Newman, W., Sproull, R., _Principles of Interactive Graphics_, McGraw Hill, New York, 2nd Ed, 1979, pp 463,464.

(5.8) _Ibid_



CHAPTER 6

EVALUATIVE EXPERIMENTS

CONTENTS

6.1.0 INTRODUCTION

This chapter describes three seperate experiments conducted to validate the new model for appraising domestic activity spaces.

For the sake of brevity and clarity, much of the detail associated with these experiments has been consigned to Appendix G.

6.2.0 Experiment 1

This section provides details of the first validation exercise.

6.2.1 Aim of the experiment

The aim of the experiment was to validate the new model by a process of comparative evaluation. That is, by comparing subjective designer assessment to objective computer appraisal.

6.2.2 Experimental precursor

This experiment arose from a small 'pilot study' which was carried out (almost for fun) when the numerical accuracy of the model was being checked (see Appendix H).

The 'pilot' consisted of five kitchen layouts assembled by the author (see end of Appendix D) on the computer. Photocopies of these layouts were then distributed to several research workers in the department. The workers were asked to rank order the kitchens from 1-5 (1 = best, 5 = worst) in terms of how efficiently they made use of the available room space. Their rankings were compared to that of the computer model - the ER value. A good correlation was found.

Although this was not a true pilot study due to the limited number of respondents and alternative kitchen layouts, not to mention the lack of scientific rigour used within the test, it did indicate that the concept of rank order correlation testing against model values could be a valuable approach to validating the model.

6.2.3 Experiment design

The experimental design was as follows. A questionnaire consisting of 20+ different kitchen layouts was assembled and presented to several discrete groups of designers.

1. Students at Scott Sutherland School of Architecture.
2. Teaching staff at the same school.
3. Professional kitchen designers throughout the Grampian Region.

The response, in the form of a rank ordering of designs from each group, was to be examined for a consensus and compared to the rank ordering of the computer and numerical evaluations for each kitchen design.

6.2.4 Questionnaire design

The raw data for the questionnaire, in the form of kitchen designs, came from professional kitchen designers.

Each kitchen designer listed in the local Yellow Pages (6.1), (see list of names and adresses contained in Appendix G, Section G.2.1), was circulated with a request for copies of domestic kitchen drawings. The response was varied, but some 30 kitchen designs were obtained.

Each kitchen was then examined for size, shape and fitment function and the manner in which these aspects related to the menu of kitchen fixtures already devised, (see Chapter 2, Section 2.3.3).

Some of the kitchens were discarded because they were too big, too small or contained oddly dimensioned 'specials'. In the event 21 kitchens were left to be examined in more detail.

The remaining kitchens were entered into the computer model. Three different menu combinations, each with some sixteen fixtures, were required. The menus were essentially similar barring minor variations such as the substitution of an upright fridge/freezer for a fridge.

Some of the room geometries underwent slight modification so as to enable them to be appraised by the computer model. (See Chapter 2, Section 2.3.1). Had the computer been more powerful, then each room would have required less modification because a more powerful menuing facility could have been incorporated in the model.

After each kitchen layout had been assembled and interpreted, a photocopy reduction of the computer display was made. The photocopies were then shuffled and assembled into a questionnaire, (see Appendix G, Section G.2.1 for a copy).

The questionnaire contained explicit instructions as to how it was to be completed together with additional information, such as fixture heights, which could not readily be described by a plan image.

Two sets of rank orderings were requested:-

1. Least area (1 = least, 21 = most)
2. Most efficient (1 = most, 21 = least)

The area ranking was included for the benefit of the second year students to ensure:-

1. They could 'read' a plan.
2. They could visualise space, at least in two dimensions.

It was thought that the degree of non-agreement found in the area rank ordering would give some indication of the 'error' factor in the second part of the questionnaire.

Teaching staff and professional kitchen designers were exempted from completing the area ranking exercise by a covering memo.

The first of the three groups to complete the questionnaire was a tutorial class consisting of nine second year architectural students 'borrowed' from a lecturer for an hour. In this time they had to complete both parts of the questionnaire. As an incentive to ensure that their best endevours were employed, two bottles of wine were offered as prizes to the individuals whose rankings most closely matched that of the computer. Seven students completed the questionnaire within the time alloted; the other two completed it in their own time later in the day.

Seventeen teaching staff were issued the questionnaire through the internal mail system and asked to complete it in their own time. Two members of staff did so.

Professional kitchen designers who had provided the original kitchen designs were telephoned to determine whether they would be willing to spend an hour completing the questionnaire. All readily agreed. As a result ten questionnaires were sent to nine businesses. No questionnaires were returned.

6.2.5 Results and evaluation

Tables of summarised results can be found in Appendix G.

As can be seen in figure 6.1 below, the students were quite capable of reaching a close measure of agreement as regards the rank ordering of room areas.

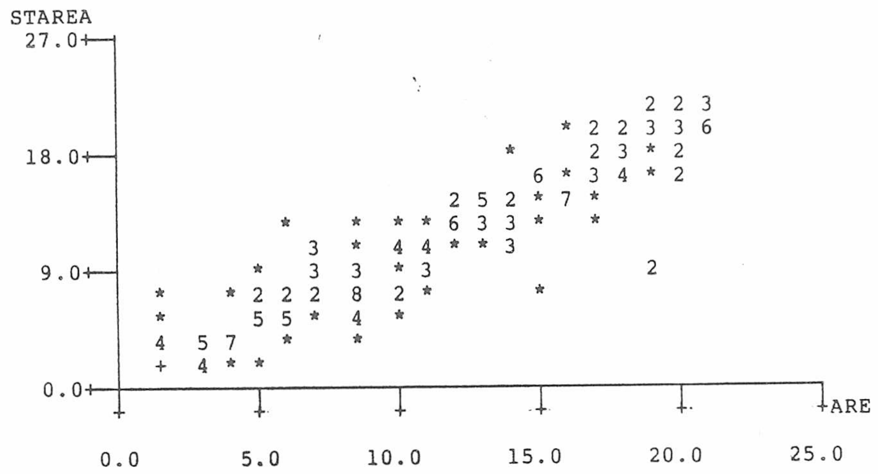


Fig. 6.1 Scatterplot of student area results against ranked calculated area of room

This is borne out by a high cross-correlation with the numerical evaluation of the room areas which averaged at 0.935 (6.2) where 1 indicates an absolute correlation and -1 indicates an inverse relationship.

However the scatterplots of efficiency ranking against ER appear completely random. There is no correlation with the ER value, see figure 6.2 below.

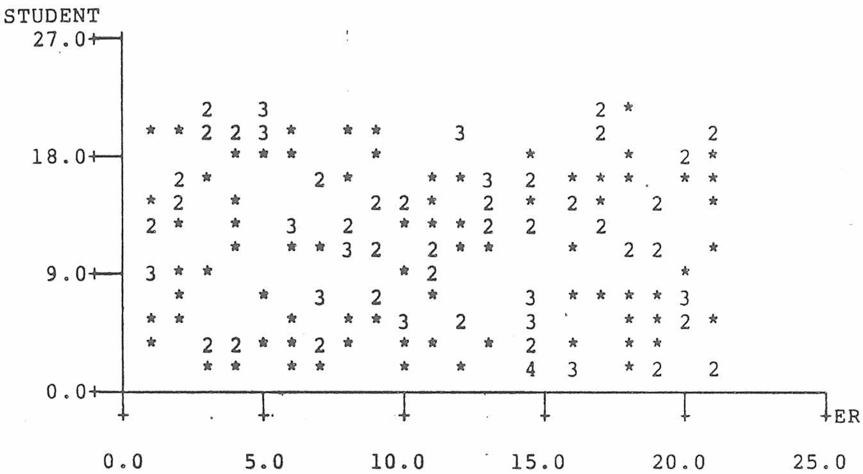


Fig. 6.2 Scatterplot of student results against ranked efficiency ratio values

Furthermore there is no appreciable correlation with any of the other penalties, see fig. 6.3 and Appendix G.

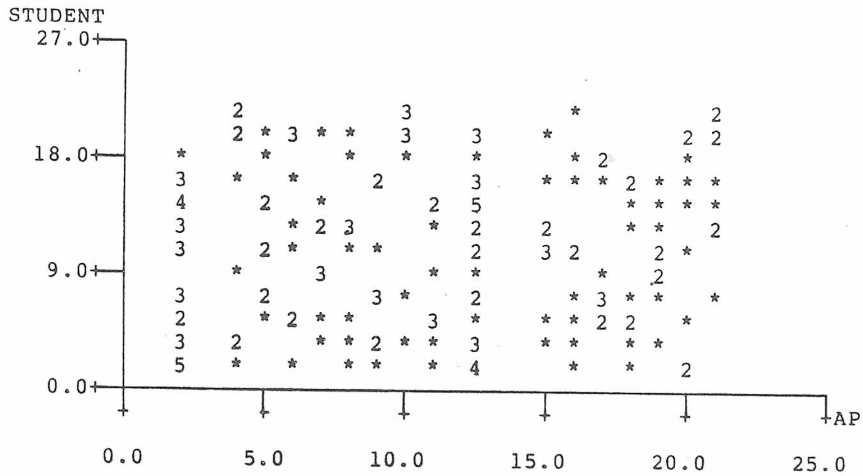


Fig. 6.3 Scatterplot of student results against ranked association penalty values

There were insufficient returns from the teaching staff to be statistically significant, nevertheless the results are included in Appendix G.

The results that were available would seem to indicate that the range of results was likely to mimic that of the second year students.

No returns were obtained from professional kitchen designers - so no analysis was possible.

6.2.6 Conclusions

By any standards this experiment was less than a complete success.

In the first instance, it was dogged by a low questionnaire response. The reasons for this were various:-

1. The questionnaire was large and complex. It took a considerable amount of time and effort to complete. This was probably sufficient to deter professional kitchen designers who were being asked to do 'something for nothing'.

2. The questionnaire was sent out at a particularly busy point in the academic calendar (just before the easter recess) and at a time when academic staff had up to four other questionnaires to complete.

3. Students had enough to do already without volunteering for extra.

In terms of the response of the students to the questionnaire, not only could no positive correlation be found with the computer model evaluations of space efficiency, but no consensus could be established in their response (see figure 6.4 below).

	ER	STUDENT	C2	C3	C4	C5	C6	C7	C8
STUDENT	-0.134								
C2	-0.164	0.112							
C3	-0.217	0.148	-0.295						
C4	-0.087	-0.126	-0.215	0.297					
C5	-0.099	0.403	0.342	-0.306	-0.317				
C6	-0.084	0.462	0.322	-0.252	-0.381	0.622			
C7	-0.407	0.474	0.495	-0.129	-0.309	0.759	0.511		
C8	0.106	0.469	-0.141	0.006	-0.036	0.435	0.321	0.146	
C9	0.153	0.125	-0.357	0.165	0.214	-0.591	-0.156	-0.559	-0.156

Fig. 6.4 Table of cross-correlation of student ranked efficiency results

Although one or two students had a degree of correlation to each other (approx. 0.4); this is to a certain extent explained by the seating arrangement when completing the questionnaire. There certainly is no consensus across the whole tutorial group.

To some extent this is not surprising since it confirms previous experimental results (6.3) (6.4).

6.3.0 EXPERIMENT 2

This section provides details of the second validation exercise.

6.3.1 Aim of this experiment

The aim of this experiment was to validate the new model by simple comparative evaluation. That is, by comparing subjective designer assessment to objective computer appraisal, within a situation where few design decisions have to be made.

6.3.2 Experimental precursor

Experiment 1 acted as the precursor to this experiment, and a number of valuable lessons were learned;-

1. The questionnaire must be easy to complete and require few critical judgements.
2. The questionnaire should avoid the need for flicking to and fro through the pages since this distracts the respondent from his task, and increases the time required for completion.
3. The questionnaire should be distributed more widely so as to ensure an adequate response.

4. Teaching staff requested to complete the questionnaire should have some gentle pressure applied to them to ensure an adequate number of returns.

5. A decision was made to utilise students since they were to a certain extent a 'captive audience'. Bottles of wine would not be used again since they appeared to have little effect.

6. No professional designer would be asked to participate since;-

    1. Nil response to previous questionnaire

    2. Time factor - it could take upwards of a month to receive any returns.

6.3.3 Experimental design

The experiment was as follows. A questionnaire consisting of 10 pairs of kitchen layouts, each pair being on a single page and consisting of two alternative layouts for a single room, was assembled and presented to several discrete groups.

1. Students at Scott Sutherland School of Architecture.

2. Teaching staff at the same school

3. Students at the School of Home Economics.

The response, in the form of a dichotomous best/worst ordering for each pair, was to be examined for a consensus and compared to a similar dichotomous classification of each pair based on computer generated values.

6.3.4 Questionnaire design

The raw data for the questionnaire came in the form of kitchen designs from the previous questionnaire. Ten of the designs were chosen at random and served as the basis for each kitchen pair in the second questionnaire.

These designs were then modified, for better or worse, by the author to provide ten alternative layouts, one for each room.

Photocopy reductions were then assembled into a questionnaire with each page consisting of the two alternatives for that room. Positioning of any element of the pair at the top or bottom of the page was done randomly. (See Appendix G, Section G.3.1)

The questionnaire contained explicit instructions on how it was to be completed, together with additional information similar to that provided in questionnaire 1.

The respondents were asked to examine each pair of kitchen layouts and tick whichever of the two they considered made better use of the available space.

The first group of respondents to complete the questionnaire was the teaching staff at the school of architecture. As expected, some gentle 'arm-twisting' was required. One of the returns included in this group was by a visiting lecturer.

The second group of respondents to complete the questionnaire were the students studying Home Economics. This group consisted of two classes of students; those in the second year with no formal training in kitchen design (12 returns out of 30) and those in the third year with some lectures on domestic kitchen design behind them (7 returns out of a class of 20). All the respondents in this group were female.

The third group consisted of architectural students in their second and third years. Due to the timetabling of these students they could not be approached as a class. Instead they were approached individually and in small groups and asked for their cooperation.

Although not originally targeted for questionnaire completion, some post part 2 architecture students (six+ years of architectural training) also consented to complete questionnaires.

Five others who cannot be classified as belonging to any of the above groups but who have experience of using or designing domestic kitchens also completed the questionnaire.

Although this seems a somewhat undisciplined manner in which to elicit information, it successfully ensured an adequate level of response.

6.3.5 Results and evaluation

Tables of results for each group and sub-group can be found in Appendix G, Section G.3.2). Below in figure 6.5 can be seen a summary of those results.

Fig. 6.5 Summary of sample results.

Totals.	Staff (12)	Post Part 2 (9)	Students ARC3 (9)	Students ARC2 (6)	Students HE III (7)	Students HE II (12)	Misc Others (5)		Total Total (60)	Staff + Post (21)	Students ARC (15)	Students HE (19)		Total Total % (60)	Staff & Post % (21)	Students ARC % (15)	STUDENTS HE % (19)	NET TOTAL % (60)	NET STAFF & POST II % (21)	NET ARC % (15)	NET STUDENTS HE % (19)
Kitchen 1a	0	1	2	0	0	2	1		6	1	2	2		10·0	4·8	13·3	10·5	-80·0	-90·4	-73·1	-79·0
Kitchen 1b	12	8	7	6	7	10	4		54	20	13	17		90·0	95·2	86·7	89·5				
Kitchen 2a	12	8	8	6	7	9	5		55	20	14	16		91·7	95·2	93·3	84·2	+83·4	+90·4	+86·6	+66·4
Kitchen 2b	0	1	1	0	0	3	0		5	1	1	3		8·3	4·8	6·7	15·8				
Kitchen 3a	5	4	3	6	0	5	1		24	9	9	5		40·0	42·9	60·0	26·3	-20·0	-14·2	+20·0	-47·4
Kitchen 3b	7	5	6	0	7	7	4		36	12	6	14		60·0	57·1	40·0	73·7				
Kitchen 4a	3	4	7	6	5	7	3		35	7	13	12		58·3	33·3	86·7	63·2	+16·6	-33·4	+73·4	+26·4
Kitchen 4b	9	5	2	0	2	5	2		25	14	2	7		41·7	66·7	13·3	36·8				
Kitchen 5a	8	5	6	5	2	4	3		33	13	11	6		55·0	61·9	73·3	31·6	+10·0	+23·8	+46·6	-36·4
Kitchen 5b	4	4	3	1	5	8	2		27	8	4	13		45·0	38·1	26·7	68·4				
Kitchen 6a	6	3	3	4	4	5	3		28	9	7	9		47·5	45·0	46·7	47·4	-5·0	-10·0	-6·6	-5·2
Kitchen 6b	5	6	6	2	3	7	2		31	11	8	10		52·5	55·0	53·3	52·6				
Kitchen 7a	8	3	3	4	3	4	1		26	11	7	7		43·3	52·4	46·7	36·8	-13·4	+4·8	-6·6	-26·4
Kitchen 7b	4	6	6	2	4	8	4		34	10	8	12		56·7	47·6	53·3	63·2				
Kitchen 8a	7	6	5	2	5	8	3		36	13	7	13		60·0	61·9	46·7	68·4	+20·0	+23·8	-6·6	+36·8
Kitchen 8b	5	3	4	4	2	4	2		24	8	8	6		40·0	38·1	53·3	31·6				
Kitchen 9a	7	5	1	0	0	3	2		18	12	1	3		30·0	57·1	6·7	15·8	-40·0	+14·2	-86·6	-68·4
Kitchen 9b	5	4	8	6	7	9	3		42	9	14	16		70·0	42·9	93·3	84·2				
Kitchen 10a	3	5	1	4	1	3	1		18	8	5	4		30·0	38·1	33·3	21·1	-40·0	-23·8	-33·4	-57·8
Kitchen 10b	9	4	8	2	6	9	4		42	13	10	15		70·0	61·9	66·7	78·9				

Perhaps a better illustration is the following histogram (figure 6.6). A and B results in each sample group have been converted to a percentage (allowing comparison between samples) and then the net difference in the percentage value, where A results are positive and B results are negative, has been plotted for each of the ten cases.

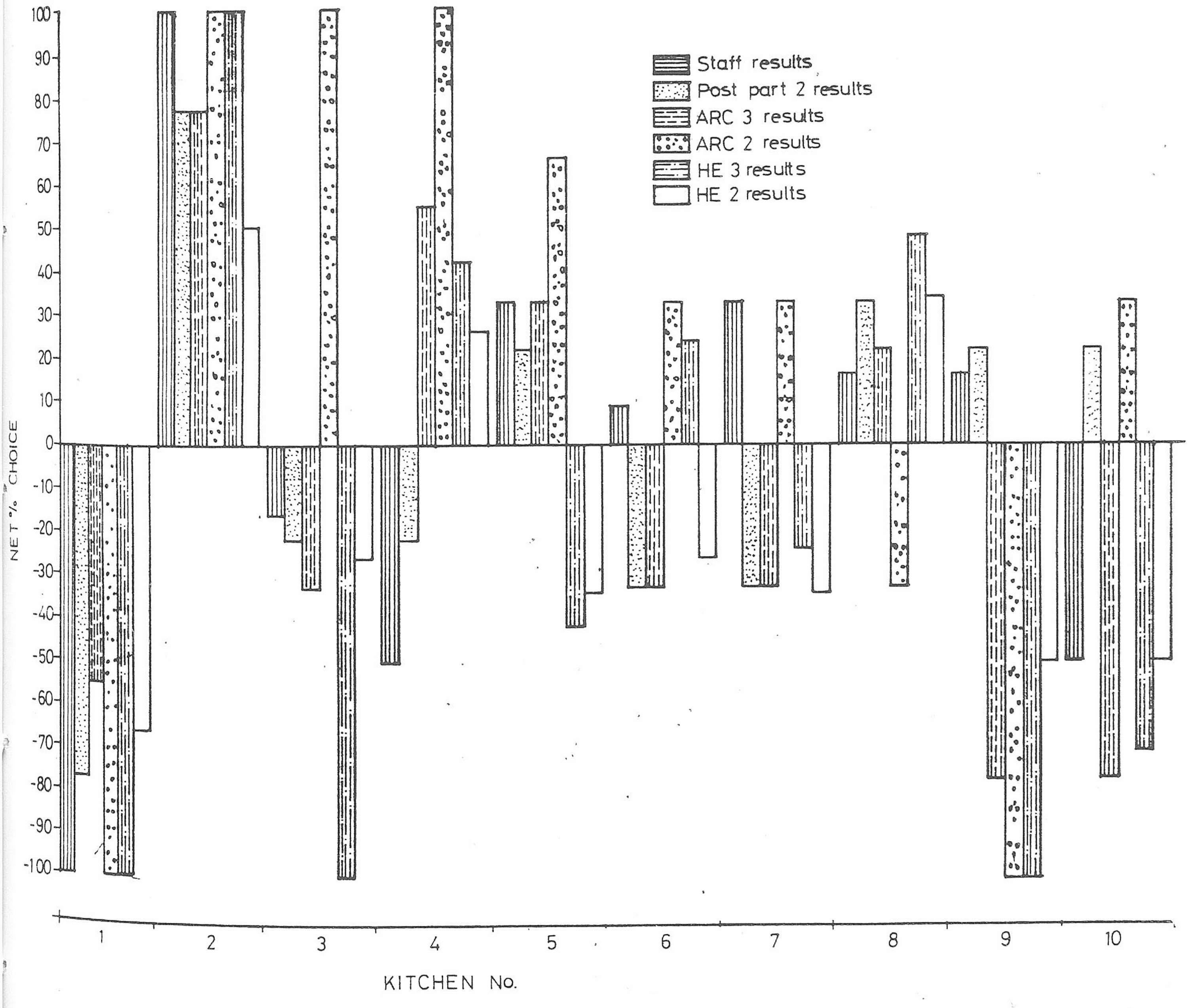


Fig. 6.6 Histogram of net % choice of samples

The results of this experiment are somewhat harder to analyse, and perhaps the best method of examining the results would be to take each kitchen pair in turn.

In kitchen 1, all groups (indeed almost all respondants) clearly favoured Layout B, by at least 3 to 1.

Kitchen 2 if anything provided a more emphatic picture with all groups clearly favouring Layout A, again by at least 3 to 1

It is with kitchen 3 that the situation becomes more confused. The ARC 2 sample and the HE 3 sample are both unanimous in their decision - the trouble is they disagree with each other! The remaining samples marginally favour Layout B in each case by less than 2 to 1 Both the ARC 2 and the HE 3 samples are small so it is probably safe to assume that their results can be ascribed to statistical variation.

Kitchen 4 has similar results to kitchen 3. Only the ARC 2 students are unanimous in selecting Layout A. They are supported in this selection by the ARC 3 students (3+ to 1) and the HE 3 students (2.5 to 1) and opposed by the staff (3 to 1). The other groups show no clear consensus. The unanimity of the ARC 2 students may again be due to statistical variation. On average Layout A is slightly favoured but no overwhelming consensus emerges.

Kitchens 5 to 8 repeat the above pattern, slightly favouring A or B but with no clear decision.

The situation with Kitchen 9 resembles that for kitchens 1 and 2. Three groups show unanimous or near unanimous support for Layout B (ARC 3, ARC 2, HE 3) with HE 2 students supporting their selection 3 to 1. The staff and post graduate students reach no decisive decision but marginally support Layout A. This could be interpreted as an overall consensus on Layout B.

The situation is broadly similar for kitchen 10.

To summarise these thoughts:-

Kitchen 1	Definite consensus	B
Kitchen 2	Definite consensus	A
Kitchen 3	No consensus	
Kitchen 4	No consensus	
Kitchen 5	No consensus	
Kitchen 6	No consensus	
Kitchen 7	No consensus	
Kitchen 8	No consensus	
Kitchen 9	Consensus	B
Kitchen 10	Consensus	B

This is shown diagramatically below in figure 6.7.

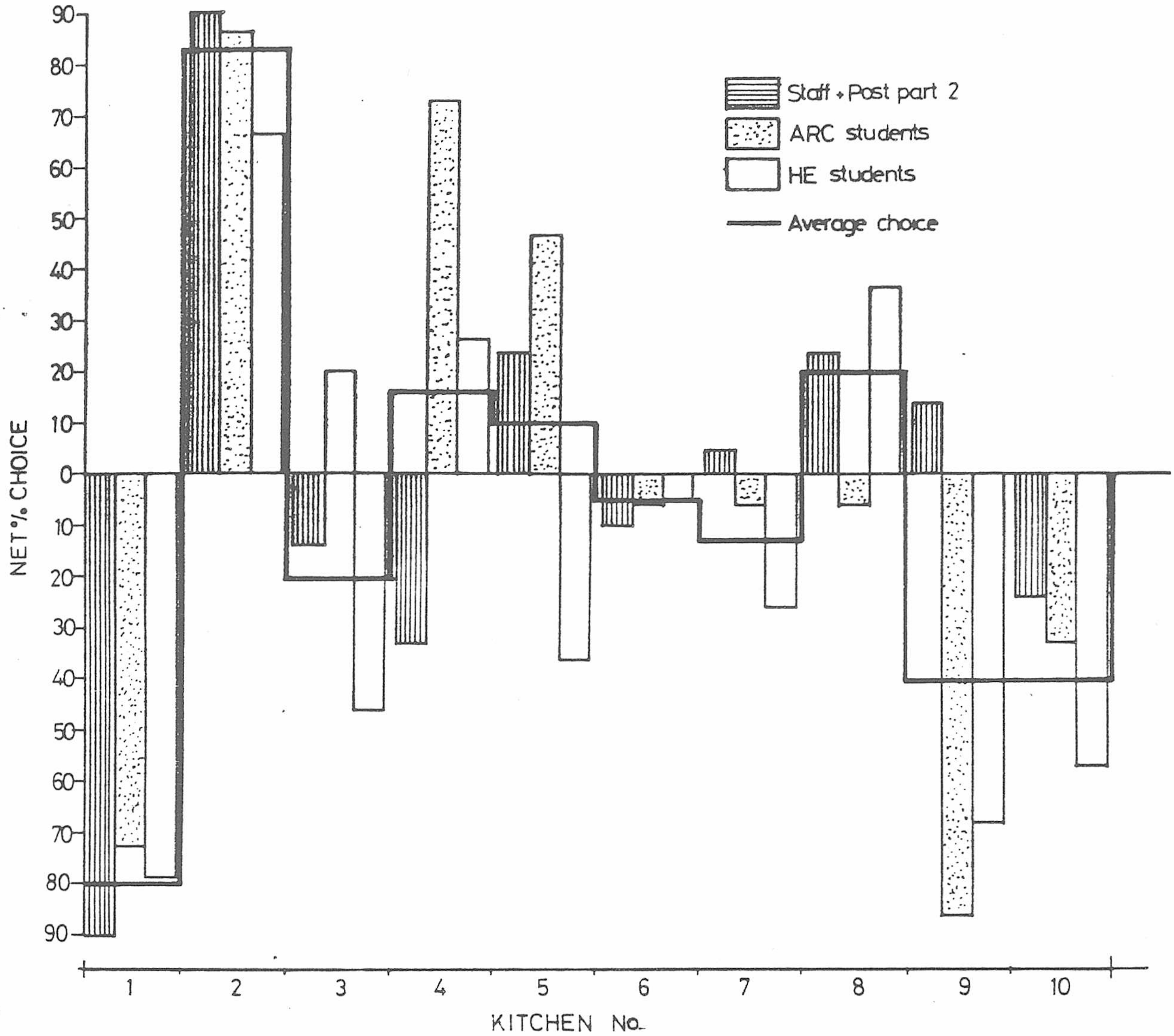


Fig. 6.7 Summary histogram of net % choice of samples

The next stage of analysis was to determine why four out of the ten kitchen pairs demonstrated an obvious consensus whilst the remaining six did not.

Bearing in mind the 'result' of experiment 1 it was thought that a relationship between consensus and complexity of comparison might be found. To this end, a complexity index was assembled for each of the ten kitchen pairs based on the minimum number of fixture transformations that had to be performed to convert one layout to the other. See figure 6.8 below.

Kitchen	Index	Kitchen	Index
1	2	6	4
2	7	7	7
3	9	8	6
4	8	9	3
5	13	10	12

Fig. 6.8 Table of complexity index for ten kitchens

Assessing complexity in this manner has some problems, for instance when different types of fixtures are required as part of the transformation process or when different numbers of fixtures are required.

Nevertheless, it is clear that consensus is not related to a crude measure of the number of transformations required.

As a result of this finding it was clear that a more detailed examination of each kitchen pair with regard to the sort of decision that was required was in order. The decisions presented below are largely based on the comments written on the questionnaire returns and the comments made to the author by respondents when they were completing the questionnaire.

Kitchen1 Did moving the fridge away from the cooker make B a better layout?

Kitchen2 Is it better to have the fridge on the same wall as the cooker or is it better to have the washing machine adjacent to the sink and plenty of worktop around the cooker as in A?

Kitchen3 Is it better to have the sink further from the door and the fridge closer to the cooker as in B, or is it better to have the effective use of all worktop as in A?

Kitchen4 Is it better to have more worktop between the cooker and the sink or between the fridge freezer and the cooker?

Kitchen5 Is it better to have the washing machine adjacent to the sink and the cooker or is it better to have plenty of space between the cooker,sink and fridge?

<u>Kitchen6</u> Is it better to have cooker, sink, oven and fridge located close to each other with little intervening worktop or is it better to have them further apart? How is this affected by closeness of washing machine to sink as in A and the island position of the hob as in B?

<u>Kitchen7</u> Is it better to have the fridge close to the sink or far away from the oven?

<u>Kitchen8</u> Is it more important for the cooker to be close to the fridge or far away from the door and through circulation?

<u>Kitchen9</u> Which is worse, to have a tall oven unit directly adjacent to the cooker or to the fridge/freezer?

<u>Kitchen10</u> Is it better to have oven and fridge adjacent but isolated from sink and hob or to have the fridge closer to the hob/sink position?

The above comments describe the level of the complexity of the decisions that needed to be made. In kitchens 1 and 2 it is clear that one layout was superior to the other and required no balancing of conflicting requirements.

Kitchen 10 is somewhat similar though not quite so clearly defined.

The majority of respondents found kitchen 9 a clearly defined case, but it is possible that many of them did not realise that the oven housing unit was a tall unit and could well obstruct the use of the cooker. If they had made this mistake then the decision would be comparatively easy.

Those kitchens in which no consensus could be demonstrated show one of three properties:-

1. Marginal decision required ie. when there is little to choose between the layouts as in kitchen 4.

2. Very complex decisions, such as kitchen 6.

3. Resolving conflicting requirements such as kitchen 7.

Having to a certain extent explained the results of the second questionnaire with regard to their internal consistency; let us examine how these results relate to the computer evaluation. See figure 6.9 below.

These figures can be represented in a manner similar to that used for the questionnaire results. See figure 6.10.

Numeric & Ranking Results (2nd Quest)	SP penalty	VU penalty	AP penalty	OP penalty	ER penalty			AP ranking	OP ranking	ER ranking		3 * AP penalty	ER = SP + VU + 3AP + OP		NEW E.R. RANKING
Kitchen 1a	1·004	1·135	0·101	1·548	0·947							1·303	1·998		
Kitchen 1b	1·004	1·135	0·051	1·351	0·885			b	b	b		0·153	0·911		b
Kitchen 2a	1·016	0·858	0·000	2·527	1·100			a				0·000	1·100		a
Kitchen 2b	1·016	0·858	0·055	2·328	1·064				b	b		0·165	1·142		
Kitchen 3a	1·001	1·240	0·023	1·900	1·041			a	a	a		0·069	1·052		a
Kitchen 3b	1·001	1·240	0·033	1·976	1·063							0·099	1·079		
Kitchen 4a	1·004	1·342	0·000	1·328	0·919			a	a	a		0·000	0·919		a
Kitchen 4b	1·004	1·342	0·006	1·464	0·954							0·018	0·957		
Kitchen 5a	1·000	1·182	0·107	1·757	1·011				a	a		0·321	1·065		a
Kitchen 5b	1·000	1·182	0·081	1·837	1·025			b				0·243	1·066		
Kitchen 6a	1·019	1·120	0·471	2·843	1·363							1·413	1·599		
Kitchen 6b	1·019	1·120	0·103	2·790	1·258			b	b	b		0·309	1·310		b
Kitchen 7a	1·015	1·079	0·203	2·103	1·100							0·609	1·202		
Kitchen 7b	1·015	1·071	0·091	1·893	1·018			b	b	b		0·273	1·065		b
Kitchen 8a	1·038	0·945	0·098	2·419	1·125			a	a	a		0·294	1·174		a
Kitchen 8b	1·038	0·984	0·192	3·013	1·307							0·576	1·403		
Kitchen 9a	1·002	1·251	0·128	2·001	1·095				a	a		0·384	1·160		b
Kitchen 9b	1·002	1·251	0·101	2·072	1·107			b				0·303	1·157		
Kitchen 10a	1·033	1·213	0·551	1·836	1·158							1·653	1·434		b
Kitchen 10b	1·033	1·213	0·376	1·307	0·981			b	b	b		1·128	1·170		

Fig. 6.9 Table of computer evaluations of ten kitchens

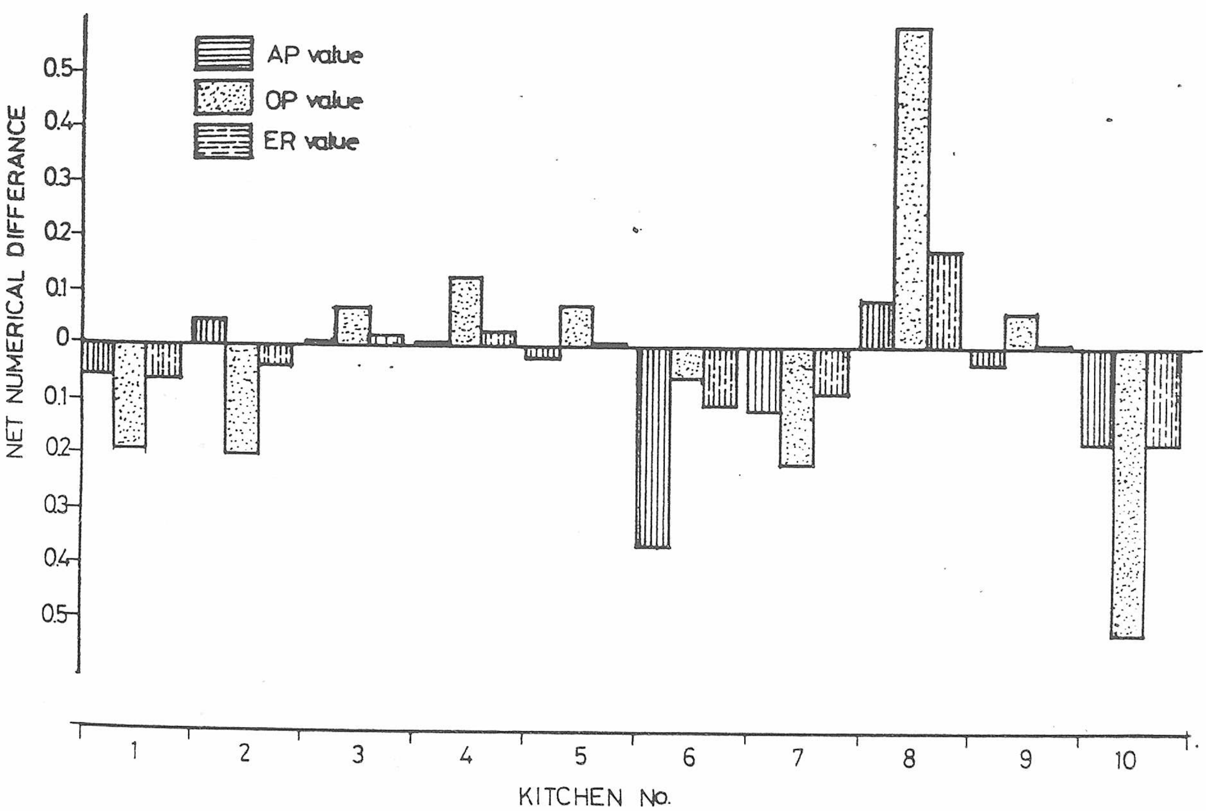


Fig. 6.10 Histogram of net numerical differance in computer values for ten kitchens

As a first step in the correlation of the computer selection to the sample selection, see figure 6.11 below.

Kitchen no.	1	2	3	4	5	6	7	8	9	10
Sample choice	B	A	B	A	A	B	B	A	B	B
AP choice	B	A	A	A	B	B	B	A	B	B
OP choice	B	B	A	A	A	B	B	A	A	B
ER choice	B	B	A	A	A	B	B	A	A	B

Fig. 6.11 Summary of sample and computer choices for ten kitchens

In figure 6.11 above, underlining of a choice indicates a marginal (very marginal if underlined twice) decision in favour of that layout.

In the case of the sample, marginal is defined as a net % difference of 20% or less (ie. a split of 3:2 or less). A very marginal decision involves a net difference of 10% or less (11:9).

In the case of AP value, marginal is defined as a numeric difference of less than 0.030. Very marginal is 0.010 or less.

In the case of the OP value, marginal is defined as a numeric difference of 0.080 or less.

In the case of the ER value, marginal is defined as 0.040 or less with very marginal being 0.015 or less.

Also in the diagram, boxed selections indicate the computer based selections which diverge from the average selection of the sample.

One thing to notice from the table is that the OP selection and the ER selection are identical. This demonstrates the dominance exerted by the former over the latter. This is evidenced by the table of computer values (figure 6.9) where OP values are clearly the largest constituent of the ER value.

Examining the ER selection in detail, we see that it differed from the sample selection in three cases. In layout nine it very marginally selected A over B. In layout two it marginally selected B over A, whilst in layout three it marginally selected layout A over B.

Examining the AP selection in detail, we see that it differed from the sample in two cases. In layout five it marginally selected B over A. It should be noted that the sample choice of layout A was very marginal. AP also marginally selected A over B in layout three.

Considering that the OP value dominates the ER value, and the AP selection matches the sample selection closely, a modified ER value was calculated by multiplying the AP value by three before adding it to the other constituent penalties. A new ER ranking was assembled (see right of figure 6.9).

A comparison of this new ER selection with the sample selection is presented below in figure 6.12.

Kitchen no.	1	2	3	4	5	6	7	8	9	10
Sample	B	A	B	A	A	B	B	A	B	B
Modified ER	B	A	A	A	A	B	B	A	B	B

Fig. 6.12 Comparison of modified ER based choice to sample choice

In the diagram above marginal is defined as a numeric difference of 0.030 or less, very marginal being 0.005 or less.

As can be seen, the new ER value gives a very good match with the average sample selection.

Considering kitchen three, the reason for the discrepancy is evident. In layout A the sink is very close to the door. This factor influenced many respondents to choose layout B. Sink/door closeness should have attracted an association penalty but as can be seen in figure 2.9 (see Chapter 2, Section 2.3.2) it does not. Calculation of the modified ER value including an association penalty for sink/door closeness causes a change in the ER based selection to layout B.

However consideration of the modified ER results should not stop there. The ER based selection marginally chooses layout B in kitchen nine. The sample shows no such hesitancy, It is thought that this divergence can be ascribed to some of the respondents failing to notice that in both layouts the oven has a tall housing unit.

The one other interesting comparison to be drawn from figure 6.12 is that the ER based selection in kitchen 6 decides quite definitely for layout B. The sample chooses layout B by a small margin. There is nothing as obvious as a 'missing' association penalty to explain this discrepancy. However, several respondents stated that both layouts in kitchen 6 were bad (indeed one member of the staff refused to make a choice for that reason and it may be that people find it difficult to make a design choice when niether alternative has much to recommend it.

It should be noted that the modifying factor of three as applied to the AP value was selected to allow the ER value to more closely match the sample selection. Another experiment is required to test whether the value of three is applicable to a completely different kitchen sample (which would allow the tentative conclusion that the value of three is an appropriate multiplying factor for domestic kitchens).

Should it be possible to draw this conclusion, further experiments would still be required to establish whether this modifying value is appropriate to non-kitchen domestic activity spaces - or indeed to non-domestic activity spaces.

This is probably an appropriate point to comment on the instability of the ER measure. Each of the four constituent penalty factors measure different aspects of room efficiency, and the device of a (weighted) average of these values to derive the ER value has little logical basis since each measure has disparate units, a case of "apples and oranges". Even so, on occassion an overall figure, however unstable, may be useful.

However, when using the developed software to actually modify designs I found the ER value of minimal importance prefering to balance each of the other four numerical values against the intuitive feeling for the quality of the designed space.

6.3.6 Conclusions

This experiment was considerably more successful than the first.

It showed that in four out of ten cases respondents showed a clear consensus in their choice of layout.

It showed that the degree of consensus could be seen as being dependant on:-

1. Problem complexity.
2. Conflict resolution.
3. Marginality of decision.

It showed that the original computer evaluation approximately matched the respondents choice over all ten layouts.

It demonstrated that a choice based on a modified ER value matched the respondents choice more closely and that the inclusion of a 'missing' association penalty resulted in an exact match.

However, there was no evidence of a linear correlation between degree of sample consensus in their choice and degree of change in the computer evaluation.

6.4.0 EXPERIMENT 3

This section provides details of the third validation exercise.

6.4.1 Aim of the experiment

The aim of this experiment was quite different from that of the previous two experiments.

In this experiment the aim was twofold:-

1. To establish some of the criteria used by the sample in the previous questionnaire when making their choice.

2. To establish, if possible, the degree of importance attached to individual element associations by the sample.

6.4.2 Experimental precursor

No formal precursor was used in this experiment.

However some of the respondents to the second questionnaire wrote comments on the edge of each page which to some extent explained their thinking when making a design choice.

These comments inspired this experiment and have been used as an additional source of relevant information.

6.4.3 Experimental design

Bearing in mind that the purpose of this research was to investigate a general activity space model and not to conduct an in-depth investigation into the design of domestic kitchens, this experiment was small scale and low key.

The experiment was to be based around an open interview of between four and six designers and kitchen users. The purpose of this interview was to externalise some of their design thinking.

The findings from this interview would be combined with the comments found on the second questionnaire in an attempt to meet the aims of the experiment as defined in the previous experiment.

6 4.4 Questionnaire design

Although the experiment was to consist of an open interview, a memo sheet of open questions was assembled to aid the author in directing the interview. See Appendix G, Section G.4.1 for examples. Each interview was carried out at an individual session lasting between one half and one hour.

6.4.5 Results and evaluation

Below are itemised some of the criteria by which kitchen designs were judged. There were two main classes of criteria; positive and negative. The classifications of positive and negative are to a certain extent arbitrary since any negative 'rule' can be expressed in a positive manner. However, I believe, this classification gives an insight into the manner in which the respondents judged each layout.

The two sets of criteria are presented below in a rough ranked order of importance (determined largely by the total number of times any particular 'rule' was referred to by the respondents).

1. Negative criteria

   1. Cooker or oven unit should not be adjacent to fridge or fridge/freezer.

   2. Cooker should not be adjacent to a door or window.

   3. Sink should not be adjacent to door.

   4. Cooker should not be adjacent to sink, wall or tall unit.

   5. No overlap of functional area between cooker and washing machine.

   6. High level units should not end close to a door or window.

   7. Tall units should not be placed near low corners.

   8. Rooms should not be too small nor too narrow.

2. Positive criteria

   1. Good work triangle between cooker, sink and foodstore is desirable.

   2. Washing machine should be close to sink.

   3. Cooker should have 300+ mm of worktop to each side.

   4. Cooker or oven to sink should be continuous worktop.

   5. Cooker should be close to oven.

6. Room doors and fitment doors should be sited carefully to avoid spatial clashing.

7. There should be a 900+ mm requirement between worktop faces in a 'galley' kitchen.

8. Cooker should have unobstructed access to external door in case of pan fires.

6.4 6 Conclusions

The most obvious conclusion to be drawn from this experiment is that, in the main, respondents perceived layouts in terms of functional relationships between different furniture element clusters.

The concept of 'user areas' tended to be non-explicit. However a somewhat similar concept was used by the respondents as can be seen in the positive criteria list where elements 3,6 and 7 obviously require a spatial judgement.

Room shape and size were of less importance.

To conclude, the respondents choice of evaluative criteria matched those published advisory notices upon which much of the input data required by the model was based. An exception to this was that many respondents felt that the sink should not be located adjacent to a door. The design guides consulted did not make this suggestion.

6.5.0 GENERAL CONCLUSION

The general conclusion to be drawn from this series of experiments is that within the limits of time, funding and resources the activity space model devised by the author is a valid representation of reality and constitutes a valuable appraisal aid to the designer.

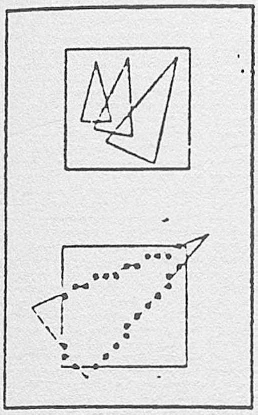
6.6.0 REFERENCES

(6.1) British Telecom, Yellow Pages Aberdeen 1984, British Telecommunications plc., London, 1984.

(6.2) Statistical calculation courtesy of program 'Minitab', by Statistics Dept., Penn. State University, USA.

(6.3) Cakin,S., 'Decision making in Design: Evaluation of Holiday Houses', DMG/DRS Journal, Vol. 9, No. 2, Berkeley, California, 1975.

(6.4) Lawson,B., How Designers Think, Architectural Press, London, 1980, p.61



CHAPTER 7

A MINI USER MANUAL

CONTENTS PAGE

7.1.0 INTRODUCTION

This chapter is an abbreviated user manual to the KAPABLE modelling system. (Reference is made to a more substantial manual in Appendix I.1.2). It is intended to give the flavour of the operation, rather than an in-depth study of the actual mechanics of the software.

7.2.0 SYSTEM OVERVIEW

This section gives an overview to the total system configuration.

7.2.1 Hardware configuration

KAPABLE runs on a Tektronix 4054 (A update), Option 31 (Refresh Graphics) Desktop micro-computer using Tektronix Graphics System BASIC, in conjunction with a Tektronix 4907 file manager with in-built floppy disk drive. Hardcopy is through a Tektronix 4631 hard copy unit. This is shown diagramatically in figure 7.1 below.

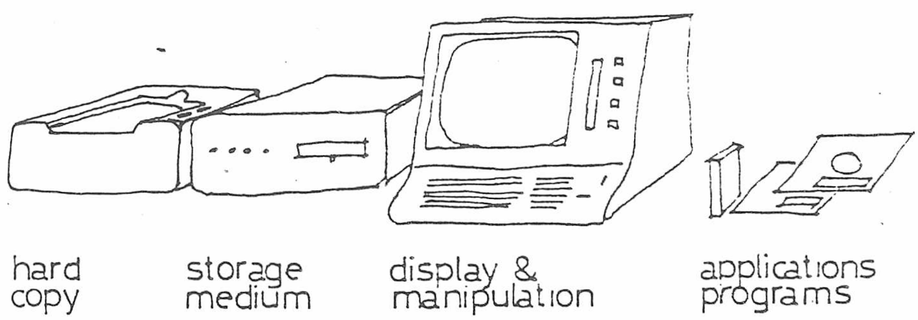


Fig 7.1 Hardware Configuration

7.2.2 Software configuration

Program and data files are held on a double density 8" floppy disk (630k) residing in the file manager. In addition, the magnetic tape in the tape-slot of the 4054 has some system functions. See figure 7.2 below.

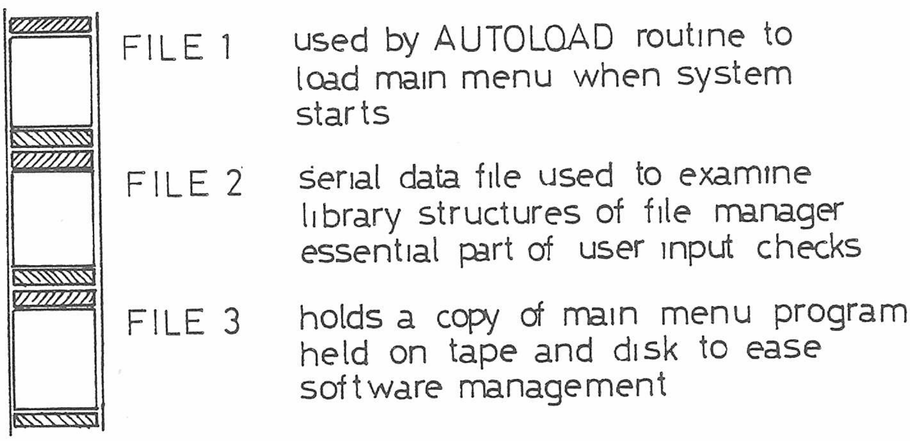


Fig 7.2 Tape Structure

The disk within the disk drive performs a dual function. The first function is to hold system information - essentially program segments. This is shown diagramatically in figure 7.3 below; the diagram mimics the program structure rather than the library structure of the disk.

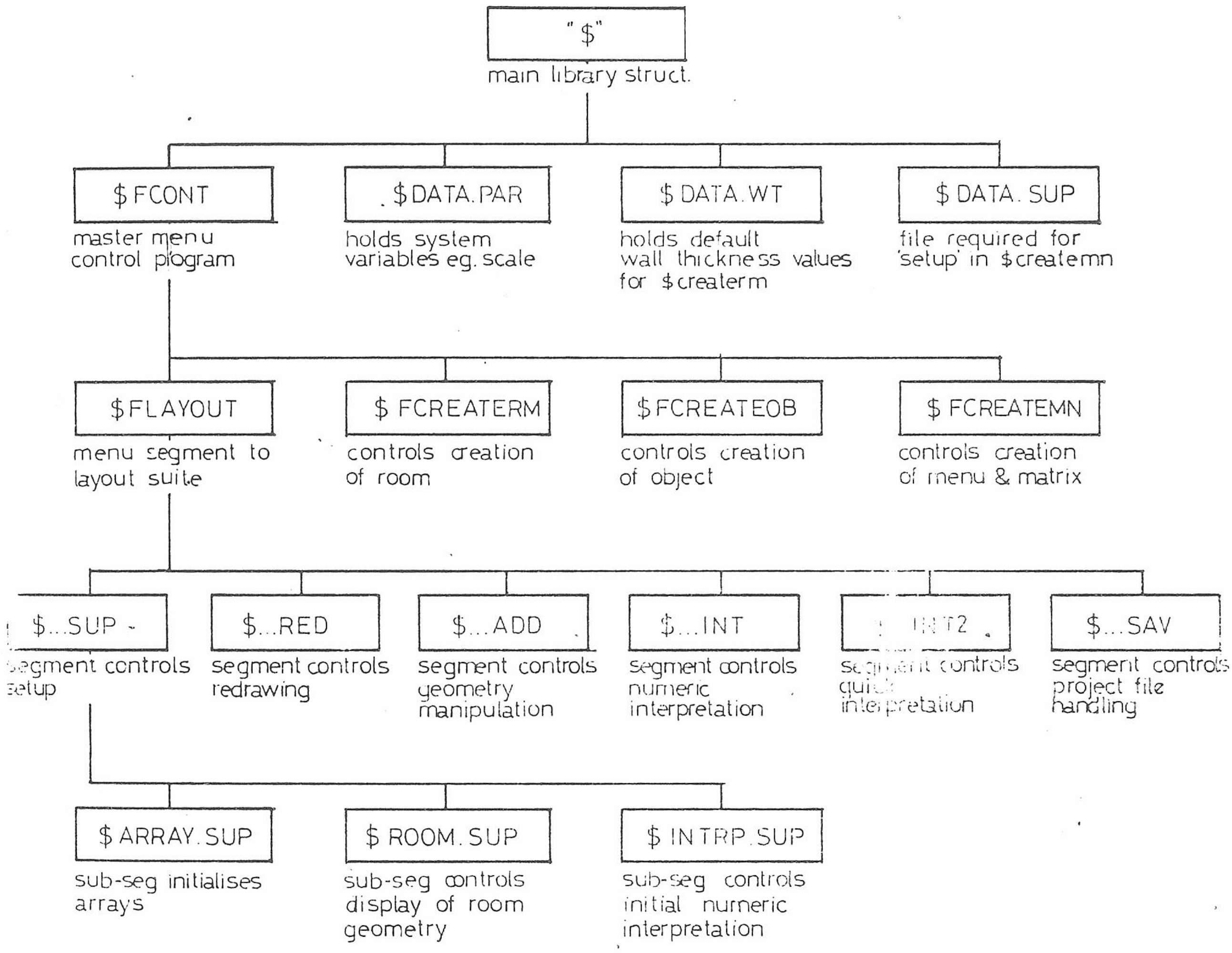
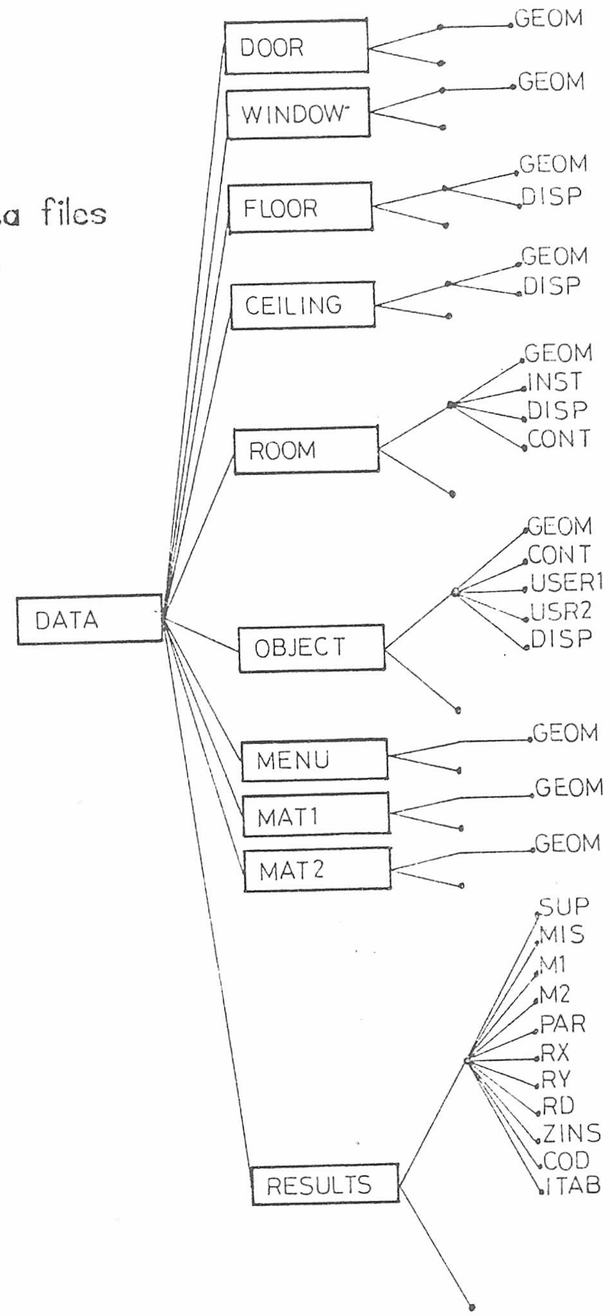


Fig 7.3 Program structure on disk

The second function of the disk is to hold and manipulate user created data. The disk library structure for this data is shown below in figure 7.4

Fig 7.4 User data files on disk



Since the following sections describe the LAYOUT suite of programs, some mention of the file types indicated in figure 7.4 is in order.

The only files LAYOUT is capable of creating are RESULTS files - holding data describing the current position of a particular job. The file extensions for each RESULTS file structure have the following meaning:-

.SUP - file holds set-up information
.MIS - file holds miscellaneous variable
.M1 - file holds copy of MAT1
.M2 - file holds copy of MAT2
.PAR - file holds system parameters
.RX - file holds x-coordinates of room
.RY - file holds y-coordinates of room
.RD - file holds descriptions of room object locations
.ZINS- file describes locations of furniture objects
.COD - file holds references to ITAB
.ITAB- file holds geometry descriptions of all room and furniture objects

7.3.0 RUNNING KAPABLE

KAPABLE as a system is very easy to use. The following section describes how to start-up the system and how to use the main menu.

7.3.1 Switching on

Switch on power switches to the file manager followed by the 4054 micro-computer. The switch for the file manager is located on the front plate of the 4907 while the switch for the 4054 is located below the right hand side of the keyboard.

Put the KAPABLE (Main Copy) tape into the tape slot and press

AUTOLOAD key.

The system automatically starts up and requests that the user press

ANY key

The system then requests that the user enter the date and time in the same format as the example given. The RETURN KEY (CR) must be used to terminate input. If the user should make an error at this stage the easiest thing to do is to restart by pressing the AUTOLOAD key again.

The system then prompts the user to load a KAPABLE disk (a disk that has been formated and had certain default information such as wall thickness tables and system parameters together with a copy of program files saved onto it). Do this and then press

ANY key

The system mounts the disk. The red indicator light on the file manager marked 'clock' should now go out. At this point the main menu appears.

7.3.2 The main menu

Fig.7.5 below shows the main menu. All menus in KAPABLE are driven by the 'user definable keys' located at the top left hand corner of the 4054 keyboard or by the appropriate keyboard letter indicated on the menu.

```
(K)          (L)          (M)           (N)          (O)
(A) Format   (B) KAPABLE  (C)FILE/COPY
(P)          (Q)          (R)           (S)          (T)
(F) Cr Room  (G) Cr Obj   (H) Cr MENU   (I) Layout   (J) Exit

USE USER DEFINABLE KEYS TO CONTINUE
```

Fig 7.5 Main menu

UDK	Letter	Function name	Function
1	A	FORMAT	formats disk
2	B	KAPABLE	adds KAPABLE information
3	C	COPY	copies user data files between disks
6	F	CREATRM	creates room geometry
7	G	CREATEOB	creates object geometry
8	H	CREATMN	creates menus and matrices
9	I	LAYOUT	manipulates and interprets particular geometries
10	J	EXIT	dismounts disk and ends

To use LAYOUT successfully the user must create a SETUP in CREATMN; this requires prior creation of a room and some furniture objects.

However,since this is a much abbreviated user manual, the only element of the main menu to be considered in any detail is the segment called LAYOUT. This segment is accessed by pressing UDK 9 or the letter I. The system then automatically loads the correct software segment.

7.4.0 LAYOUT SUITE

After the user has pressed the appropriate key to enter LAYOUT, the system automatically loads the correct program segment ($FLAYOUT), and begins the initialisation process.

The following messages appear at the top left hand corner of the screen:-

1. LAYOUT
   INITIALISATION

2. If the user has not previously created a room layout and menus and no SETUP file has been created ($DATA.SUP) then:-

   LAYOUT
   NO SET-UP CONFIGURATION

   appears and a long bell sounds. The main menu segment is then reloaded into the 4054 memory.

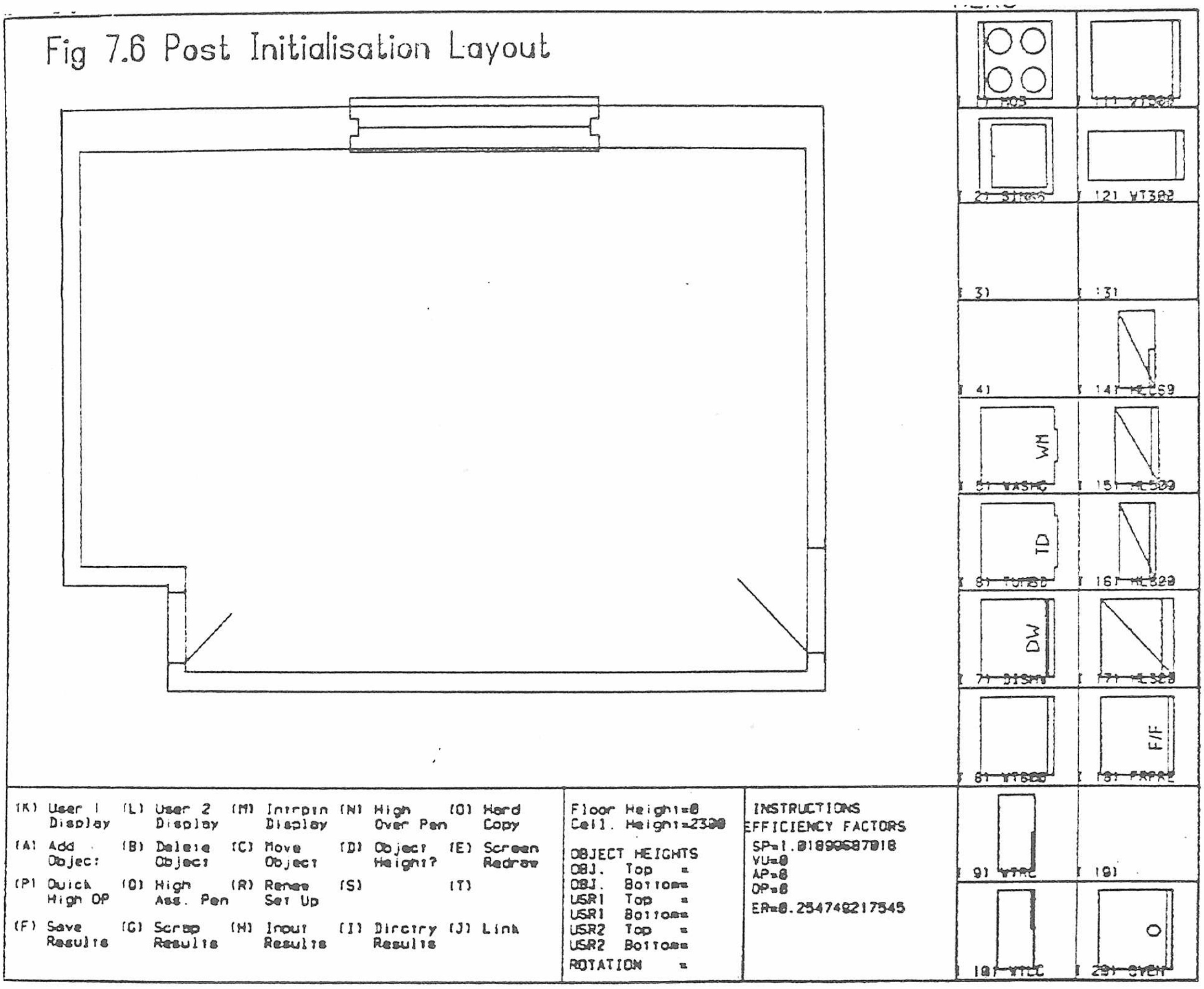
3. Normally, the following appears:-

   LAYOUT
   SETTING UP ARRAYS

4. LAYOUT
   SETTING UP ROOM GEOMETRY

5. LAYOUT
   SETTING UP INITIAL INTERPRETATION VALUES

This process lasts some five minutes or so and requires no user interaction. The bulk of this time is taken by the requirement to execute the last stage (interpretation). The screen then pages and the LAYOUT menu and the SETUP geometry appear in the display section of the LAYOUT screen. Typically as in figure 7.6 below.



From this point the user drives LAYOUT by menu commands in much the same manner as described earlier in section 7.3.2.

7.4.1 File handling

This section describes the operation of those keys outlined in figure 7.7 below.

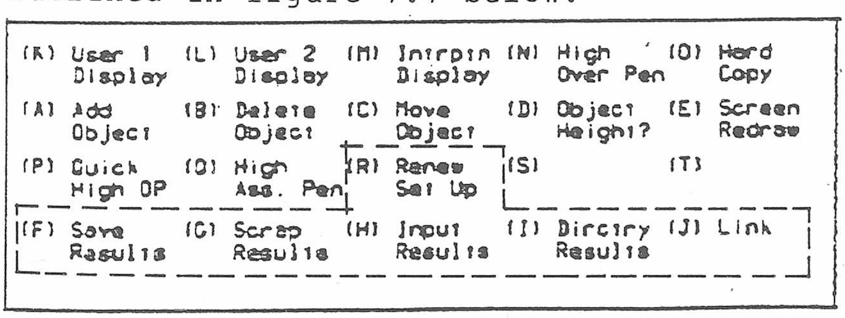


Fig 7.7 File Handling Keys

(R) RENEW SET UP

$FLAYOUT.SAV program segment is loaded into memory unless it is already there, and a warning is printed in refresh in the instructions box of the LAYOUT screen, together with a short prompt bell.

```
WARNING
CURRENT PROJECT WILL BE
SCRAPPED

CR=CONFIRM OPERATION
 0=RETURN TO MENU
```

The user must then press CR or zero. The system waits until the user does so; before removing the warning. Zero returns the user to the LAYOUT menu. CR causes the system to repeat the initialisation process described in section 7.4.0.

This key is intended to allow the user a means of quickly recovering a basic room shape and menu so that he can try out a variety of layouts.

(F) SAVE RESULTS

$FLAYOUT.SAV program segment is loaded into memory unless it is already there, and a warning is printed in refresh in the instructions box of the LAYOUT screen, together with a short prompt bell.

```
SAVE PROJECT
ENTER NAME OF FILE

????????
0=RETURN TO MENU
```

If the user presses zero he is returned to the LAYOUT MENU. Otherwise, the user is required to enter a name for his project file. As the user enters the file name the prompting question marks are replaced by the letters of the file name. Input ends after a CR or after eight characters, whichever occurs first.

The system then checks to ensure that the file name does not already exist. If it does,

NAME UNSUITABLE
TRY AGAIN

appears together with a long error bell. The prompt for a file name reappears.

Once the user has entered a valid file name a DATA/RESULTS file is created on disk which contains all the required information concerning the project. The user is then returned to the LAYOUT menu.

(G) SCRAP RESULTS

$FLAYOUT.SAV program segment is loaded into memory, unless it is already there. The message

SCRAP PROJECT

appears in the instruction box together with a prompt for a file name as describes earlier at (F) SAVE RESULTS.

As before, zero returns the user to the LAYOUT menu. Entering a file name causes the warning message below to appear:-

SCRAP PROJECT
WARNING

(file name) WILL BE
SCRAPPED

CR=CONFIRM OPERATION
0=RETURN TO MENU

When CR is pressed the following message appears briefly,

(file name) HAS BEEN
SCRAPPED

before the user is returned to the LAYOUT menu. This key allows the user to clear some disk space.

(H) INPUT RESULTS

$FLAYOUT.SAV program segment is loaded into memory, unless it is already there. In the instruction box the following message appears together with a short bell prompt.

RECALL PROJECT

ENTER NAME OF FILE
????????

0=RETURN TO MENU

On entering a file name, a warning is issued similar to that described for (R) RENEW SET UP.

When a CR is pressed by the user, the system loads all the data from the appropriate RESULTS file into memory and executes a REDRAW to display the new job.

(I) DIRECTORY RESULTS

$FLAYOUT.SAV program segment is loaded into memory, unless it is already there. In the instruction box the following message appears together with a short bell prompt.

PROJECT DIRECTORY

CR=NEXT ITEM
0=RETURN TO MENU

(file name 1)

Pressing CR gives the next file name in the directory. That is, file name 1 is replaced by file name 2. This process continues until all RESULTS files have been displayed, whereupon the system prints as a file name NONE and the user is returned to the LAYOUT menu.

(J) LINK

$FLAYOUT.SAV program segment is loaded into memory, unless it is already there. In the instruction box the following message appears together with a short bell prompt.

WARNING
CURRENT PROJECT WILL BE
SCRAPPED

CR=CONFIRM OPERATION
0=RETURN TO MENU

Pressing CR causes the system to exit from LAYOUT and reload the main menu segment ($FCONT) and display the main menu.

7.4.2 Graphical display keys

This section describes the operation of those keys outlined below in figure 7.8. The system ensures that $FLAYOUT.RED program segment is present when these keys are used since that segment incorporates the major part of the redrawing mechanism.

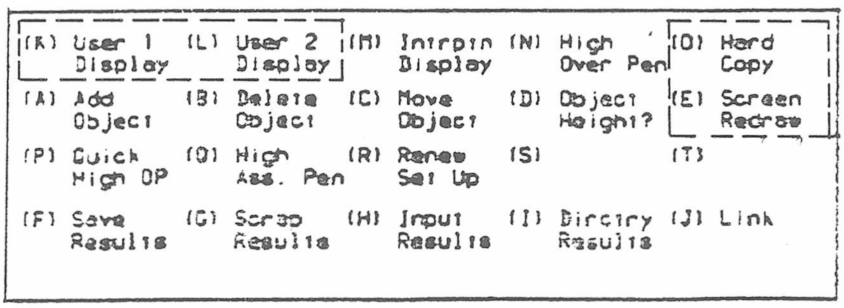
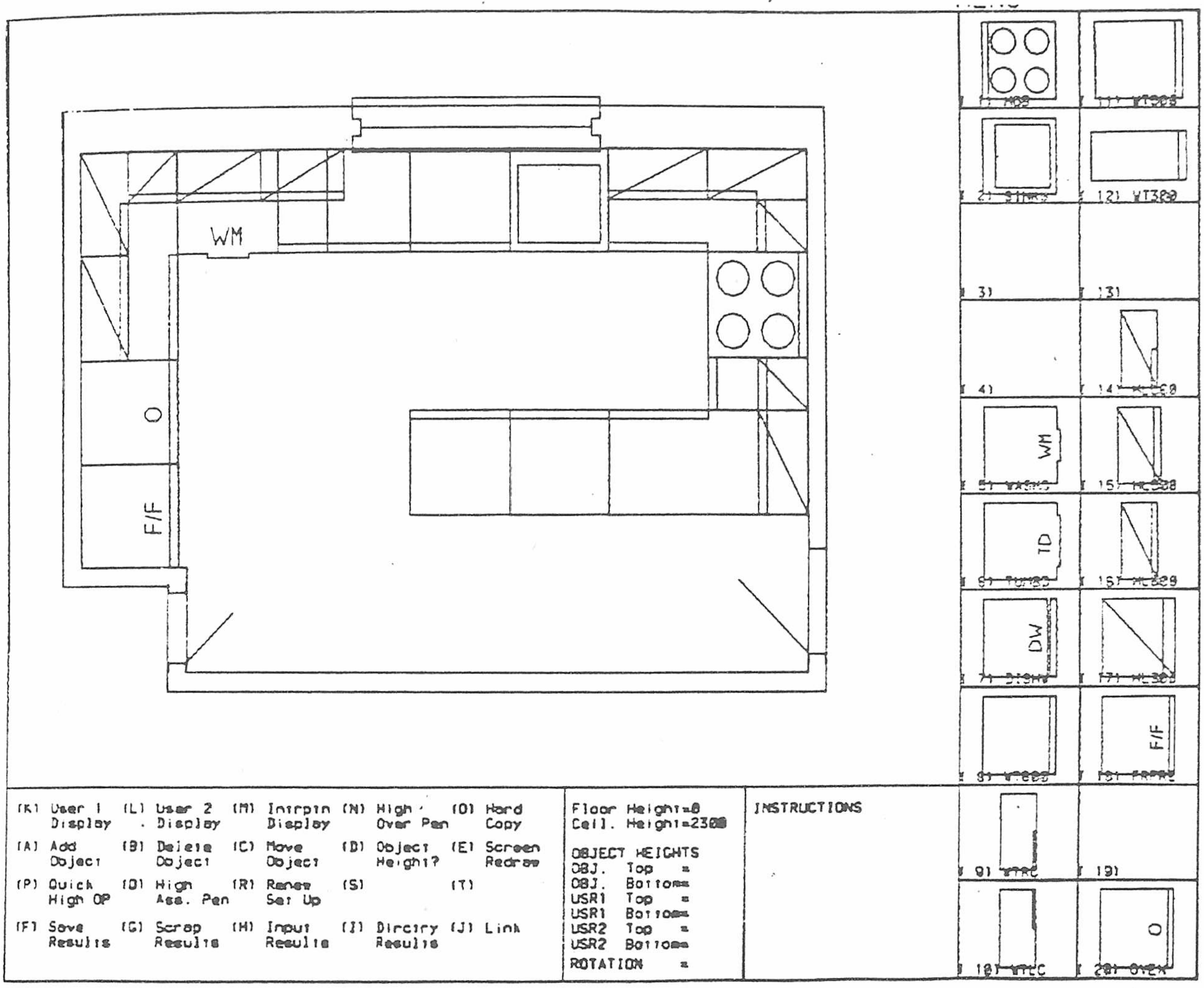


Fig 7.8 Graphical Display Keys

(E) SCREEN REDRAW

As is implied by the name, pressing this key causes a REDRAW of the entire screen followed by a return to the LAYOUT menu. The system has been designed to obviate the need to REDRAW because of multiple images on the screen; and as such, the key represents a hangover from the development of the system. However, it has been found useful when the memory limitations of the hardware cause a dumping of refresh graphics images into storage display. This can occur when a large number of objects (30+) are being manipulated on the screen.

A



B

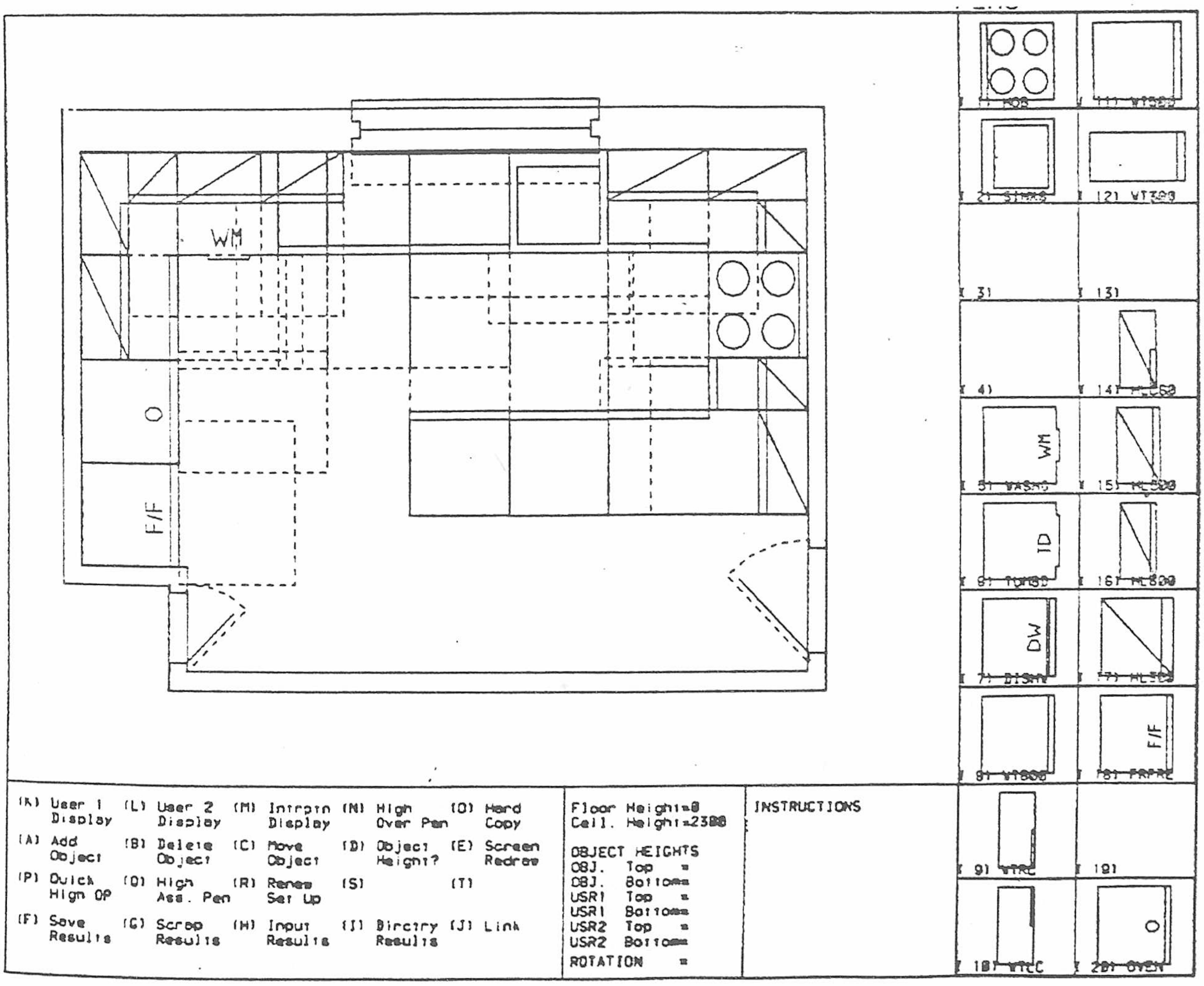
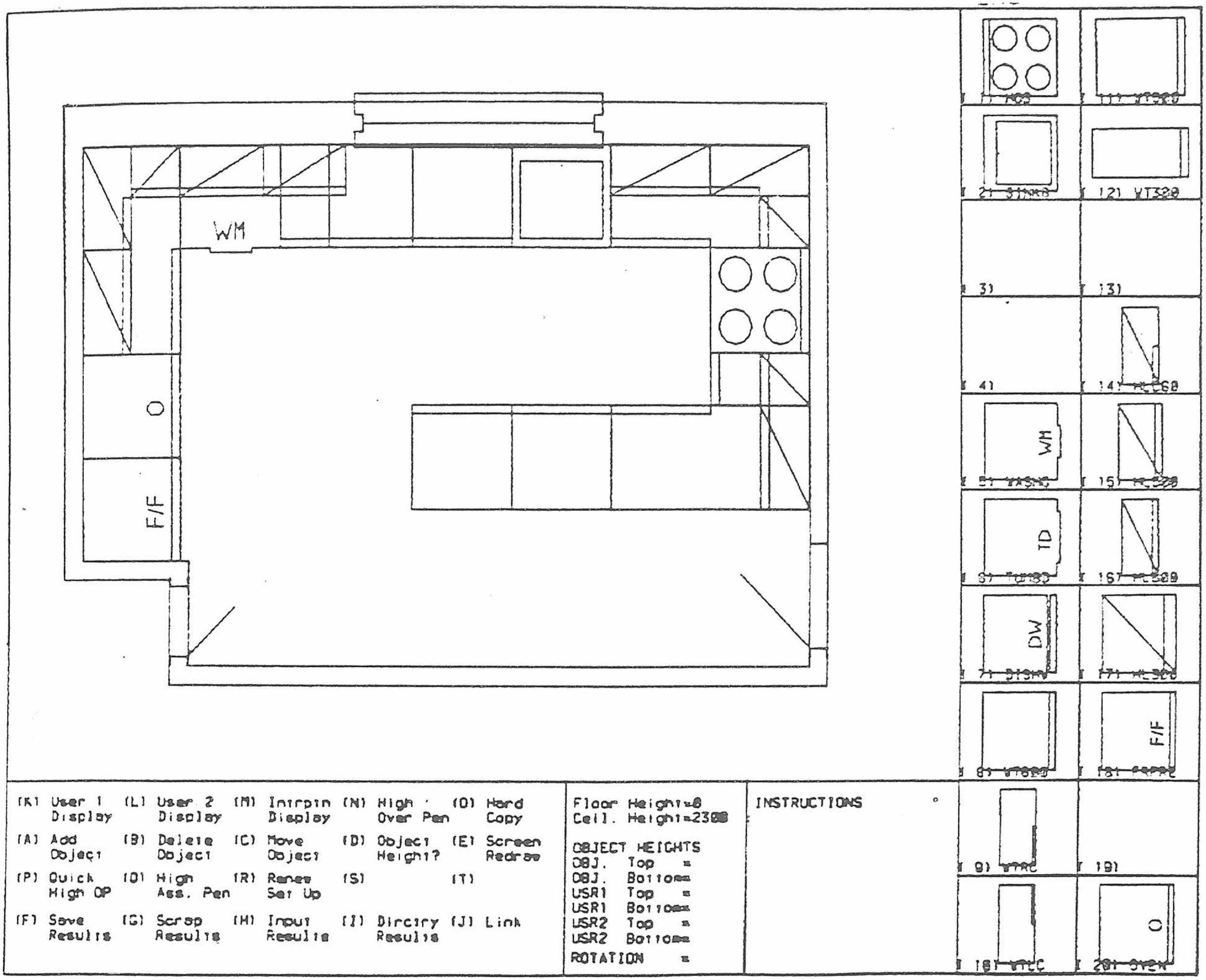
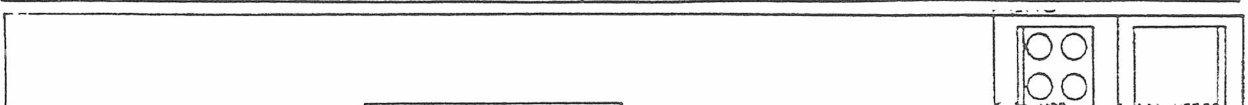


Fig 7.9 USER1 Graphics Comparison

A


WM
O
F/F
WM
TD
DW
F/F
O
3)
13)
4)
9)
19)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

B



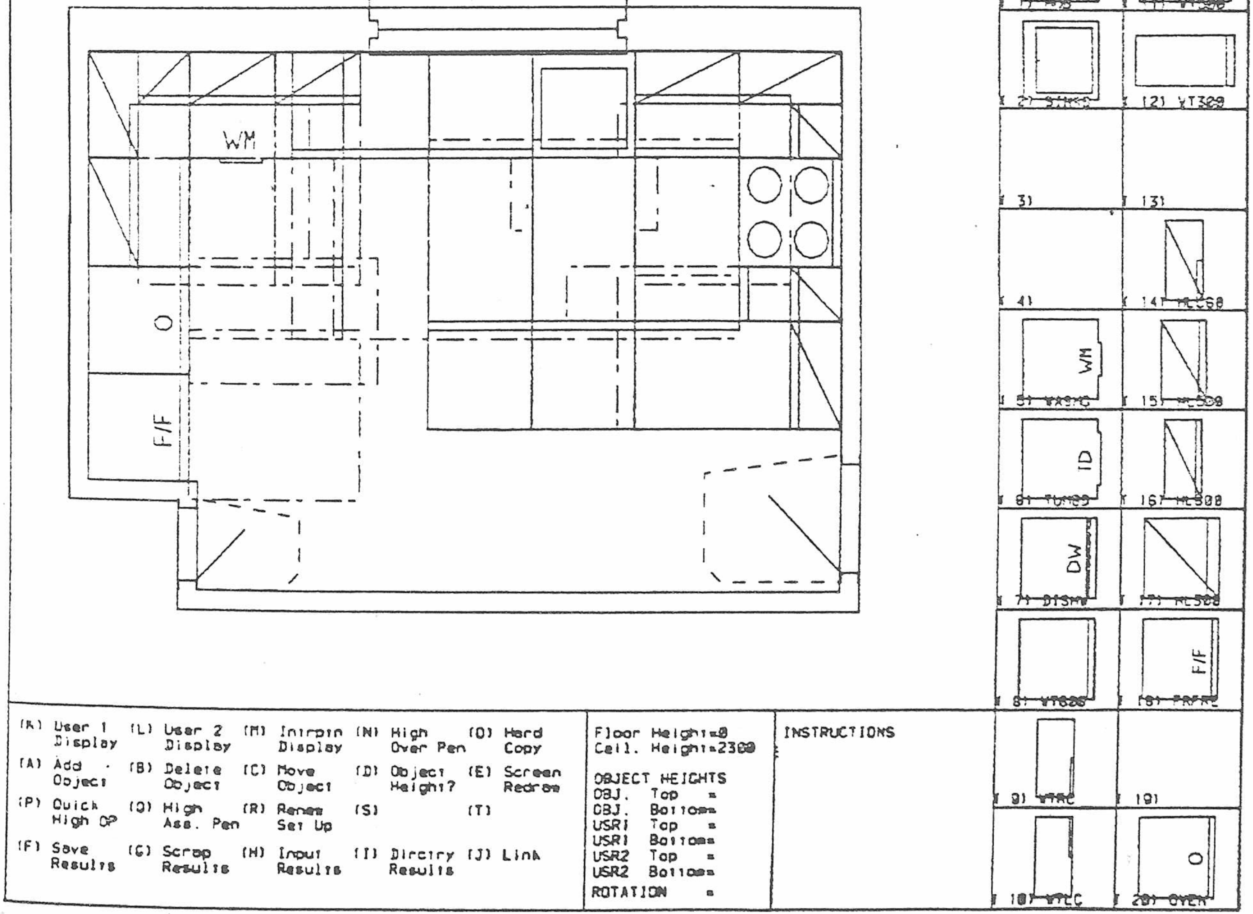


Fig 7.1Ø USER2 Graphics Comparison

(K) USER1 DISPLAY

The USER1 key is a toggle switch. The first time it is pressed the display section of the terminal screen is modified to show USER1 boundaries. The second time it is pressed the USER1 display is removed. The effects of this partial REDRAW is evident by the two halves of figure 7.9 overleaf. Part (A) is the normal display whilst part (B) also shows USER1 boundaries. After this REDRAW the user is returned to the LAYOUT menu.

(L) USER2 DISPLAY

This key operates in an identical manner to that of (K) USER1 DISPLAY. In figure 7.10 overleaf the change in display is shown. Again after the REDRAW the user is returned to the LAYOUT menu.

(O) HARD COPY

When this key is pressed the following message appears in the instruction box:-

ENSURE COPY UNIT IS
CONNECTED AND
SWITCHED ON

WAIT 2 MINUTES UNTIL
COPIER IS WARM

CR=COPY
0=RETURN TO MENU

Pressing zero returns the user to the LAYOUT menu. Pressing CR causes all refresh images in the display section of the screen to be 'fixed' to the screen (ie. converted from refresh to storage display). It also causes the current values of the efficiency measures to be printed out onto the screen in the instructions box. A hard copy is then automatically taken. A complete REDRAW is then done to reconvert storage images back to refresh images prior to returning the user to the layout menu.

7.4.3 Geometry manipulation

This section describes the operation of those keys outlined below in figure 7.11. The system ensures that $FLAYOUT.ADD program segment is present in memory when these keys are operated because it is this segment that contains most of the manipulation mechanisms.

(K) User 1 Display	(L) User 2 Display	(M) Intrpin Display	(N) High Over Pen	(O) Hard Copy
(A) Add Object	(B) Delete Object	(C) Move Object	(D) Object Height?	(E) Screen Redraw
(P) Quick High OP	(Q) High Ass. Pen	(R) Renew Set Up	(S)	(T)
(F) Save Results	(G) Scrap Results	(H) Input Results	(I) Dirctry Results	(J) Link

Fig 7.11 Geometry Manipulation Keys

(A) ADD OBJECT

Pressing this key allows the user to add a furniture object to the room display. It causes a small cross-hair cursor to appear on the screen together with the following instructions which appear in the instructions box.

USE CURSOR TO PICK
OBJECT TO BE ADDED

CR=PICK OBJECT
0=RETURN TO MENU

On pressing CR the system searches through the display and menu areas of the screen to locate which furniture object should be added.

Should the system fail to locate an object, a short message is printed in the instructions box together with a long error bell.

RETURNING TO MENU

Assuming that the system finds an object, the following message is displayed.

USE KEYBOARD TO
TRANSFORM OBJECT

SPACE=PUSH OBJECT
CR=PLACE OBJECT
0=RETURN TO MENU
1-6=ROTATION

At the same time, a refresh image of the selected furniture object appears in the centre of the screen.

Pressing keys 1-6 causes the furniture object to be rotated on the screen by one of the following amounts (measurement in degrees):- +90, +10, +1, -1, -10, -90.

Pressing the space bar causes the following message to appear together with the cross-hair cursor.

```
FIRST INDICATED EDGE
IS ALIGNED TO SECOND
INDICATED EDGE

CR=CONFIRM SELECTION
 0=RETURN TO MENU
```

If the system cannot find the edge indicated, a warning message is displayed together with a prompt indicating that either the first or the second edge should be indicated again.

Pressing CR 'fixes' the object. Location and orientation are entered into the data base. It is probably worth mentioning at this point that USER1 and USER2 displays can be turned on at this point by pressing either K or L.

Once the object is located, the system prints another prompt.

```
ENTER NEW HEIGHT
FOR TOP OF OBJECT
DISPLAY TO LEFT

CR=NO CHANGE
```

This gives the user the capability to amend the location of the z-axis reference point. The method of input is similar to that for entering file names as described earlier in section 7.4.1.

With the final CR the user is returned to the LAYOUT menu.

(B) DELETE OBJECT

Pressing this key allows the user to remove a furniture object from the room display. It causes a small cross-hair cursor to appear and the following instructions to be printed in the instructions box.

USE CURSOR TO PICK
OBJECT TO BE DELETED

CR=PICK OBJECT
0=RETURN TO MENU

The user uses the cross-hair cursor to select the object to be deleted. The room is then redrawn such that objects placed on the screen later than the object being deleted are shuffled forward in the display file overwriting all information concerning the deleted object.

Should the system fail to locate an indicated object, it issues a warning message to the user before returning him to the LAYOUT menu.

(C) MOVE OBJECT

Pressing this key allows the user to move a furniture object about the room display. It causes a small cross-hair cursor to appear and the following instructions to be printed.

USE CURSOR TO PICK
OBJECT TO BE MOVED

CR=PICK OBJECT
0=RETURNS TO MENU

Pressing CR causes the system to DELETE tha object and ADD a replica with the same location and orientation. The system then allows the user to continue with the ADD commands outlined previously.

(D) HEIGHT?

This key can be thought of as incorporating the last stage of the ADD command. Functionally, it is very similar, allowing the user to change the reference Z height of any existing furniture object within the room display.

The one additional function this key allows is that the user can interrogate the data base concerning the Z-parameters of room objects. Thus he can discover the min/max heights of objects such as doors, windows, floors and ceiling.

7.4.4 INTERPRETATION

This section describes the operation of those keys outlined below in figure 7.12. The system ensures that either $FLAYOUT.INT or $F2LAYOUT.INT program segments are in memory after any of these keys have been pressed, since they contain the interpretive mechanisms. The main mechanism being the algorithm to determine the area of overlap between two polygons. This algorithm is fully described in appendix F.

(K) User 1 Display	(L) User 2 Display	(M) Intrptn Display	(N) High Over Pen	(O) Hard Copy
(A) Add Object	(B) Delete Object	(C) Move Object	(D) Object Height?	(E) Screen Redraw
(P) Quick High OP	(Q) High Ass. Pen	(R) Renew Set Up	(S)	(T)
(F) Save Results	(G) Scrap Results	(H) Input Results	(I) Dirctry Results	(J) Link

Fig 7.12 Interpretation Keys

(M) INTERPRETATION DISPLAY

Pressing this key starts a lengthy process (some 20 minutes on average) of interpretation of the spatial efficiency of the room. Each furniture element is checked against the room geometry, the room objects and against every other furniture object. The system 'blinks' the refresh image of each object as it is being checked.

Before the user is returned to the LAYOUT menu a graphic display of the relative effect of each of the four efficiency measures is presented in the instruction box. This is followed by a numeric itemisation of each of the efficiency values.

Pressing this key updates all the values which will be printed when the user takes a hard copy.

(N) HIGH OVERLAP PENALTY

Pressing this key starts a very lengthy process of interpretation. Unlike the previous interpretation key, this key only examines overlap penalties. It determines which user area overlap pair has attracted the highest penalty.

The area of this overlap is then presented to the user as a hatched area on the display section of the screen.

The user is then given the option of returning to the LAYOUT menu or of determining which is the next highest overlap penalty pair.

To look at all overlapping penalty pairs takes a very long time indeed. For this reason, a quick overlap penalty appraisal measure was developed. This measure is discussed below.

(P) QUICK HIGH OP

Pressing this key allows the user to rapidly determine where the greatest overlap penalties occur.

When the room layout is interpreted, the interpretation algorithm adopts an incremental approach, whereby each furniture object has assigned to it, the total additional overlap penalty that has resulted from its placing in a particular location within a room layout.

The QUICK HIGH OP key accesses this data and presents to the user, via refresh 'blinking' of each object, a sequence of high to low overlap penalties.

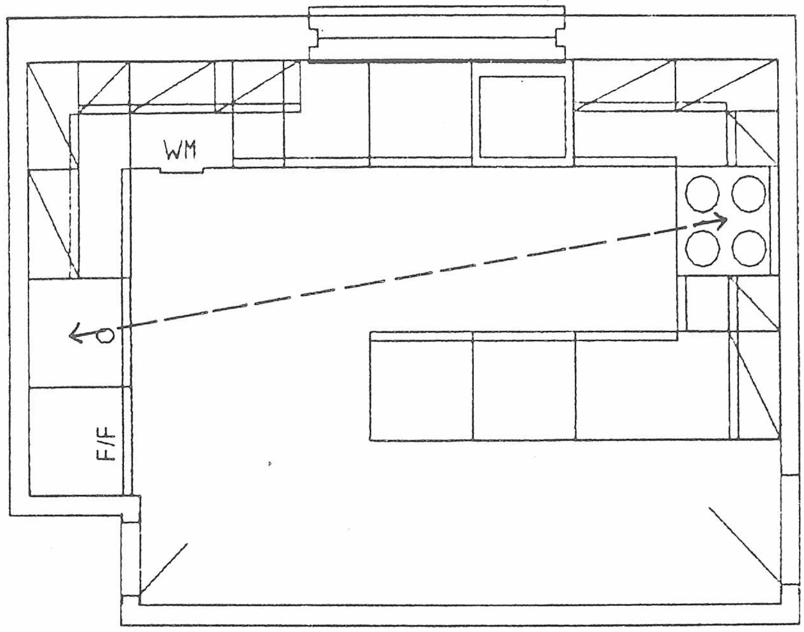
This process is obviously not as accurate as HIGH OVERLAP PENALTY as objects positioned later tend to overlap with more objects and consequently have a greater overlap penalty. However, it gives the user worthwhile appraisal of overlap penalties with the minimum of delay.

(Q) HIGH ASSOCIATION PENALTY

Pressing this key starts a rapid process of interpretation of association penalties.

The user is presented with a 'blinking' dashed double headed arrow joining the centrepoints of the two objects with the highest association penalty. The system then prompts the user to find the next highest association pair or return to the LAYOUT menu.

Fig 7.13 High Association Penalty



7.6.0 COMMENT ON CURRENT CONFIGURATION

Comment is limited in this section to those aspects of the software/hardware configuration discussed in this chapter.

Further comment on the system, whith reference to different configurations, is offered in Chapter 8, Section 8.2.0.

7.6.1 The main menu

Functions satisfactorily, though there is a need to extend the capabilities of the COPY function. Also, although swopping program segments in and out of memory is successful 99% of the time, the user can still run into a system (hardware) bug causing a total crash. Some effort needs to be devoted to surmounting this problem.

7.6.2 Layout

In terms of general comments about the layout suite; the main criticism is the requirement to start the suite off with the default room layout. On reflection, it would be better for the user if some of the initialisation proceedures of LAYOUT were pushed back into 'earlier' program segments. This would allow the user to start LAYOUT with the minimum of initialisation delay.

7.6.2 Layout file handling

This aspect is entirely satisfactory; the only minor point to be considered is that it might be desirable to be able to SCRAP all RESULTS files simultaneously.

7.6.4 Layout graphical display keys

This aspect is also satisfactory. Perhaps some 'tinkering' with the order that graphics commands are issued to the system would help to remove remaining problems associated with screen flicker.

7.6.5 Layout geometry manipulation

This aspect is adequate given the current level of system development. However, two aspects clearly need further development.

1. It should be possible to avoid the need for object shuffling due to deletion of an object.

2. The controlling algorithm for the push command in ADD object should be amended to improve its performance for non-rectilinear geometries.

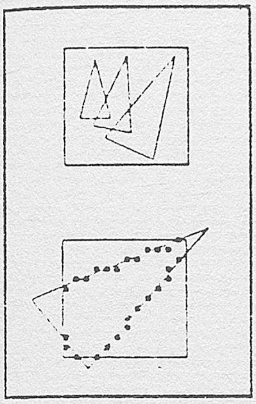
   The author is aware that others have solved the problem of non-rectilinear pushing by treating rectilinear and non-rectilinear movement as seperate cases (7.1). However, the author has demonstrated within KAPABLE that combining of rectilinear and non-rectilinear movement is possible within the same algorithm and using the same input information.

7.6.6 Layout interpretation

Given the existing system configuration and the lack of raw computing power that it imposes, interpretation aspects are adequate.

7.7.0 REFERENCES

(7.1) User's Manual for Gable, Draft Edition, Oct. 1981, Genesis Ltd., UK.



CHAPTER 8

SCOPE FOR FUTURE WORK

CONTENTS

8.1.0 INTRODUCTION

In this chapter we examine the scope for future work arising from this research project. We examine this work under three broad classifications:-

1. Future work as regards implementing the existing computer model on a more powerful hardware configuration.

2. Future work as regards potential applications of a commercially developed system.

3. Long term applications as it interfaces with the work of others.

8.2.0 IMPLEMENTATION ON IMPROVED HARDWARE

In this section we consider a hardware configuration which might be more suited to the application. After detailing harware improvements we will consider how this would affect the software implementation.

8.2.1 Hardware improvements

Hardware can be considered under four seperate headings:-

1. Input device.
2. Display.

3. Memory.
4. Off-line storage.

Most of the currently available input devices are variations on thumbwheels, joysticks (which includes the mouse since it can be considered to be a very sophisticated joystick) or lightpens. Substitution of the thumbwheel input device, used in the current implementation, by either of the two latter devices would not appreciably improve the system. However, it should be noted that, software changes allowing a selection of input devices to be used, to suit the personal preference of the designer, would be an improvement in the system.

One interesting new input device which is radically different from those mentioned above is a device which electronically tracks eye movement to locate a cross-hair cursor on a screen. This device is not yet commonly available but a prototypical version was demonstrated on the TV program Tomorrows World as an aid to handicapped children learning to read and is somewhat similar in operation to the stereoscopic display system discussed by Mitchell (8 .1).

In terms of display others (8.2, 8.3) have argued for a VDU which is very similar to a drawing board and operated with a light pen.

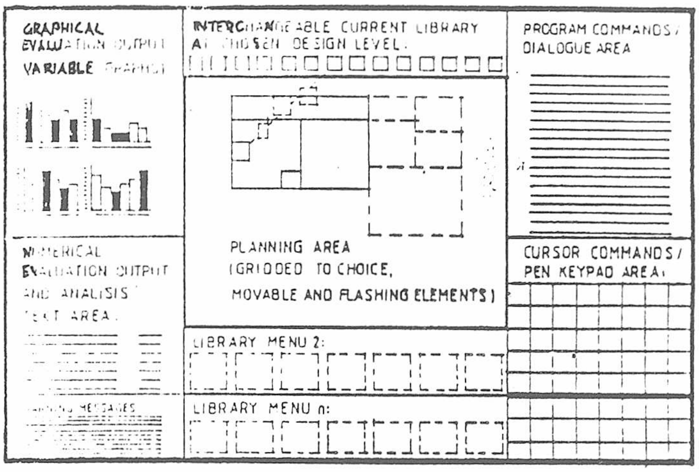


Fig.8.1 VDU like a drawing board (after Langskog)

Although this type of approach might be less intimidating to the designer who is used to a traditional drawing board, it confers no great ergonomic advantage, possibly the reverse.

Rather a colour raster VDU would be the preferred display device. The screen should be large, possibly the dimensions of an A3 sheet of paper, and the resolution high so as to accomodate detail at quite large scales. Since the system is schematic and representational there is no great requirement for a large palette of colours - eight would suffice.

In terms of RAM, the more the better. It is estimated that at least four times the currently available RAM would be required and in the light of some of the software changes outlined in Section 8.2.2, 512k might be more realistic. With this amount of memory, it would be possible not only to improve the user display but also ensure that object and geometry files can be retained in core memory, thus speeding up the process of geometry manipulation.

The current level of processor speed (9600 baud) is sufficient for the application. However parallel processing might be particularly appropriate to this system. For example, when determining the overlap penalty (see Appendix F) several processors could each calculate overlap areas for a single furniture element. These penalties could then be totalled for the overall overlap penalty.

In terms of off-line storage the single floppy disk drive and manager unit used in the current implementation are clearly inadequate. Having said that, given that the system had more RAM there should be no need to change to hard disks. A file manager incorporating three floppy disk drives would be sufficient.

It might be desirable to have additional solid state memory such as the Option 28, 512k Extended memory file manager offered by Tektronix for use with the 4054 (8.4). This option functions in a manner somewhere between a built in hard disk and additional RAM. This form of memory might prove useful for holding the coding, thus allowing more rapid overlapping of program segments, and for temporary holding of system variables and intermediate results.

8.2.2 Software improvements

Possible improvements to the software can be considered under three headings:-

1. Enhancement of prototypical routines.
2 Improvemnts as a result of enhanced hardware capabilities.
3. Improvements which incorporate both of the above features.

The system needs development of its geometry input segments, particularly as regards scaling, windowing and viewporting.

Contrary to my earlier beliefs (see Appendix A) routines should be made available to allow the 'expert' user to enter geometry descriptions in an alphanumeric format. Geometric feedback of the input should still be available. The expert is likely to find this a more rapid means of input, at least with some geometries, than a purely interactive input mechanism.

In terms of output, it would be an improvement to have a mechanism which allowed the visual presentation of the model's three dimensionality, such as perspectives, part elevations and lid-offs.

A direct consequence of improving the off-line storage facilities of the system would be the better organisation and manipulation of files. With three disks available for use, one could be reserved for source code, temporary system variables and controlling files, another could be reserved for object descriptions, menus and matrices whilst the last disk could be reserved for 'job specific' information - room geometry and results. An indirect benefit of multi-disk availability would be that the copying of files from one disk to another would be greatly simplified.

As a result of a raster display system, perhaps aided by parallel processing, a more powerful interpretation routine could be devised possibly based on pixel counting.

In addition, the continuous feedback incorporated in the system (see Chapter 5, Section 5.4.2) could be machine rather than user driven. That is, the computer could interrogate the cursor position automatically on a time basis. This would improve the geometry input routines considerably.

Another improvement made possible by an increase in processor speed and power would be the introduction of user experience levels. That is, the degree of user feedback is determined by the user. A novice user would receive all prompts automatically whilst an expert user would receive very few abbreviated prompts.

A general increase in hardware capability would also allow the system to be extended by providing interfacing routines to other commercial packages.

8.2 3 Conclusion to system improvement

The improvements outlined in the previous two sections indicate the amount of work required to raise the currently implemented prototypical system to the level required of a commercial package.

8.3.0 POTENTIAL APPLICATION OF SUCH AN ENHANCED SYSTEM

This section outlines the applicability of an enhanced system.

Given the current level of cuts in local authority housing budgets, the system is unlikely to be used in the design of public sector housing in the UK. There is more scope for this type of application abroad. Indeed, the Italians (see Chapter 4, Section 4.3.1) have this application very much in mind for their software.

A more likely application would be in the field of facilities planning. That is, in the design of the internal layout of offices, factories and commercial premises where space is often at a premium and the efficient use of space is usually related to functional requirements rather than personal preference.

Currently the prototypical system is being used as part of an ongoing research project at SSSA investigating the design of medical installations on offshore drilling rigs (8.5). Here is an ideal situation in which to use the activity space model, since the efficient use of space is clearly very important. It will be interesting to see how the model performs in a non-domestic application.

This application is illustrated in figures 8.2 and 8.3 below.

The algorithm could also be used for space planning outside the buiding. For instance, it could be used to aid in the planning of housing estates. The site would be defined as the 'room' and each house would be an individual 'furniture' element. By choosing appropriate user areas and association distances, the housing layout could be planned to obviate overlooking. Already, the Regional Estate Monitoring Unit of the PSA has expressed an interest

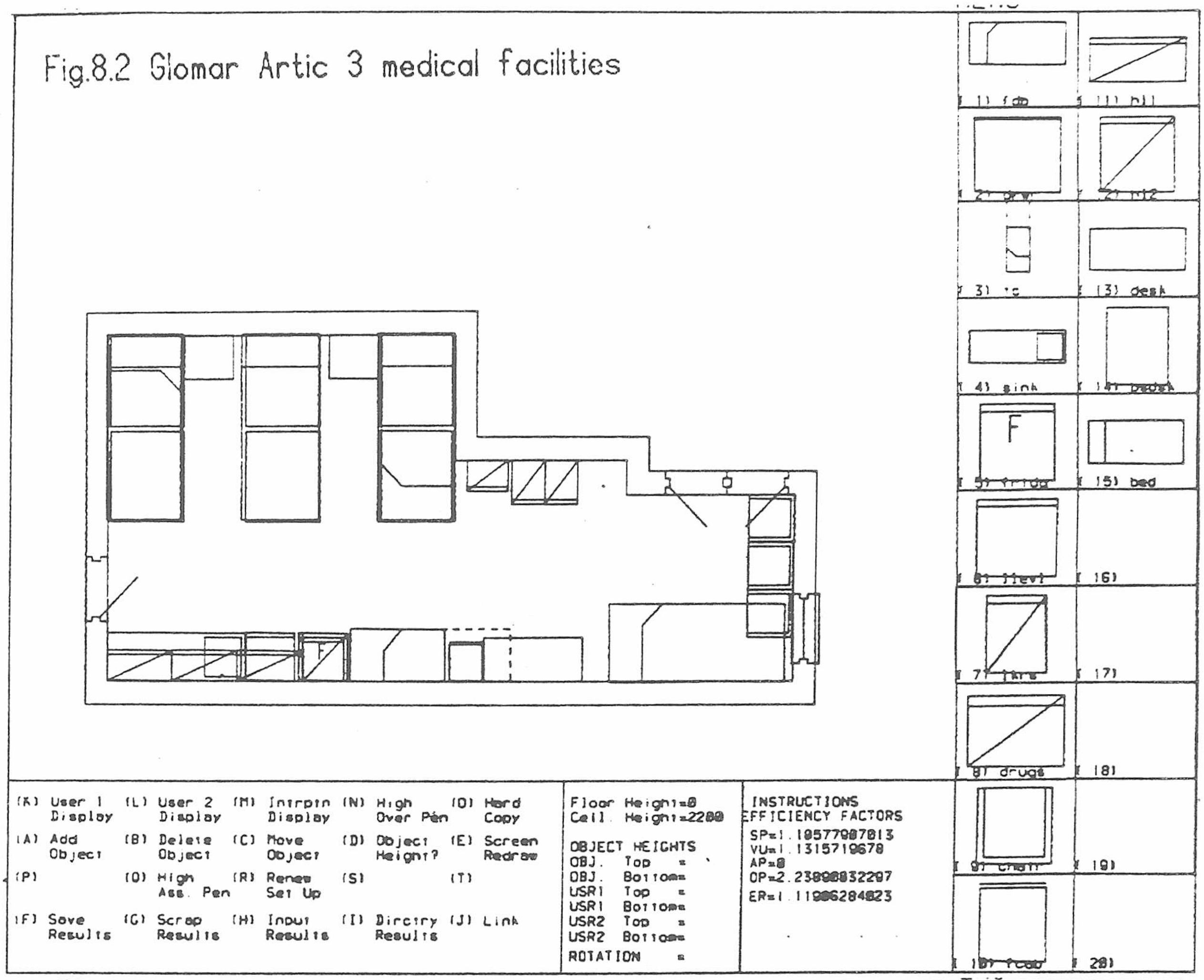


Fig.8.2 Glomar Artic 3 medical facilities

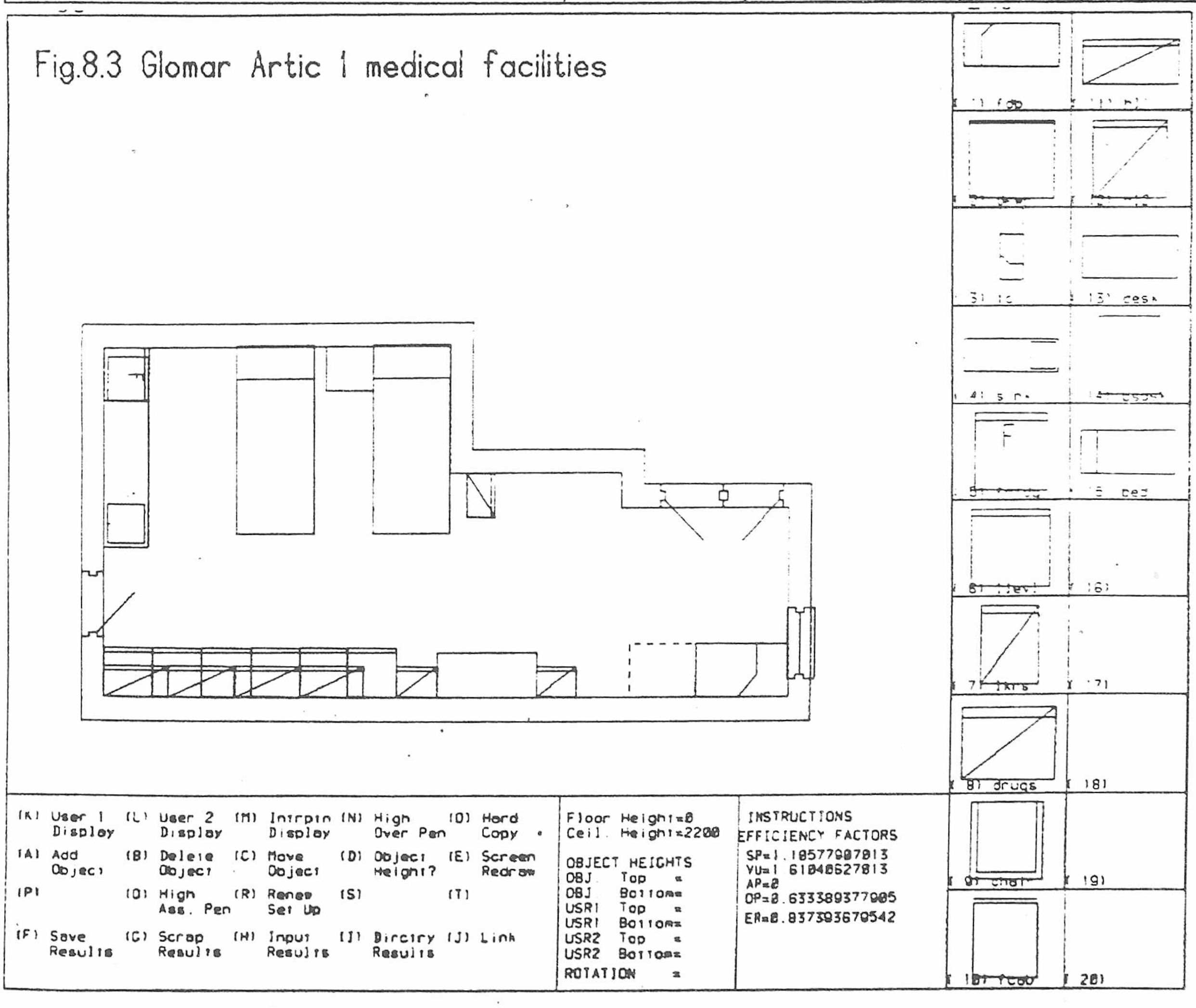


Fig.8.3 Glomar Artic 1 medical facilities

in using the system for such a purpose.

However, the core of the algorithm is not limited in application to the building industry. It would be possible to use such an algorithm in many layout/planning situations. For example, it could be utilised to aid in the laying up of garments where the VU value gives a good measure of the area of wasted cloth. It might even be possible to utilise the algorithm for laying out air traffic routes or electronic circuit boards where interference of 'user areas' would have to be minimised.

Indeed, the system will find a use in any application in which the interference of user areas is undesirable, in which the association of elements is important or in which the laying out of elements with minimum wastage is desirable.

8.4.0 LONG TERM APPLICATION

In this section we examine the scope of the algorithm as it interfaces with the work of others - particularly within the emergent fields of expert systems and artificial intelligence.

Feigenbaum gives as a definition:-

> 'An expert system is an intelligent computer program that uses knowledge and inference proceedures to solve problems that are difficult enough to require significant human expertise for their solution. The knowledge necessary to perform at such a level, plus the inference proceedures used, can be thought of as a model of the expertise of the best practicioners in the field'(8.6).

This is an adequate definition of an expert system incorporating deductive artificial intelligence. That is, where the machine is capable of deducing from the general rule base, representing human expertise, the outcome of a specific situation.

It also highlights one of the major problems with this form of expert system - the abstraction of the 'expertise' used by the best practioners and its assembly into a set of rules.

The algorithm presented earlier in this thesis could be of undoubted assistance in this regard since it gives an objective measure of layout efficiency. By combining objective evaluations of many designs of a particular room type with the subjective assessment of the experts, the process of rule definition might be eased.

The algorithm might be even more useful in conjunction with an expert system using the inductive rule approach. In this situation, instead of defining a set of general rules and using them to deduce facts about a particular case, the computer examines a number of specific cases and induces the rules; ie. it 'learns' from experience.(8.7)

It is clear that the algorithm might prove to be a useful starting point for such an inductive system. For example, the association distances for each element pair might be loaded into the expert system as a series of rules; the penalty factors for each pair might then be determined by the expert system by case interpretation.

8.5.0 REFERENCES

(8.1) Mitchell,W.J. Computer Aided Architectural Design, Petrocelli/Charter, New York, 1977, p.367

(8.2) Langskog,K., A Computer Aided Architectural Design System for the Analysis, Synthesis and Appraisal of Domestic Activity Spaces and House Plans, Ph.D. Thesis, Scott Sutherland School of Architecture, RGIT, Aberdeen, 1981, pp.226,228.

(8.3) Leifer,D.M., The Machine Refinement of Raw Graphic Data for Translation into a Low Level Data Base for Computer Aided Architectural Design (CAAD), Ph.D. Thesis, Scott Sutherland School of Architecture, RGIT, Aberdeen, 1984, pp.45,50.

(8.4) Tektronix, Tek Products 1984, Tektronix Inc., Oregon, USA, 1984, p.130

(8.5) Elliot,T.P, Space Requirements and Functional Constraints Affecting the Design of Medical Installations on Offshore Drilling Rigs, ongoing Ph.d. research at Greys School of Art jointly with Scott Sutherland School of Architecture, RGIT, Aberdeen, Registered 1984.

(8.6) Fiegenbaum,E.A, 'Expert Systems in the 1980's', in Machine Intelligence, (Ed.) A. Bond, Infotec State of the Art Report, Pergamon-Infotec, 1981

(8.7) Lamb,J., 'Making Friends with Intelligence', New Scientist, (Ed.) Kenward,M., 22nd. Aug. 1985, No.1470, London, pp.30,32

SECTION R

BIBLIOGRAPHY AND ABBREVIATIONS

CONTENTS

R.1.0 BIBLIOGRAPHY

This bibliography contains all the references appended to the end of each chapter and appendix. It also contains works consulted during the course of the research project which were found to be useful.

Abrams, M.D., Stein, P.G., Computer Hardware and Software: An Interdisciplinary Introduction, Addison-Wesley, London, 1973.

ACM/SIGGRAPH Workshop, Battelle Seattle Conference Centre, June 1982, 'Graphical Input Interaction Technique (GIIT)', Reported in Computer Graphics, January 1983.

Adams, J.A., Rogers, D.F., Computer Aided Heat Transfer Analysis, McGraw Hill, New York, 1973.

Aish, R., 'Initial Development of a Participatory Computer Based Design Aid', ABACUS Occasional Paper No. 72, University of Strathclyde, Glasgow, 1979.

Angell, I.O., A Practical Introduction to Computer Graphics, Macmillan, London, 1981.

Auger, B., The Architect and the Computer, Pall Mall Press, London, 1972.

Baer, J.L., Computer Systems Architecture, Computer Press, 1980.

Baxter, R., Computer and Statistical Techniques for Planners, Methuen, London, 1976.

Bensasson, S., Micros in Construction: A Study of the Effect of Microelectronics and Information Technology on the Construction Industry, DOC, Cambridge, 1980.

Berztiss, A.T., Data Structures: Theory and Practice, Academic Press, London, 1971.

Bijl, A., et al., 'Building Description Techniques to Support Computer Aided Design and Production', Section 25 of SERC Grant Application, EdCAAD, University of Edinburgh, December 1981.

Bijl, A., Stone, D., Rosenthal, D., 'Integrated CAAD Systems', Final Report to DOE on funded research project DRG 470/12, EdCAAD, University of Edinburgh, March 1979.

Bitter, G.G., Basic for Beginners, 2nd. Edition, McGraw Hill, New York, 1980.

Bjorktu, R., 'Vurdering av boligens bruksveroi - metodesporsinal (Evaluation of the Usable Value of the Dwelling Plan), BD Orientering (Reports from the Norwegian Directorate of Housing), No. 3, 1963.

Bowyer, A., A Programmers Geometry, Butterworth, London, 1983.

Bridges, A. (Ed.) 'Draughting Systems', BoCAAD, ABACUS, University of Strathclyde, Glasgow, No. 24 May 1977.

Bridges, A. (Ed.) 'Man Machine Interaction in Problem Solving', BoCAAD, ABACUS, University of Strathclyde, Glasgow, No. 24 May 1977.

Bridges, A.H., 'A New Approach Towards the Computer Aided Analysis of Architectural Design Problems', CAD 78 - Proceedings, Science and Technology Press, 1978.

British Standards Institution, BS 4821, Recommendations for The Presentation of Theses, London, BSI, 1972.

British Standards Institution, BS 6222 Part 1: 1982, Domestic Kitchen Equipment: Specification for Coordinating Dimensions, London, BSI, amended 1985.

British Telecom, Yellow Pages Aberdeen 1984, British Telecommunications plc., London, 1984.

Cakin, S., 'Decision making in Design: Evaluation of Holiday Houses', DMG/DRS Journal, Vol. 9, No. 2, Berkeley, California, 1975.

Campion, D., Computers in Architectural Design, Elsevier, 1968.

Carolin, P. (Ed.), 'Update: Domestic Kitchen Design', _Architects Journal_, No.40, Vol.180, 3 Oct. 1984.

Carolin, P. (Ed.), 'Conventional Planning', _Architects Journal_, No.40, Vol.180, 3 Oct. 1984.

Chandler, G., _How to Find Out_, 3rd. Edition, Pergamon Press Ltd., 1967.

CICA, 'Fact Sheet on the Use of Computers in the Construction Industry', _CICA Bulletin_, CICA, Cambridge, Mar. 1982.

Clocksin, W.F., Mellish, C.S., _Programming in PROLOG_, Springer-Verlag, 1981.

Coutts, N., 'Low Cost Computing - 2.Analysing your Requirements', _Architects Journal_, Vol. 177, No. 5, 2nd. Feb. 1983.

Cross, N., _The Automated Architect: Human and Machine Roles in Design_, Pion Ltd. 1977.

Danish National Institute of Building Research, Report 60, _User Study in the housing estate Vaerebo Park_, Copenhagen, The Institute, 1969.

Dawson, F.T., 'CAD - A Users Perspective', _Computer Aided Design_, Vol. 12, No. 3, May 1980.

DEC, _Basic Plus 2: Language Manual Users Guide_, DEC, Maynard, Mass., 1977.

DEC, _Edit Reference Manual_, DEC, Maynard, Mass., 1978.

DEC, _Edit Users Guide_, DEC, Maynard, Mass., 1978.

DEC, _Getting Started with RUNOFF: Text Formatting Program_, DEC, Maynard, Mass., 1978.

DOE, _Design Bulletin No.6 Space in the Home: Metric Edition_, London, HMSO, 1968.

DOE, _Bibliography on the Application of Computers in the Construction Industry_, 2nd.Edition, DOE, London, 1962-71.

DOE, _Computer Program Descriptions for Architects_, DOE, London, 1972.

DOE, Design Bulletin No.24 Part 2, Spaces in the Home: Kitchens and Laundering Areas, London, HMSO, 1972.

DOE, Final Report by the Standing Committee on Computing and Data Coordination, Computing and Communication in the Building Industry, DOE, London, 1980.

Eastman, C.M., 'General Purpose Building Description Systems', Computer Aided Design, Vol. 8, No. 1, January 1976.

Dromey, R.G., How to Solve It By Computer, Prentice/ Hall International, London, 1982.

Eastman, C.M., 'The Concise Structuring of Geometric Data for Computer Aided Design', Data structures, Computer Graphics and Pattern Recognition, Klinger et al. (Eds.), Academic Press, London, 1977.

Eastman, C.M., 'Prototype Integrated Building Model', Computer Aided Design, Vol. 12, No. 3, May 1980.

Electricity Council, Guidelines to Kitchen Planning, Electricity Council, Free.

Elliot, T.P., Space Requirements and Functional Constraints Affecting the Design of Medical Installations on Offshore Drilling Rigs, ongoing Ph.D. research, Greys School of Art jointly with Scott Sutherland School of Architecture, RGIT, Aberdeen, registered 1984.

Emery, G. Electronic Data Processing, Pitman, London, 1968.

Fairweather, L., Sliwa, J.A., AJ Metric Handbook, 3rd. Edition, The Architectural Press, London, 1977.

Fiegenbaum,E.A, 'Expert Systems in the 1980's', in Machine Intelligence, (Ed.) A. Bond, Infotec State of the Art Report, Pergamon-Infotec, 1981.

Fielden, C.J., Ede, T.J., Computer Based Energy Management in Buildings, Pitman, London, 1982.

Forsythe, A.I., et al., Computer Science: BASIC Language Programming, John Wiley and Sons Inc., New York, 1960.

Forsythe, R., The Basic Idea: An Introduction to Computer Programming, Chapman and Hall, London, 1978.

Forsythe, R.(Ed.), Expert Systems: Principles and Case Studies, Chapman and Hall, London, 1984.

Foy, N., Computers and Commonsense, Longman, London, 1972.

Gambro, G., Gambro, P., Pienovi, C., Un Sistema CAAD per la Verifica della Qualita' Residenziale (A CAAD System for the Verification of Residential Quality), Istituto per la Matematica Applicata (Institute of Applied Mathematics), Genova, Italy, N.163(84), 1984.

Genesis Ltd., Users Manual for Gable, Draft Edition, Genesis Ltd., UK, Oct. 1981

Gero, J,S., Computer Applications in Architecture, Architectural Science Series, Applied Science Publishers, London, 1977.

Giloi, W.K., Interactive Computer Graphics, Prentice Hall, 1978.

Guedj, A.(Ed.), IFIP Workshop Methodology of Interaction, North Holland Publishing Co., Amsterdam, 1980.

Gutteridge, B., Wainright, J.R., Computers in Architectural Practice, Crosby Lockwood Staples, London, 1973.

Halasz, F., Moran, T., 'Analogy Considered Harmful', Proceedings Conference on Human Factors in Computer Systems, Gaitherburg, Maryland, 1982.

Hales, H.L., Industrial Engineering 9, Computer Aided Facilities Planning, Marcel Dekker Inc., New York, USA, 1984.

Hamilton, I., Computer Draughting for the Building Team, Construction Industry Computing Association, Cambridge, 1981.

Hammond, B.G., Techniques for the Input of Geometry Data in Computer Aided Architectural Design: Report No. 1 (unpublished), Scott Sutherland School of Architecture, RGIT, Aberdeen, March 1983.

Hammond, B.G., 'Feedback Aids to Geometry Data Input', Occassional Paper No. 4, (unpublished), Scott Sutherland School of Architecture, RGIT, Aberdeen, March 1983.

Hammond, B.G., Techniques for the Input of Geometry Data in Computer Aided Architectural Design: Report No. 2, (unpublished), Scott Sutherland School of Architecture, RGIT, Aberdeen, July 1983.

Hammond, B.G., 'Techniques for the Input of Geometry Data in Computer Aided Architectural Design: Report No. 3', Occassional Paper No. 11, (unpublished), Scott Sutherland School of Architecture, RGIT, Aberdeen, October 1983.

Hammond, B.G., Leifer, D., 'A Graphics Interface to Complement Traditional Input Techniques', PARC 83 Proceedings, Online Conferences Ltd., Pinner, 1983.

Haskell, R.E., Apple Basic, Prentice Hall, Englewood Cliffs, NJ, 1982.

Harris, D., Computer Graphics and Applications, Chapman and Hall, London, 1984.

HDD Occasional Paper 2/81, A Survey of Tenants Attitudes to Recently Completed Estates, London, HMSO, 1981.

Hillier, B., Leaman, A., The Architecture of Architecture: Foundations of a Mathematical Theory of Artificial Space, Report to the RIBA Intelligence Unit, 1973.

HMSO, The Building Standards (Scotland) Regulations 1981, HMSO, London, 1981, Amended 1982,1984.

Hopgood, F.R.A. et al., Introduction to the Graphic Kernal System (GKS), Academic Press, London, 1983.

House, C.W.(Ed.), Interactive Computer Graphic Systems, Petrocelli, New York, 1982.

Humcke, D.J., 'Interactive Computer Graphics in CAD', Computer Aided Design, Jones, G.W.(Ed.), IPC Business Press Ltd., Guildford, Volume 8, No. 4, Oct. 1976.

Hunt, R., Computers and Commonsense, 2nd.Edition, Prentice Hall, London, 1979.

Hutton, G., Rostron, R.M., Computer Programs for the Building Industry, Hutton and Rostron/The Architectural Press/McGraw Hill, London, 1974.

Jones, R.M., Introduction to Computer Applications using Basic, Allyn and Bacon, Boston, Mass., 1981.

Jones, R.M., Face Report No. 3: Computers in Engineering and Architecture, in Japan, FACE, Melbourne, 1983.

Kemeny, J.C., Thomas, E.K., Basic Programming, 3rd. Edition, Wiley, New York, 1980.

Krantzer, K.R., 'Kitchens with a frontal siting of a diner space', Wirtschaftlich Bauen, October 1968.

Kunii, T.L., Kunii, H.S., 'Architecture of a Virtual Graphic Database System for Interactive CAD', Computer Aided Design, Vol. 1, No. 3, May 1979.

Lafue, G.M.E., 'Integrating Language and Database for CAD Applications', Computer Aided Design, Vol. 11, No. 3, May 1979.

Laing, L.W.W., et al., 'Enigma; an Enhanced Interpretive Graphics Module for Architects', Occassional Paper No. 10, Scott Sutherland School of Architecture, RGIT, Aberdeen, 1983.

Lamb,J., 'Making Friends with Intelligence', New Scientist, (Ed.) Kenward,M., 22nd. Aug. 1985, No.1470, London.

Lansdown, J., Expert Systems: Their Impact on the Construction Industry, Report to the RIBA Conference Fund, 1982.

Langskog, K., A Computer Aided Architectural Design Technique for the Analysis, Synthesis and Appraisal of Domestic Activity Spaces and House Plans, (Ph.D. Thesis), Scott

Sutherland School of Architecture, RGIT, Aberdeen, 1981.

Langskog, K., Laing, L.W.W.,'A Computer Aided Architectural Design Technique for the Appraisal of Domestic Activity Spaces', CAD 84 Proceedings, Butterworths, 1984.

Laver, F.J.M., Introducing Computers, 2nd. Edition, HMSO, London, 1973.

Lawson,B., How Designers Think, Architectural Press, London, 1980.

Lawson, B., Selby, P.L., The Evaluation in Architectural Practice of a Desktop (Microprocessor) Computer Aided Architectural Design System, A Report to the Architects Registration Council of the United Kingdom, 1980.

Lawson, B.R., Selby, P., Davies, C.J., 'Tektronix 4050 Series Desk Top Computers', BoCAAD, Bridges, A. (Ed.), ABACUS, University of Strathclyde, Glasgow, No. 36, May 1980.

Lawson, B.R., Riley, J.P., 'ISAAC A Technique for the Automatic Interpretation of Spaces from Drawn Floor Plans', CAD 82 Conference Proceedings, Pipes, A.(Ed.), Butterworths, 1982.

Lawson, B.R., Riley, J.P., 'RODIN A System of Modelling Three Dimensional Roof Forms', CAD 82 Conference Proceedings, Pipes, A.(Ed.), Butterworths, 1982.

Leifer, D., 'Man Before the Machine', BoCAAD, University of Strathclyde, Glasgow, Vol.43-46, Feb.1983

Leifer, D. The Machine Refinement of Raw Graphic Data for Translation into a Low Level Data Base for a CAAD Decision Matrix, (Ph.D Thesis), Aberdeen, Scott Sutherland School of Architecture, Robert Gordons' Institute of Technology, 1984.

Leighton, N.L., Computers in the Architectural Office, Van Nostrand Reinhold Co., New York, 1984.

Longley, D., The Microcomputer Users Handbook, Macmillan, London, 1984.

Mitchell,W.J. Computer Aided Architectural Design, Petrocelli/Charter, New York, 1977.

Muther, R., Hales, H.L., Systematic Planning of Industrial Facilities, Vol I and II, Kansas City, M.O., Management and Industrial Research, 1980.

Neundorf, N., Computer Aided Drawing using the Tektronix Graphic System, Prentice Hall, 1983.

Newman, W., Sproull, R., Principles of Interactive Graphics, McGraw Hill, New York, 2nd Edition, 1979.

Newman, W., 'Some Notes on User Interface Design', Methodolody of Interaction - IFIP Workshop, (Ed.) Guedj, R.A., North Holland Publishing Co., Amsterdam, 1980.

Nievergilt, J., Weydert, J., 'Sites, Modes and Trails; Telling the user of an interactive system where he is, what he can do, and how to get places', Methodolody of Interaction - IFIP Workshop, (Ed.) Guedj, R.A., North Holland Publishing Co., Amsterdam, 1980.

Ohsuga, S., 'Towards Intelligent Interactive Systems', Methodolody of Interaction - IFIP Workshop, (Ed.) Guedj, R.A., North Holland Publishing Co., Amsterdam, 1980.

Parkins, R.P., 'BIBLE: A Computer Program for Generating Perspective Vi ews of Buildings', Occasional Paper No. 75, ABACUS, University of Strathclyde, Glasgow, 1979.

Paterson, J., Architecture and the Microprocessor, Wiley, Chichester, 1980.

Pemberton, J.E., Undertaking Enquiries, Part 2: Literature Search and Compiling a Bibliography, The Open University Press, Milton Keynes, 1974.

Paulidis, T., Algorithms for Graphics and Image Processing, Computer Science Press, 1981.

Peters, G., Construction Project Management using Small Computers, Architectural Press, London, 1984.

Pohl, J., Introduction to Computer Systems and Programming in Architecture and Construction: Fortran IV and BASIC, EDUCOL, California, 1978.

Ralston, A.(Ed.), Encyclopedia of Computer Science, Petrocelli/ Charter, New York, 1976.

Ray-Jones, A.R., Clegg, D., SfB Agency, CI/SfB Construction Indexing Manual, RIBA Publications, 1976.

Reynolds, R.A., Computer Methods for Architects, Butterworth, London, 1980.

Rosenthal, D., et al., 'Integrated CAAD Systems', BoCAAD, Bridges, A.(Ed.), ABACUS, University of Strathclyde, Glasgow, No. 32, May 1979.

Ryan, D.L., Computer Aided Architectural Graphics, Marcel Dekker, New York, 1983.

Sandhu, M., 'Automation and the Silicon Chip', Bibliography, GLC, Jan. 1979.

Scott, J.E., Introduction to Interactive Computer Graphics, Wiley, New York, 1982.

Scottish Special Housing Association, Computer Aided Architectural Design, CAAD Interface: User Manual, SSHA, Edinburgh, 197-.

Scrivener, S. 'How to handle an image',Design Studies,Vol.14, No.1, Jan.1983. Butterworth Scientific Ltd., in cooperation with the Design Research Society.

Shelly, J., Microfuture, Pitman, London, 1981.

Smith, H.T.(Ed.), Human Interaction with Computers, Academic Press, London, 1980.

Stout, T., 'SUE', BoCAAD, Bridges, A.(Ed.), ABACUS, University of Strathclyde, Glasgow, No. 35, Feb. 1980.

Sussock, H., 'GOAL', ABACUS Occassional Paper No. 62, ABACUS, University of Strathclyde, August 1982.

Sussock, H., Ross, J., 'BIBLE', ABACUS User Manual No. M11, ABACUS, University of Strathclyde, December 1982.

Sutherland, I.E., 'SKETCHPAD: a man machine graphical communication system', AFIS Conference Proceedings, Detroit, 1963.

Sutherland,I.E, Sproull,R.F, Schumaker, R.A, 'A Characterisation of Ten Hidden-Surface Algorithms', Computing Surveys, The Survey and Tutorial Journal of the ACM, Vol.6, No.1, March 1974. Association for Computing Machinery.

Svennar, E., Boligens planlosning (The Dwelling Plan)', Handbook No.25, Norwegian Building Research Institute, Oslo, 1975.

Tektronix, 4631 Hard Copy Unit: Users Instruction Manual, Tektronix Inc, Oregon, USA, Set. 1974.

Tektronix, 4952 Joystick Option 2: Instruction Manual, Tektronix Inc, Oregon, USA, Oct. 1979.

Tektronix, 4050 Series Graphic System Reference Manual, Tektronix Inc, Oregon, USA, Nov. 1979.

Tektronix, Introduction to Programming in BASIC Manual, Tektronix Inc, Oregon, USA, Nov. 1979.

Tektronix, Introduction to Graphic Programming in BASIC Manual, Tektronix Inc, Oregon, USA, Apr. 1980.

Tektronix, 4907 File Manager Operators Manual, Tektronix Inc, Oregon, USA, May 1980.

Tektronix, 4663 Interactive Digital Plotter Operators Manual, Tektronix Inc, Oregon, USA, Oct. 1980.

Tektronix, Tektronix Inc., Plotter, Pen, Media and Ink, Users Guide, Tektronix Inc, Oregon, USA, Feb. 1981.

Tektronix, 4054 Options 30 and 31 Dynamic Graphics Operators Manual, Tektronix Inc, Oregon, USA, October 1981.

Tektronix, 4051 Option 1, Data Communication Interface Operators Manual, Tektronix Inc, Oregon, USA, Nov. 1981.

Tektronix, 4050 Series R06 Editor: Operators Manual, Tektronix Inc, Oregon, USA, Nov. 1981.

Tektronix, 4663 Interactive Digital Plotter: Programmers Reference Manual, Tektronix Inc, Oregon, USA, Jan.1982.

Tektronix, 4050 Series Graphic System Operators Manual, Tektronix Inc, Oregon, USA, July 1982

Tektronix, 4052A/4054A BASIC and GPIB Enhancements: Programmers Reference Manual, Tektronix Inc, Oregon, USA, July 1982.

Tektronix, Tek Products 1984, Tektronix Inc., Oregon, USA, 1984.

Tektronix, Tekniques: The IDG Applications Newsletter, Tektronix Inc., Oregon, USA, Vols. 5-8.

The Marketing Consultancy Group, Furniture Industries Research Association, A Study of Furniture and Appliances Used in the Home, 1979, Prepared by MAS Survey Research Ltd.

Thiberg, A., Room Layouts: Report R:41 1970, Swedish Research Institute, Stockholm, 1970.

Th'ng, R., Davies, M., ABACUS Occasional Paper 23, SPACES 2 A Program Package for Producing Sketch Layouts, University of Strathclyde, 1972.

Trout, R.G., 'Looking Ahead', A Plain Buiders Guide to Computing - Occasional Paper No.14, Institute of Building, Berks., England, 1977.

Walker, B.S., et al., Interactive Computer Graphics, Crane Russak, New York, 1976.

Wheatley, D.M., The Algorithm Writers Guide, Longman, London, 1972.

Whitaker, D., OR on the Micro, Wiley, Chichester, 1983.

Willey, D.S., 'Approaches to Computer Aided Architectural Sketch Design', Computer Aided Design, Vol. 8, No. 3, July 1976.

Williams, R., 'On the Application of Relational Data Structures in Computer Graphics', Information Processing, IFIP Congress, Stockholm, 1974.

Williams, R., 'A Survey of Data Structures for Computer Graphics Systems', Data Structures, Computer Graphics, and Pattern Recognition, Academic Press, London, 1977.

Williams, R., et al., 'Data Structures in Computer Graphics', Data Structures, Computer Graphics, and Pattern Recognition, Academic Press, London, 1977.

Wilson, J.R.,Cooper, S.E., and Ward, J.S.A., Manual of Domestic Activity Space Requirements, compliled by Institute for Consumer Ergonomics for DOE, Loughborough University, unpulished.

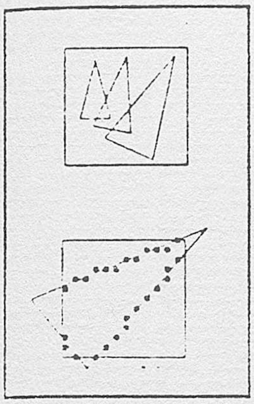
R.2.0 ABBREVIATIONS

The following abbreviations have been used in the text, but mainly in the references and bibliography.

ABACUS	Architecture and Building Aids Unit Strathclyde, Department of Architecture and Building Science, University of Strathcylde, Glasgow.
AJ	Architects Journal.
AP	Association Penalty.
ARC	Architecture students.
ARC 3	3rd. year Architecture students.
ARC 2	2nd. year Architecture students.
ARIAS	Associate of the Royal Institute of Architects in Scotland.

ARIBA	Associate of the Royal Institute of British Architects.
ASSA	Association of Scottish Schools of Architecture.
BSI	British Standards Institution.
BoCAAD	Bulletin of Computer Aided Architectural Design, published by ABACUS.
CAAD	Computer Aided Architectural Design.
CAD	Computer Aided Design.
CICA	Construction Industry Computing Association.
COM	Computer Output Microfilm.
CR	Carriage Return key.
CNAA	Council for National Academic Awards.
CSU	Computer Services Unit, RGIT.
DEC	Digital Equipment Corporation Incorporated, Marlboro, Massachusetts.
DOC	Design Office Consortium.
DOE	Department of the Environment (UK).
EdCAAD	Edinburgh Computer Aided Architectural Design, Department of Architecture, University of Edinburgh.
ER	Efficiency Ratio.
FIRA	Furniture Industries Research Association.
GLC	Greater London Council.
GPIB	General Purpose Interface Bus.
HDD	Housing Developement Directorate (UK).
HE	Home Economics students.
HE 3	3rd. year Home Economics students.
HE 2	2nd. year Home Economics students.

HMSO	Her Majesty's Stationery Office.
IFIP	International Federation for Information Processing.
LA	Local Authority.
NBI	Norgesbyggforskingsinstitutt (Norwegian Building Research Institute).
OP	Overlap Penalty.
PSA	Property Services Agency.
RAM	Random Access Memory.
RGIT	Robert Gordons Institute of Technology, Aberdeen.
SERC	Science and Engineering Research Council.
SP	Space Efficiency Penalty.
SSSA	Scott Sutherland School of Architecture, RGIT.
VDU	Visual Display Unit.
VU	Volume Utilisation Penalty.



APPENDIX A

CONTENTS

Occasional Paper No 04
"Feedback Aids to Geometry Data Input"

B G Hammond (March 1983)

The author's interest in the application of the computer to architectural design was roused in his undergraduate days at Scott Sutherland School of Architecture, Aberdeen, from which he graduated with B.Sc(Hons) in 1981. He is currently undertaking an SERC funded Research Degree at that same school under the direction of Dr. L.W.W. Laing investigating techniques for the input of geometric data by the user of CAAD systems.

Feedback aids to Geometry Data Input

This paper examines methodologies for the input of geometric data within CAAD applications. It examines the man/machine interface and postulates how new technology might affect that structure. Potential algorithms are presented which outline a new data input technique.

1 Introduction

Computer Aided Architectural Design (CAAD) has been a topic of conversation within the architectural profession for some considerable length of time [1]. However, considering the undoubted computational power of the modern computer both as a mainframe and as a micro, it is something of a surprise that in that time the utilisation of the computer within architectural practice has been minimal in comparison with its utilisation in other professions associated with building design, such as quantity surveying and structural engineering [2].

Writers have postulated a variety of considerations inhibiting the architect's utilisation of CAAD technology ranging from cost implications to a feeling of loss of control over the design process [3]. Some of these considerations, such as those related to cost, have become less valid as the years have passed; as is evidenced by the growth in micro-based systems directed primarily to the 'business' of architecture [4]. However the breakthrough of CAAD into architectural practice predicted by early writers remains as elusive as ever [5].

2 Problem Identification

Quantity surveying and structural engineering, as disciplines, essentially are well defined processes dealing with data and its manipulation; such work is the bread and butter of computing. Architecture is quite different. Firstly the process of design is ill-defined and secondly 'number crunching' is something alien to the architect - except when working out his accounts!

These factors continue to inhibit the architect's use of computer technology. Current CAAD applications have overcome the first of these hurdles by concentrating on those aspects of design which are well defined such as energy simulation or 2D representations of 3D data, leaving design synthesis to the architect himself. Because of this, the second hurdle, the use of numeric data as a medium of representing the geometry of a building within the computer has been accepted as an imposition dictated by the machine.

The creation of a data structure (plans, sections specifications etc.) encoding a building model - which can be considered to be an abstract representation of a buildings geometric and physical properties - can be deemed to be the cornerstone of an architects work. Similarly, for CAAD, the building model AND its creation ought to be central to any applications software. Data structures already exist which are more than adequate for encoding a building geometry [6]: however, the creation of the numeric form of that structure is more problematical. Various methodologies and technologies have been utilised, ranging from the light pen to the digitising tablet. To the designer, all have suffered from the same problems: he must either laboriously align each building element at some late stage in the design process,so as to ensure geometric accuracy,or he must suffer a constant interruption to his flow of thought as he answers 'obvious' questions posed by the computer. In either case the result is the same; the time spent on data entry and checking may be so great as to outweigh the value of the resultant output [7]. Until this problem is overcome CAAD will probably remain a limited specialism outwith the mainstream of architectural practice.

3 The Interface

This problem primarily concerns the design of the man machine interface Newman [8] has identified four componants that are intrinsic to the design of the interactive user interface:-

1. A user's model
2. A command language
3. Feedback
4. Information display

These componants are illustrated graphically in figure 1 over.

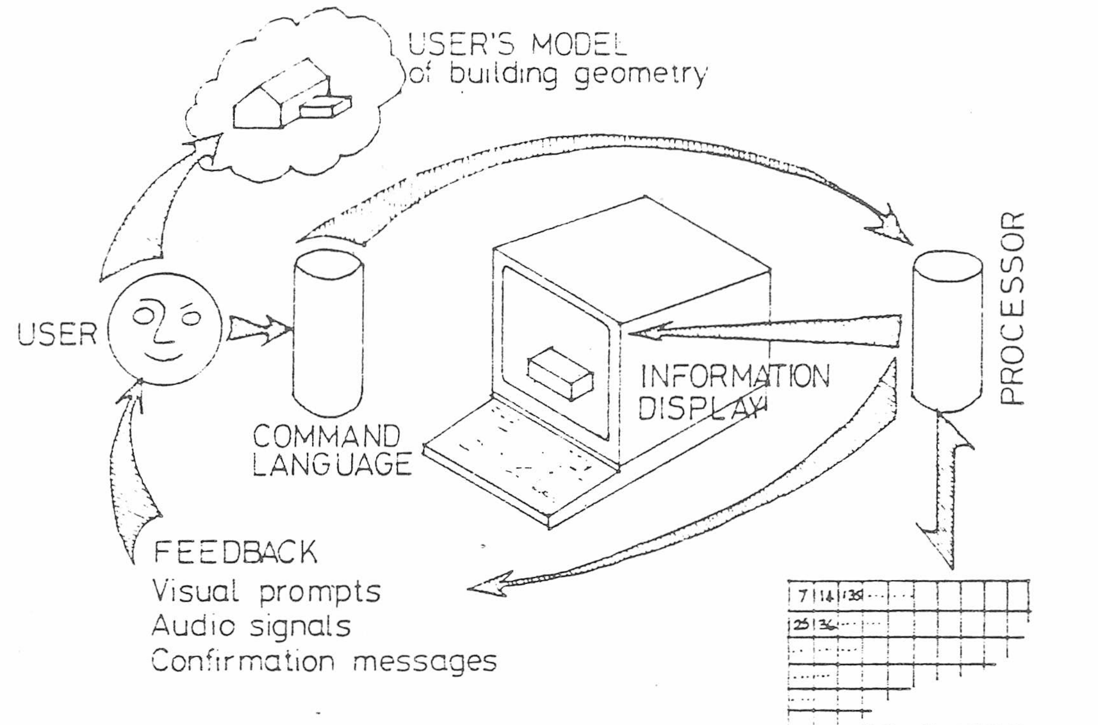


FIGURE 1 — Graphical representation of componants involved in man / machine interaction

From the diagram, it is evident that of the four componants mentioned, the two that play the greatest part in successful user interaction with the machine are those componants common to both man and machine - namely, the command language and feedback.

Command languages have received widespread attention and investigation. Feedback on the other hand is comparitively neglected. Feedback by its nature must be rapid, and consequentially it is usually treated as a by-the-way to the information display.

This situation is further complicated by the device dependancy of both command languages and of feedback [9], for example, intensity modulation of a displayed image is virtually impossible utilising storage tube technology. However this last fact would seem to indicate that when alternative display technology becomes available, then exploitation of that technology to the full would have implications on the perceived quality of the man/machine interface.

4 Technological Innovation

A comparitively recent addition to the architects computer arsenal comes in the form of the Tektronix 4054A Graphics Terminal, complete with Dynamic Graphics (Option 30). As a member of the Tektronix 4050's series family; it is downward compatible with previous 4050's machines enabling existing software, such as GABLE 80, to be utilised successfully.

As a 64k micro-computer, the problems of command language are simplified; software of any complexity has to be assembled as a series of inter-related subroutines - if only for reasons of memory storage. The natural consequence of this is that the system designer utilises the user definable keys as a method of obtaining direct program control, leaving the user with a limited but easilly comprehensible and relatively flexible choice of operations within any one program segment.

This terminal also offers interesting possibilities in terms of feedback. Due to its dynamic (vector drawn refresh graphics) capabilities, the enhancement of feedback to the user becomes a real possibility. The function of feedback can be described as the computer informing the user as to where he is, what he can do here, what he can do next, and how he does it[10]. Using dynamic graphics, messages displayed in refresh mode can be displayed and altered as necessary, answering all the above questions in a manner which would be impracticable on a storage tube.

Furthermore, there is the possibility of utilising this dynamic capability to obtain a significant leap in the quality of feedback for the architectural user, particularly in regard to the generation of geometrical input data, whilst obtaining an integration of this feedback with information display functions that could not be contemplated on the more conventional, at least for architectural purposes, storage tube display.

5 Geometry Input

Geometry input for machine implementation can take one of two approaches; the Point Set method or the Boundary description method. See diagrams 2 and 3 for graphical representations of both methodologies.

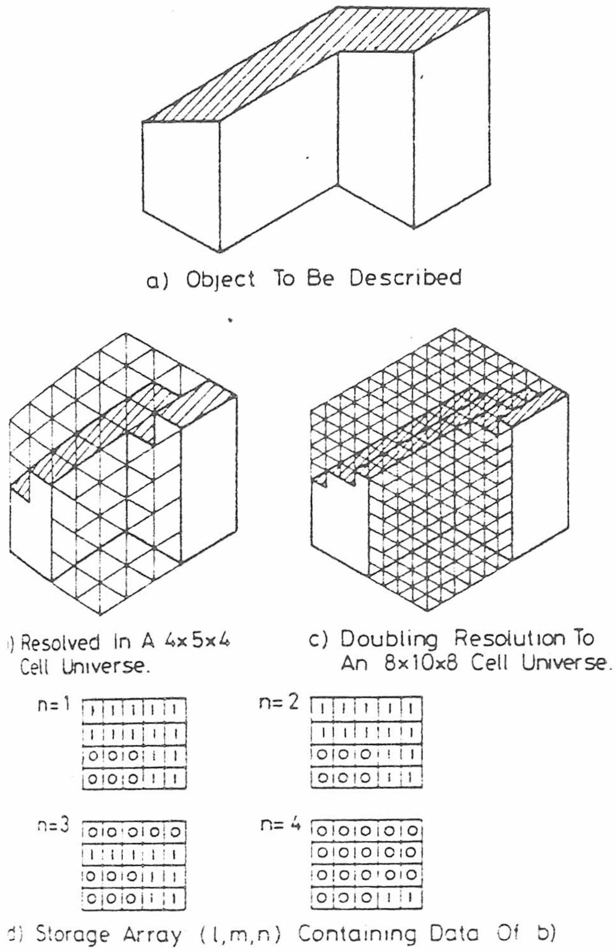
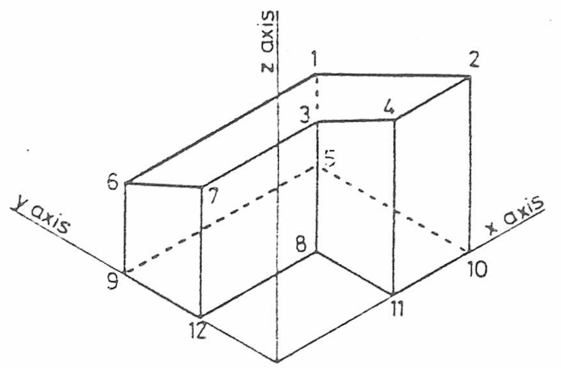


Diagram 2 Point Set Method Of Encoding Shape Descriptions



a) Object To Be Described In A Cartesian Universe

Surface i.d	General Equation	Values
1	$l_1 x + m_1 y + n_1 z = c_1$	l_1, m_1, n_1, c_1
2	$l_2 x + m_2 y + n_2 z = c_2$	l_2, m_2, n_2, c_2
3	$l_3 x + m_3 y + n_3 z = c_3$	l_3, m_3, n_3, c_3
	"	
	"	
	"	
8	$l_8 x + m_8 y + n_8 z = c_8$	l_8, m_8, n_8, c_8

b) Data Storage Schema For Planar Boundary Description

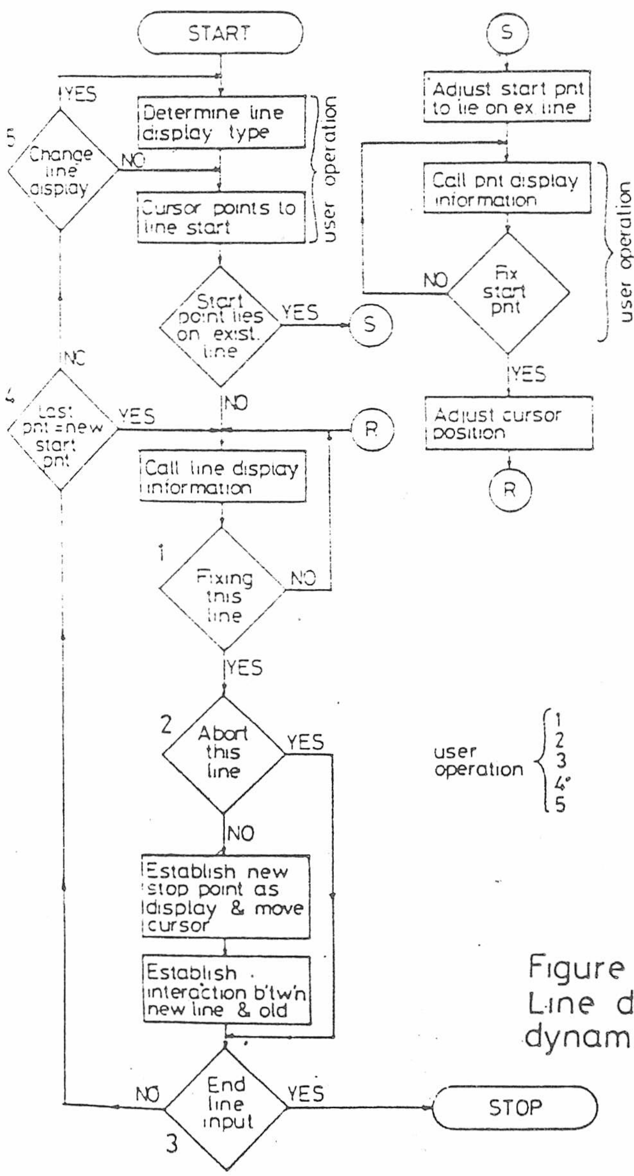
Diagram 3 Boundary Description Method Of Encoding Shape

Due to problems associated primarily with resolution, the point set method is virtually unused for architectural purposes. Within the spectrum of boundary description methodologies two broad approaches can be perceived for data entry via a screen device. Input can be directly through the keyboard, as a series of co ordinates (polar or cartesian relative or absolute) or alternatively, a cross-hair cursor can be utilised to indicate directly on the screen, start/stop points. Both methodologies have advantages and disadvantages. Using the cross-hair cursor is generally faster than co-ordinate input but is considerably less accurate. Co ordinate input. although more accurate than the cross-hair is slower and more tedious resulting in a greater risk of errors and is undoubtedly inhibiting to professionals whose traditional communication device is the drawing board.

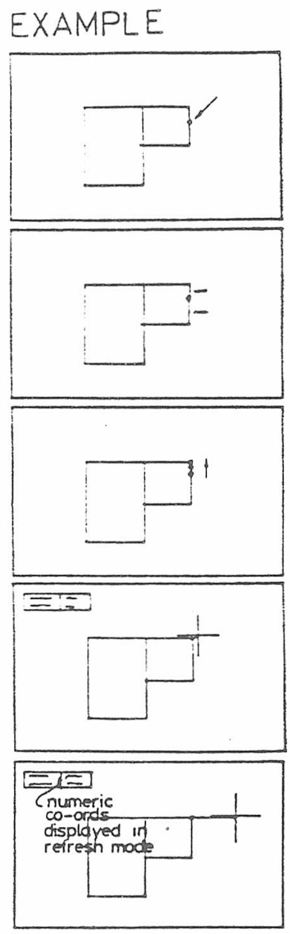
Considerable work has been done in attempts to overcome some of the disadvantages inherent in each system. The overall speed of both systems has been increased by utilising chaining algorithms to reduce the data entry requirements. The use of a construction grid has been popular as a method of increasing cursor accuracy, however. it can result in a loss in the flexibility of the geometrical form allowed.

Dynamic graphics when combined with a feedback technique, allows the effective combination of both methodologies. A cross-hair cursor can be utilised in conjunction with dynamic displays of co ordinate information such that while the user is creating the geometric form of a building element (wall, partition, etc.) the computer renders feedback as to what it currently believes the user intends; the user then informs the computer when its assessment is correct and upon this confirmation the computer updates its building model to the latest data it has fed-back to the user. The user has gained numeric accuracy without the loss of either flexibility or speed.

Below in figure 4 an algorithm for utilising such a dynamic feedback technique is postulated.



EXAMPLE



User points to start point of line, also indicating the type of display required

Computer decides that the start point indicated is within a tolerance value of an existing line and adjusts the co-ords. to lie on that line. It then displays information on that point position relative to the line.

User dynamically repositions the start point of the line obtaining feedback concerning that points position relative to the identified line. User confirms when displayed data is correct. Computer re-adjusts start point of line to comply with displayed information.

Computer sets end point of line to equal start point and creates a rubber band joining start and stop points. It then displays information concerning the new line.

User confirms when the displayed information is correct and the computer adjusts line end point to comply with the displayed information. Whilst confirming the user has the option to inform the computer whether he wishes to continue inputting lines by chaining, abort the most recent line, or stop inputting lines.

The last operation is when the computer updates the existing data base to take account of the new data. At this point tests for line intersection etc. may be applied.

Figure 4
Line data entry - with dynamic feedback technique

Clearly this algorithm does not cover every eventuality. However, the approach outlined is no less valid for that - precise relative geometry could be entered for any one line with no more than two operations. Furthermore, additional labelling can easily be attached to the line data to give the computer further information as to what the line represents to the user. Such additional information might encompass a coding representing line thickness, or type of construction being represented etc.

It must also be noted that the introduction of such a large amount of feedback, whilst not altering or ammending the four basic componants of the man/machine interface, radically alters their relative importance. This is shown diagramatically below in figure 5.

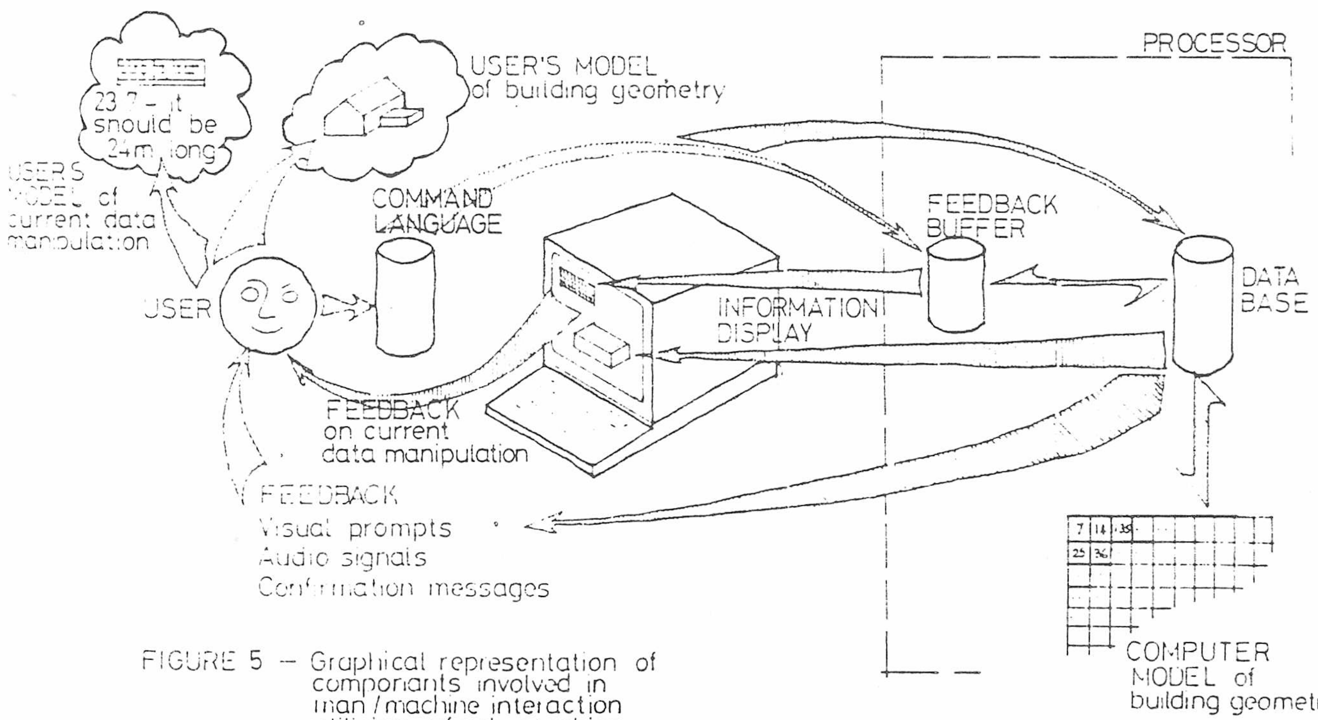


FIGURE 5 – Graphical representation of componants involved in man/machine interaction utilising refresh graphics capabilities

As can be seen, when the user is given feedback on geometric data as he is entering it into the system the computer preprocesses that data during user operation until the data is satisfactory to the user, prior to the despatching of 'correct' data to the data base.

6 Geometry Primatives

Despite the advantages accruing from the use of 'continuous' feedback whilst creating geometric data, this technique alone would be insufficient to overcome the inhibitions of the naive architectural user about utilising a computer.

Part of this inhibition lies with the fact that, even with a feedback technique, the user would be required to operate on the geometry model at the crude level of boundary description - being able to create, delete or modify building geometry characteristics a line at a time.

However, the architectural designer customarily designs with spaces which by definition are multi-sided. Furthermore, many data modifications of that initial geometry model are essentially modifications of its spatial characteristics.

Surely within a CAAD system it would be possible to give the computer the intelligence to operate at a spatial level and comprehend that spatial modification must necessarily result in the modification of the associated boundary descriptions.

Such a system is not really feasible if the data structure encoding the building geometry is composed of data pertaining solely to the boundaries of each space; the computer would require relational data concerning the spaces themselves. If the computer requires this sort of information to solve that type of problem then it would seem to make sense to adjust the methodology of data entry so that relational data of that nature is implicit within the data entry process.

This data entry process would require the utilisation of graphical primatives, since graphical primatives are, almost by definition ordered sets of points within the overall framework of the software data structure ; and these ordered sets, by their very ordering, must contain relational data pertaining to the lines composing the primative.

The proceedure outlined previously for one dimensional line data entry feedback, whilst useful at that level, would become of greater significance if data entry was in the form of interactions between two dimensional enclosed shapes. The potentialities for the user utilising such a data input technique are great. Not only would the user be able to manipulate more than one data item at a time but there would be the possibility of attaching target design values for such qualities as daylighting, number of air changes etc. to the shape description itself.

An essential requirement for any postulation of a 2D data entry technique whereby the user manipulates geometry primatives to encode the building description, is a clear understanding of the range of potential data manipulations required - so that the inherent flexibility in such a system is optimised.

These manipulations could be broadly classified under four headings: -

1) Whole primative transformations such as:
   a) rotation of single primative
   b) translation of single primative
   c) scaling of single primative
   d) handing of single primative
   e) X-distortion of single primative
   f) Y distortion of single primative
   g) creation deletion or repetition of a primative

2) Redefinition of primative transformations such as:
   a) rotation of single line within primative
   b) translation of a single line within primative
   c) linear distortion of single line within primative
   d) addition of an extra line to a primative
   e) deletion of a single line from a primative

3) Compositional primative transformations involving more than one primative such as;
   a) manipulation of groups of primatives in much the same way as outlined for whole primative transformations above
   b) manipulation of parts of a group of primatives in much the same way as outlined for redefinition of primative transformations above.

4) De-compositional primative transformations such as:
   a) division of one primative into 2 enclosed primatives
   b) joining of two primatives to create one
   c) creation of undivided primatives when two or more primatives intersect.
   d) area deletion

Although this list cannot be considered to be comprehensive the broad groupings outlined above give some indication of the nature of the transformations that would be required.

Transformations belonging to the first group are relatively straight forward and the data manipulations required are quite minimal Transformations in the second group would be at the core of any such data entry technique and would require a considerable amount of data handling Transformations in the third group are more complex versions of the first two groups, while transformations in the fourth group exist primarily to ensure that the data has no irregularities in the form of overlapping primatives

Clearly for this geometry data entry system to be successful the command language must be simplified to the point where the user can inform the computer of his intentions by issuing only one or two commands. However. with that success the user would be able to describe the building model accurately (through the feedback technique), rapidly (primative manipulation involves dealing with more than one line at any one time) and in a manner unlikely to disturb his design concentration.

7 Summary

1 CAAD has been a topic of conversation among architects for some considerable length of time. In that time it has had minimal impact on the practice of architecture.

2 The generation of building geometry data is central to CAAD applications software; current data entry techniques are unfriendly to the user and inhibit the more widespread application of CAAD within architectural practice.

3 Difficulties associated with geometry data entry are part of the design of the man/machine interface. Given new technology, new solutions to these problems may become possible.

4 The Tektronix 4054A Graphics Terminal represents new technology for the architectural user.

5 Boundary description is the usual method of encoding building geometry for CAAD applications. Dynamic graphics through a feedback technique offers the possibility of combining the advantages of cursor and co-ordinate screen input when describing the boundary of a building geometry.

6 An extension to the feedback technique would be to enter data in the form of 2D geometrical primatives. Extension of the command language to take account of this would enable a more user friendly geometry data entry system to be created.

8 References

1. Sutherland.I.E. "SKETCHPAD: a man machine graphical communication system AFIPS Conference Proc. Detroit (1963) pp.329 346

2. Standing Committee on Computing and Data Co-ordination. Computing and Communication in the Building Industry. Department of the Environment. Mar. 1979

3. Nigel Cross, The Automated Architect: Human and Machine roles in Design, Pion Ltd., 1977, pp. 126-127

4. Nicholas Coutts,"Low cost computing - 2.Analysing your requirments" Architects Journal, 2 Feb. 1983 No. 5 Vol. 177

5. Boyd Auger, The Architect and the Computer Pall Mall Press London 1972 p118

6. Lawson,B.R Riley,J P. 1982 Isaac, A technique for the automatic interpretation of spaces. CAD '82 Conference Proceedings

7. Setsuo Ohsuga "Towards Intelligent Interactive Systems", Methodology of Interaction. Guedj et al eds. IFIP Amsterdam 1980.p339

8. William Newman "Some Notes on User Interface Design". Methodology of Interaction. Guedj et al. eds., IFIP Amsterdam 1980, pp.325-6

9. William Newman ibid.

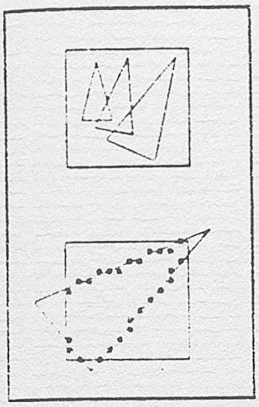
10. Nievergilt.J and Weydert J.
"Sites, Modes and Trails; Telling the user of an interactive system where he is, what he can do, and how to get places." . Methodology of Interaction Guedj et al. eds., IFIP, Amsterdam 1980 p 327

11. Wolfgank.K. Giloi,
Interactive Computer Graphics. Chap. 3 Prentice Hall. 1978.p.80

BRIAN G. HAMMOND

MAR '83

2250 words



APPENDIX B

CONTENTS

Occasional Paper No 05
"A Graphics Interface to Complement Traditional Techniques"

D M Leifer and B G Hammond
(June 1983)

The authors are currently engaged in SERC funded research degrees at the Scott Sutherland School Of Architecture in Aberdeen under the direction of Dr L.W.W.Laing. They are currently examining aspects of computer graphics with particular emphasis on the man-machine interface. Both authors have experience of working in private practice in London, and one of whom was last employed in the capacity of Job Architect.

A Graphics Interface to complement Traditional Techniques

ABSTRACT

Noting the reluctance of architects in small private practices to adopt CAAD aids, the crudity of existing graphic interfaces is identified as an inhibiting factor.

A suite of computer programmes currently under development are described which are designed to permit the input of geometric plan forms by traditional pencil and paper techniques, whilst utilising the computers processing power to edit and manipulate the data so 'captured'.

INTRODUCTION

Many reasons have been offered to explain why architects have failed to utilise computers more fully than they have done[1]. Despite the promise of the micro-chip revolution, most architects' design work is still carried out by manual methods; computer aids being viewed as a specialised adjunct applicable to a few atypical projects. This is perhaps understandable in a profession where some 85% of all registered architects work in practices employing 10 or less architectural staff[2], and where the stability of workload is sufficiently uncertain to make large capital investment precarious[3].

To most architectural practitioners who are accustomed to, and moreover enjoy using, drawing board and set-square, computers represent an alien technology requiring unfamiliar and sometimes inappropriate languages and working methods. The relative magnitude of the capital investment that computers represent to the small architectural practice requires extremely efficient and close management of the system to ensure that it is run cost-effectively. Not only is this expertise expensive, but it is at present rarely available. More insidiously, principals in small practices may feel a potential loss of overall control posed by the inherent complexity of such systems. Such fears cannot be easily allayed.

It has been argued elsewhere that one of the most positve ways with which to promote CAAD would be to utilise the computers processing power to take on more of the burden of man-machine communication[4]. The machines must become more approachable by the non-computer literate user, leaving him free to devote his energies to designing rather than on communicating with, and operating the system.

THE ARCHITECTS ROLE

The architects task may be construed as the conception and communication of a hypothetical building model. Infact, the design process is one in which the designer compiles ever increasingly accurate data pertaining to the hypothetical model until such time as it is sufficiently complete and consistent to allow the client to 'experience' it and the builder to build it. (This pragmatic description does not belittle the implicit importance attached to the role of the architect as an aesthete). The common feature underlying all of the various architect generated data sets is the building models geometry; whilst the material specifications state 'what' it is and the performance specifications state what 'it'is to do, the drawings state where all of this other information is to apply, and how it is interrelated.

The corollary to this process is constant appraisal to ensure that the sub-systems do what they are supposed to, be it the adequacy of the structural system or that the project may be built within budget.

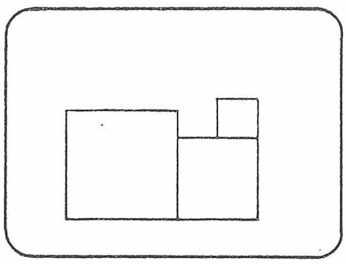
DATA MANIPULATION

The magnitude and complexity of the data generated for any one building project, not least of which is the project drawings, can only be'guesstimated' in advance. This puts extraordinary demands on any computer system which would handle such a large and various data-set. Although some data-basing systems have been evolved and utilised by the profession[5], none have been sufficiently effective for machine implementation. Perhaps one of the most significant developments in data-manipulating techniques has been the development of logical programming languages such as PROLOG[6] which offer a powerful means for interogating large data-structures. The application of these languages to graphic data is currently under investigation elsewhere[7].

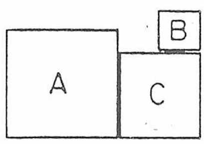
COMPUTER GRAPHICS

Within the overall CAAD context, one area of concern is the machines data-aquisition rather than its data-manipulation. This is particularly relevent to drawn information, since it is at this level that architects 'experience' computer systems. Current systems require inhibiting draughting conventions and crude levels of communication.

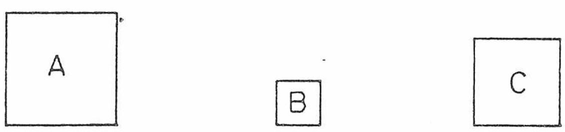
Drawings represent simultaneously many different levels of signifcance to the the user[8]. Each level is 'distilled' from it's contextural relationship to other parts of the drawing. Thus the fundemental problem with computer graphics is the radical differnce between mans and machines perception of drawings. Whilst the human can relate parts of a 'picture' to the whole, the machine is simply a collection of registers which can only cope with one data item at a time. The only 'context' the machine has for construing a drawing is the artificial one implicit in the structure of the data-base. To illustrate this, consider the example below.



a) Picture to be created



b) Association of Graphic Primitives



c) Primitives

Primitive	A	B	C
scale factor	Sa	Sb	Sc
reference point x	xa	xb	xc
y	ya	yb	yc

d) Input Data

move pen to	(xa,ya)	(xb,yb)	(xc,yc)
draw line to	(xa+Sa,ya)	(xb+Sb,yb)	(xc+Sc,yc)
draw line to	(xa+Sa,ya+Sa)	(xb+Sb,yb+Sb)	(xc+Sc,yc+Sc)
draw line to	(xa,ya+Sa)	(xb,yb+Sb)	(xc,yc+Sc)
draw line to	(xa,ya)	(xb,yb)	(xc,yc)

e) Machine Draughting Commands

Diagram 1 Example Of A Data Structure

Consider a simple CAAD graphics input system where drawings are created as combinations of squares. The user wishes to input a picture shown in 1a. The picture comprises of three squares A,B and C. Ignoring the interelationship of the squares with each other, the machine must be given sufficient information about each geometric primitive not only to carry out the calculations that might be required in the subsequent applications programme, but simply to draw them on the screen. In the case illustrated the necessary input data includes the coordinate position of a point of reference for each square (here taken as the lower left-hand corner), and a scaling factor for each square. (This input data is shown in the table, Diagram 1d.). In order to draw these squares, the machine must convert this input data such that the drawing routines may be enacted. (These drawing routines are shown in Diagram 1e.)

It will be seen that in such a system it is necessary to refer to the lower levels of the data-structure if the user wants to address any particular line. This level is of course different to the 'vocabulary' used to create the picture in the first place. This cumbersome method of dealing with drawings does not bear comparison to the ease of paper and pencil techniques.

THE PROPOSED GRAPHICS INTERFACE

To overcome the disincentive outlined above, an opposite route may be considered. Accepting the premise that architects generally design in the initial stages by 'toying' with freehand sketches, it is legitimate to place the onus on the computer to derive higher levels of significance (ie. recognition of graphic primitives), from the lowest level of input data (ie. the continuous digitisation of architects sketches); in effect to deduce the data-structure from the act of drawing.

The system being developed is summarised in Diagram 2. Architectural sketches are digitised on a graphics tablet (Tektronix 4954) which is connected via a Tektronix 4010 interface to a micro-computer with graphics display screen (Tektronix 4054).

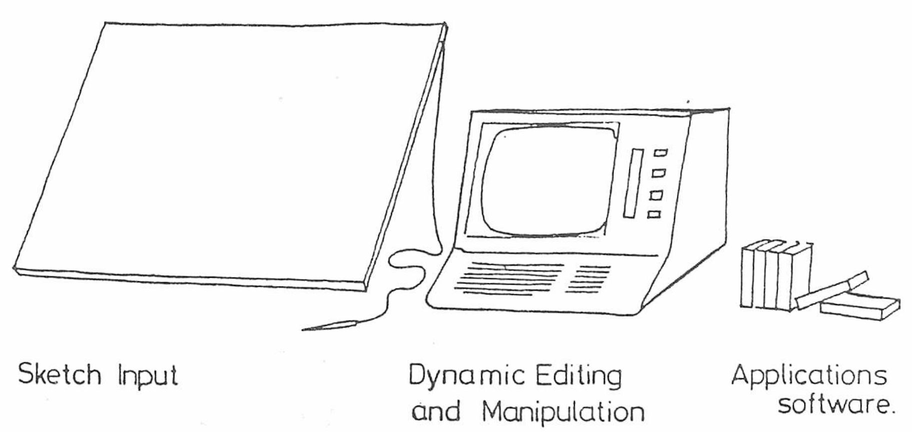


Diagram 2 Configuration Of Proposed Interface

SYSTEM DESCRIPTION

The proposed system operates in five sections.

1. As described the designer sketches his intentions on the digitising tablet, an example of which is shown in Diagram 3. The rate of data capture by the machine will depend upon the speed at which the user draws and the cycle time of the digitiser.

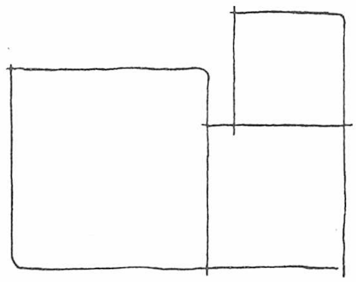


Diagram 3 Sketch As Drawn On Tablet

The drawing is echoed on the screen of the graphics terminal to confirm to the user that the drawing has been captured. The echo on the screen resulting from the sketch shown in Diagram 3, and the format that the digitised data takes is shown in Diagram 4.

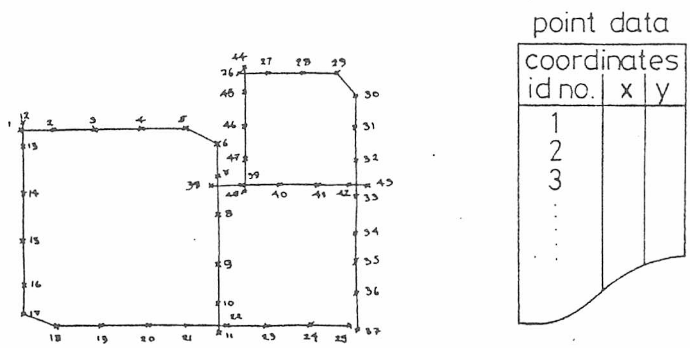
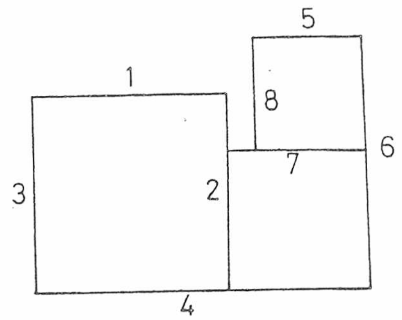


Diagram 4 Digitised Information Echoed On Screen

2. The digitised tablet data is processed, and the data is sorted into discrete lines by comparing the gradient of each successive line element to the line of closest fit through the preceeding data points. If the deviation is greater than a user defined tolerance a new line is deemed to have been encountered.

Moreover the 'image' is enhanced: Not only are freehand lines straightened, but overlapping lines are removed, clipped corners reconstructed, and almost touching lines made to touch. The infomation about the lines composing the drawing are stored in an array for further processing, and the enhanced image displayed on the screen as shown in Diagram 5.



line data

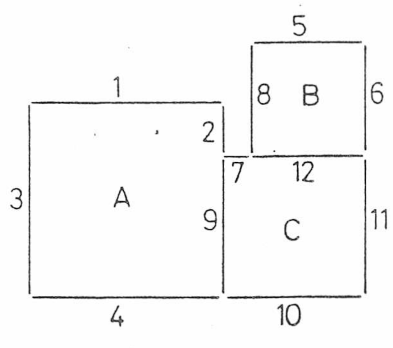
line i.d. no.	start point x1	y1	stop point x2	y2
1				
2				
3				
:				
7				
8				

Diagram 5 Computer Interpretation Of Lines

3. The user may wish to edit the interpretation the computer has generated. Using the dynamic graphic facilities of the terminal, he may delete or add lines by means of the cross-hair cursor, or indeed add information via the tablet.

Manipulation via the terminal has the advantage that the user can, if he wishes, make the drawing orthogonal, or place the lines in a range of sectors (ie. 0,15,30,45,60,75 or 90 degrees). With this editing process via the terminal, the dynamic feedback technique mentioned earlier is available to ensure the accurate placing of lines on the screen.

4. The line data generated at the completion of the above editing process is then dissasociated into discrete single line sections. Thus a line disected by another is split into two individual lines. The result of this dissasociation in shown in Diagram 6.



graphic primitive data

shape ref.	no. of edges	line i.d's				
A	5	1	2	3	4	9
B	4	5	6	8	12	
C	5	7	9	10	11	12

Diagram 6 Line Disassociation And Primitive Identification

By envoking a search algorithm, the perimeter of each enclosed space is traced and the line identifiers for the boundaries stored in an array. The system has thus abstracted geometric primitives from the data in a form which may be used directly by the advanced graphics capabilities of the graphics terminal.

Utilising the terminals capabilities, the user can manipulate complete primitives. He may repeat, move, mirror, rotate, and scale. An indication of these facilities is demonstrated in Diagram 7.

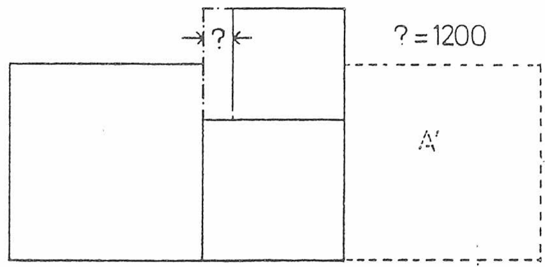


Diagram 7 Graphic Manipulation Option Utilising Dynamic Graphics Offering Instant Feedback And Rubber Banding

5. The final step for the system is to reformat the graphic data into structures suitable for input into other applications programs. Since most applications software deals with polyhedral geometries the data-structure produced by the above system contains the necessary information for transmutation to other forms.

SUMMARY

The development project described above is intended to ease the communication of drawings between the architectural user and the machine in the following ways :

1. By placing the onus of interpretation on the machine, the designer can devote his time more fully to the task of designing.
2. By automating the interpretation system the machine is made accessible to the non-computer literate user.
3. Such a system supplements the existing drawing board techniques generally used by the architectural profession at present, and can make the new technology less obtrusive.
4. Such a system frees the designer from the constraints of draughting conventions common with many existing graphics handling applications programmes.
5. Appraisal programs may be instigated much earlier in the design process, which can maximise the benefit of the advice thus rendered. Moreover there is less overhead involved in the time consuming task of 'digitising' drawings done in advance by manual methods.

REFERENCES.

1. Standing Committee On Computing And Data Coordination
"Computing And Communication In The Building Industry"
D.o.E March 1979

2. CICA
"Fact Sheet On The Use Of Computers In The Construction Industry"
CICA Bulletin Mar. 1982

3. Bijl.A, Stone.D, Rosenthal.D EdCAAD, University Of Edinburgh
"Integrated CAAD Systems"
Final Report To D.o.E On Funded Research Project DRG 470/12 Mar.79

4. Leifer.D Scott Sutherland School Of Architecture, Aberdeen
"Man Before The Machine : An Issue In CAAD"
BoCAAD University Of Strathclyde, Glasgow Vol 43-46 Feb 1983

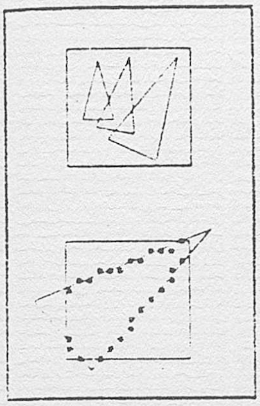
5. Ray-Jones.A.R, Clegg.D SfB Agency
"CI/SfB Construction Indexing Manual"
RIBA Publications 1976

6. Clocksin.W.F., Mellish.C.S.
"Programming In PROLOG"
Springer-Verlag 1981

7. Work currently being undertaken at EdCAAD, Edinburgh University.

8. Willey.D.S University Of Liverpool
"Approaches To Computer-Aided Architectural Sketch Design"
Computer Aided Design Vol 8 No.3 Jul 1976

Brian Hammond B.Sc
Dave Leifer B.Sc, B.Arch Apl 83.



APPENDIX C

CONTENTS

Occasional Paper No 10

"ENIGMA: An Enhanced Interpretive Graphics Module for Architects"

Dr L W W Laing, D M Leifer and
B G Hammond (July 1983)

CAD 84: BRIGHTON METROPOLE, SUSSEX

PAPER PRESENTED BY:

L.W.W. Laing BA, MSc, PhD, ARIAS
D.M. Leifer BSc, BArch, ARIBA
B.G. Hammond BSc (Hons)

ABSTRACT

The reluctance of architects to adopt the computer as a potential design aid is explained to be, in part, a consequence of the difficulties which occur in conveying visually conceived ideas to a binary logic machine. Typing-in numbers is an inhibition to free expression when compared with traditional drawing board techniques.

This paper therefore describes software currently being developed at the Scott Sutherland School of Architecture in Aberdeen and which is designed to allow the input of geometric plan forms by traditional pencil and paper techniques whilst utilising the computer's processing power to interpret, edit and manipulate the design data so "captured". The geometry data is structured to allow re-formatting as necessary to interface with applications software currently used by undergraduate students within the school.

Particular reference is included to the potential offered by refresh graphics display for dynamic manipulation of the geometry data during sketch design.

KEYWORDS

CAAD: GRAPHICS: DATA INPUT: DYNAMIC MANIPULATION.

The paper represents RESEARCH IN PROGRESS.

ENIGMA : an ENhanced Interpretive Graphics Module for Architects

Co-authors: L.W.W. LAING BA, MSc, PhD, ARIAS
D.M. LEIFER BSc, B Arch, ARIBA
B.G. HAMMOND BSc (Hons)

Scott Sutherland School of Architecture
Robert Gordon's Institute of Technology
ABERDEEN, Scotland.

Current CAAD research at the Scott Sutherland School is supported by SERC funded research studentships, represented here by Mr Leifer and Mr Hammond, under the direction of their co-author Dr Lamond Laing. All three have a cumulative total of over 30 years experience as architects in practice as a background to their CAAD research activity.

INTRODUCTION

Despite the enthusiasm which accompanied the introduction of computing into the undergraduate architecture curriculum in many schools since the 1960's, evidence of its adoption into architectural practice is still, by comparison with other building related professions, very sparse (1,2). In a profession where 85% of those registered work in quite small practices (employing 10 or less staff) and where an inconsistent work load makes high capital investment in equipment risky, it is perhaps not surprising to find most design work is still carried out using manual (drawing board) techniques. Indeed, those few practices which have acquired computer aids are almost invariably found to be involved with large-scale developments where the opportunity to use repetative design elements has encouraged the use of computerised draughting systems, Using the computer in this way, ie as a mere data storage and retrieval device for handling building elements, is a poor substitute for its potential as a design aid, helping the architects to make better design decisions and, hence, providing us with a more cost-effective and efficient built environment.

Many reasons have been offered to explain this apparent disinterest among architects for CAAD (1) but, from several years of experience working with undergraduate architects (and what could be a more receptive vehicle for innovation?) there is clear evidence that the greatest source of inhibition and frustration is right at the starting post - getting the data into the machine in the first place. Quite apart from the unfamiliarity that most designers feel when confronted with an electronic drawing device, what must not be overlooked is that most architects actually enjoy drawing and are understandably reluctant to see this particularly self-satisfying part of the process being taken away from them.

The need for improved graphical (eg building geometry) input techniques is of course well known and the 1982 ACM/SIGGRAPH Workshop (3) was particularly useful in indentifying the main issues. Yet, comparatively little research has been done to develop better designer/machine interfaces. Historically, the development of computer programs for CAAD concentrated on designing the best models for simulating building performance (visual models for perspective projections, thermal models for energy performance and models which would provide accurate cost performance predictions). These followed the apparent truism that the main potential for the computer in architectural design lay in its power as a fast appraisal tool (its potential as an automatic generator of design solutions having been found to be highly suspicious). Consequently, research effort was directed to constructing the best mathematical analogues of building performance but took the easiest (from the computer/programmers point of view) methods for getting the data into the machine, relying on the tenacity and concentrated effort of the user typing-in masses of alpha-numeric data at the terminal or, at best, using simple single-point digitising techniques on a graphics screen or digitising tablet - all of which require the user to have a fairly high computer intelligence with regard to rules of sequence, formatting etc.

One of the authors has already stated the case for utilising the computer's processing power to take on more of the burden of man-machine communication (4). While recognising that decisions must be made in choosing between "Emulation" (of an existing methodlogy) and "Innovation" (utilising new techniques) as discussed by Thomas and others at Seattle (3), we would support the view taken there that....

"As a starting point, it is essential to adopt a model which is familiar; but yet will not inhibit the future assimilation of new concepts" (5)

The designer must be free to devote his energies to designing rather than on communicating with and operating the system.

The core of this paper describes a suite of computer programs (ENIGMA) currently under development which are designed to allow the input of geometric data (plan forms) by traditional pencil and paper techniques (with or without the aid of T-square and set-square), using the computer to understand and rationalise the data input with the minimum of interference to the designer's thought processes. Through this interface, other data files can be generated automatically for direct entry into the applications software where building geometry descriptions are required.

METHODS OF GEOMETRY DESCRIPTION

"The ability to describe buildings to computers is necessary before architects can use computers to perform any task related to the design and production of buildings" (6) ... and ultimately such a description involves registering the presence or absence of material in positions in space - usually 3-D space.

In general, most CAAD software uses some form of Boundary Description Method (7) in which the geometric form is delimeated by reference to nodal points, located within a Cartesian system and with instructions describing which nodes are connected to form the prescribed shape. This information must be stored in computer memory in a format which is not only compatible with the constraints of the computers hardware but also so as to allow efficient interfacing with the graphics software routines for drawing purposes. Thus, all geometry descriptions require, ultimately, to be reduced to a collection of line segments with nodes as terminal delimeters and with associated "Move" or "Draw" commands determining the presence or absence of material. This decomposition procedure is illustrated in Figure 1.

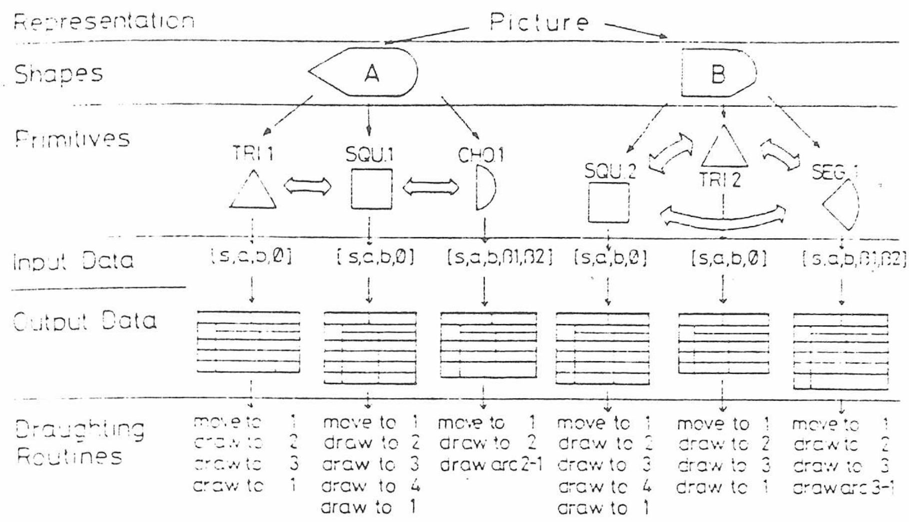


Figure 1 Decomposition of Geometry Information into Computer Graphics commands.

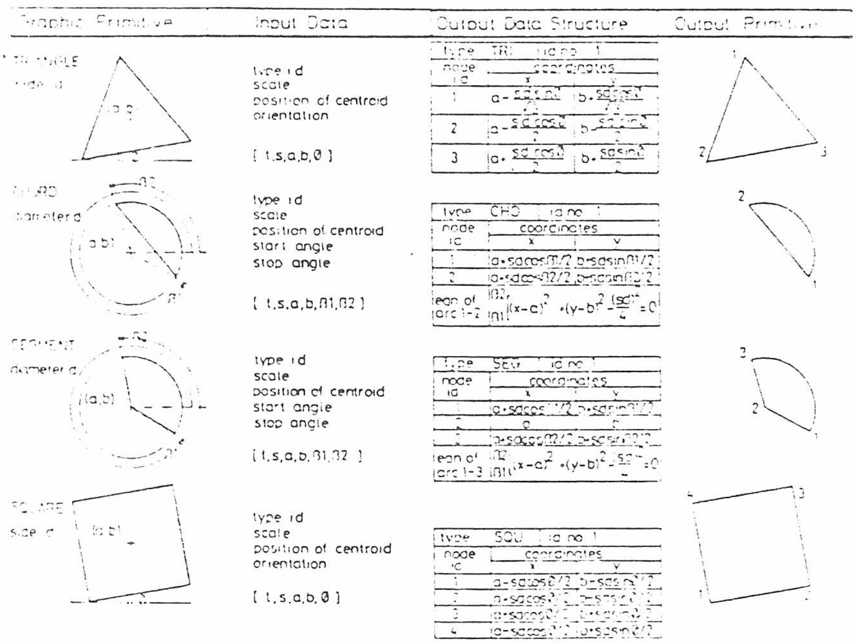


Figure 2 Shape Primitives as Basic Building Blocks for Computer Graphics

Figures 1 and 2 have been deliberately constructed to draw attention to the use of "Shape Primatives" because most applications software to date has constrained the user to develop his building geometry using such primatives - indeed most software has required the user to operate within a strictly orthogonal or rectilinear frame. Although these constraints may be, to some extent, justified by analogy with real-world building geometries, to the designer they can be an impediment at the "thinking" stage where whim and fancy prefer to remain unfettered. Furthermore, much of the frustration, experienced in inputting geometry data arises out of the extent to which small errors in the co-ordinate parameters easily occur (and these can corrupt significant parts of the final data set).

As an example, Figs 3 and 4 show typical geometry data files for the ABACUS programs 'GOAL' and 'BIBLE' (8,9) and which encodes an assemblage of 5 rectilinear boxes. The complexity of the data set increases significantly when non-rectilinear forms are accommodated (eg in 'BIBLE') and the data has to be extended to include descriptors such as number of edges, surfaces and vertices. Obviously, with larger amounts of data the risk of data error increases - often difficult to trace causing increased frustration to the user. Furthermore, co-ordinate geometry is difficult to visualise and obscures features such as proportion, scale and symmetry - concepts which are fundamental to the designer.

```
¢ TYPE G2
BLOK
5
2, 1 0 0
10, 10 0, 225 112 75, 0
3 1 0 0
310 10 0 112 225, 75 0
4 1 1 0
10 10 75 187, 112 75 0
5 1 1 0
197 10 75, 225 112 75 0
6 1 1 0
310 122, 75 112 113, 75 0
¢
```

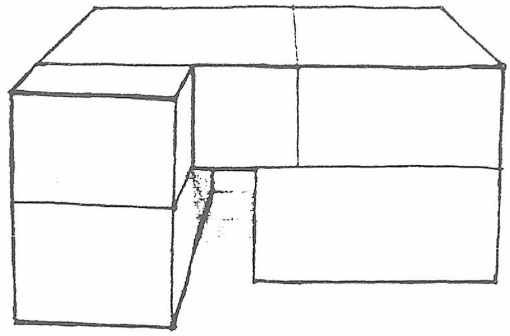


Figure 3 'GOAL*' Geometry Data File for assembly above
*(developed by ABACUS, University of Strathclyde)

```
TYPE G2B
GEN          2       1
  8   6
 10 00     10 00      0 00    235 00     10 00,     0 00
235 00    122 00      0 00     10 00    122 00      0 00

 10 00     10 00     75 00    235.00     10 00     75.00
235 00    122 00     75 00,    10 00    122 00,    75 00

4 2 1 5 6
4 3 2 6 7
4 4 3 7,8
4 1 4 8 5
4 8 7,6 5
4 1 2 3,4
GEN          3       1
  8   6
310 00,    10.00      0 00    422.00     10 00      0 00
422.00    235.00      0.00    310 00,   235.00      0 00

310 00     10 00     75.00    422 00     10 00     75 00
422 00    235 00     75 00,   310 00,   235 00,    75 00

4 2 1 5,6
4 3 2,6 7
4 4 3 7 8
4 1,4 8 5
4 8 7 6 5
4,1 2 3,4
GEN          4       1
  8   6
 10 00     10 00,    75 00    197.00     10.00     75 00
197 00    122 00     75 00     10 00    122 00     75 00

 10 00     10 00    150 00,   197 00     10 00    150 00
197 00    122 00    150 00     10 00,   122.00    150 00

4 2 1 5 6
4 3 2 6 7
4 4 3 7 8
4 1 4 8 5
4 8 7,6 5
4,1 2 3 4
GEN          5       1
  8   6
197  00      10 00     75 00    422 00     10 00,     75 00
422  00     122 00,    75 00    197 00    122 00      75 00

197  00,     10.00    150 00    422 00     10.00     150.00
422  00     122 00    150.00    197 00,   122 00     150 00

4 2 1 5,6
4 3,2 6 7
4 4 3 7 8
4 1 4 8 5
4 8 7,6 5
4 1 2 3 4
GEN          6       1
  8   6
310 00,     122 00     75 00    422.00    122 00,     75 00
422 00      235.00     75 00    310.00    235.00,     75 00

310 00      122 00    150.00    422 00,   122 00     150.00
422.00      235 00    150.00    310 00    235 00     150.00

4 2 1 5,6
4 3 2 6 7
4 4 3,7 8
4 1 4 8 5
4 8 7 6 5
4 1 2 3,4
END
```

Figure 4 'BIBLE' Data File (also by ABACUS)

These criticisms are not intended to detract from the undoubted value of programs such as those developed at ABACUS and which are outstanding among the various software systems available to architects today. The examples are only cited to illustrate the need for improved means of inputting the data set for such powerful programs.

MACHINE REFINEMENT OF RAW GRAPHIC DATA

If the designer is to be allowed the freedom to develop his geometry in a manner more analogous to the traditional freehand sketching technique then it is legitimate to expect the computer to derive higher levels of significance from the input (ie to identify, where relevant, graphic primatives) in effect to deduce the data-structure from the act of drawing. This implies a process by which a Relational Data Base (or Bases) must be derived from discrete co-ordinate data in which the relations are almost implicitly 'enigmatic' and can only be 'guessed' at by the computer. For an explanation of Relational Data Bases see, for example, Williams (10).

The process by which the ENIGMA interpretive software operates can be divided into 5 sequential operations as follows.

1. *Data Capture*

To maintain the analogy with freehand sketching, input is performed via a high resolution graphics digitising tablet (Tektronix 4954 with 4096 x 3120 addressable points) operating in continuous point mode so that co-ordinate data is generated whenever the pen is in contact with the (drawing) tablet. An ink-filled digitising pen is preferred as this reinforces the designer's cognitive awareness of his actions (rather than depending on graphical feedback through a separate display terminal).

At this stage, the only keyboard input required from the user is a declaration of the scale he is working to and the acceptable grid tolerance.

eg Scale 1:50, tolerance 50 mm.

Thus, only points which are separated by more than the declared (scaled) tolerance need be recorded. The rate of capture by the machine will also depend on the correlation between the speed at which the user sketches and the cycle time of the digitiser (in practice, working to a baud rate of 120, this has proved to have no apparent disadvantages).

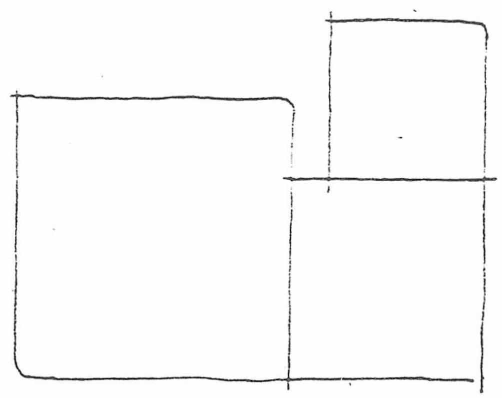


Figure 5 Sketch As Drawn On Tablet

The input sketch is echoed on the terminal display screen as confirmation to the user that his data has been captured. Figures 5 and 6 show a simple example of a freehand sketch and the resulting format that the digitised data assumes within the machine.

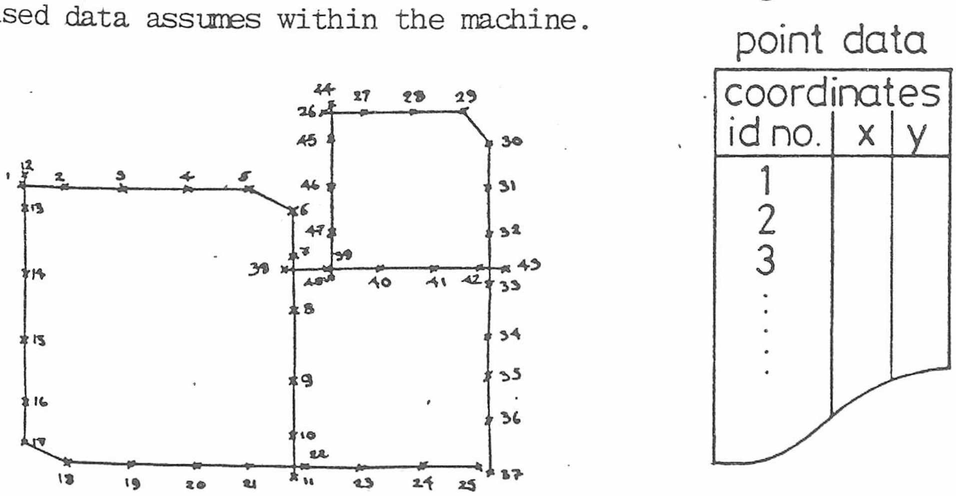


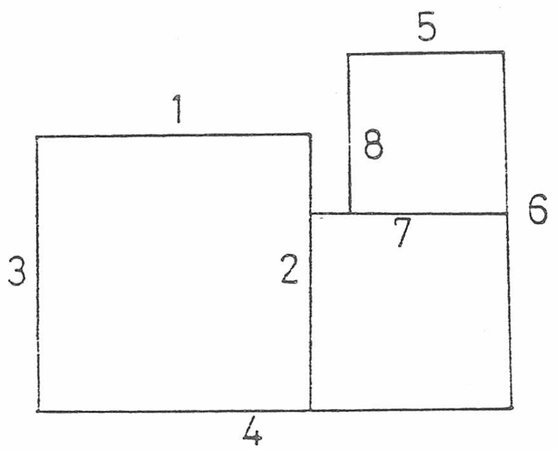
Figure 6 Digitised Information Echoed On Screen

2. *Data Interpretation & Rationalisation*

Captured data is then processed by a sequential scan which compares the gradient between each successive pair of nodes to the line of closest fit through the preceeding data points. If the deviation is greater than a defined tolerance a significant change in direction (ie a new line) is deemed to have been encountered. Thereby the data is simplified and reduced in quantity such that only the end co-ordinates of recognised lines are retained.

Moreover, the "image" is enhanced: Freehand sketched lines become straightened, overlapping lines are removed, clipped corners are reconstructed, and almost-touching lines are made to touch.

The new data is stored for further processing and the computer's rationalised "guess" of the designers intended geometry is displayed on the terminal screen as shown in Figure 7.



line data

line i.d. no.	start point x1	y1	stop point x2	y2
1				
2				
3				
:				
7				
8				

Figure 7 Computer Interpretation Of Lines

3. *First Edit*

The "guesswork" implicit during the interpretation phase may not always correspond with the designer's full intentions. Consequently, control of the data is returned to the user, this time through the display terminal, in order that he can perform routine editing operations such as deletion and addition of lines and so on.

During the edit phase, a distinction must be made between two possible hardware configurations, the software having been designed to operate under alternative modes as follows.

Where the display terminal screen is of the conventional storage type (for example Tektronix 4010 series) on-line graphical editing will require over-drawing of new data on top of old or superseded data with the screen having to be erased to display the edited data afresh.

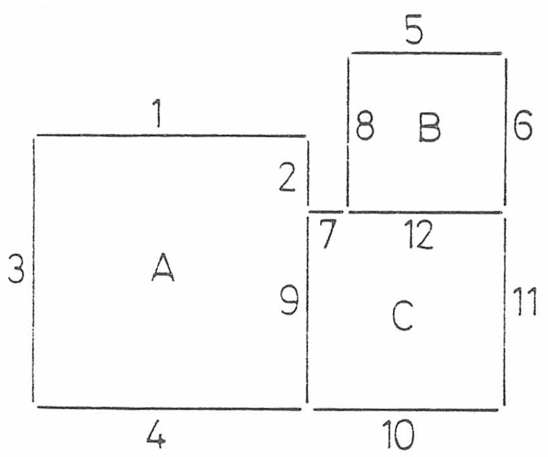
Alternatively, where the screen offers refresh display (for example Tektronix 4054 Option 2) the potential this affords in terms of dynamic graphic manipulation is enormous. Not only does "rubber-banding" allow the user to manipulate his shapes in a way which continues to model sketching techniques (lines can be seen to be being drawn) but, by including refreshed displays of the current cursor co-ordinates numerically, locational accuracy is enhanced.

Because topologically related lines are noted within the relational data base structure, it follows that for each manipulation performed on any part of the original data set, the machine can itself perform further manipulations on other parts of the data set in order to maintain integrity. This in itself reduces the amount of manipulation likely to be required in any single edit.

Additional enhanced edit facility is included such as, if required, automatic orthogonalisation of lines, addition of standard shape primatives (eg lines, rectangles, triangles etc).

4. *Shape Definition & Manipulation*

On completion of the first edit, the line data is further processed so as to disassociate the geometry into discrete single line segments - thus, any line disected by another is split into two individual lines as shown in Figure 8.



graphic primitive data

shape ref.	no. of edges	line i.d's				
A	5	1	2	3	4	9
B	4	5	6	8	12	
C	5	7	9	10	11	12

Figure 8 Line Disassociation And Primitive Identification

A search algorithm is then invoked by which the perimeter of each enclosed space is traced and the lines which define each enclosure boundary are stored as an additional relation set in the data base. The system has thereby abstracted geometric Shape Primatives from the original data in a format which may now be used in a Second Edit phase to manipulate complete primatives eg using repeat, move, mirror, rotate and scale.

Again, the dynamic graphics option enhances this second edit phase by allowing dynamic translation of whole primatives across the screen with or without distortion of shape primatives to revised proportions into new locations.

During the final phases of this stage the option is included to extend the data base to include a third dimension (ie height) where this is required. This may be done by vertical extrusion of any line or any whole shape primative as necessary.

5. *Interfacing With Applications Software*

With the completed graphical data held in a relational structure, re-formatting this data in a form suitable as input to other (applications) software becomes routine. Currently, the software is designed to do this automatically for the creation of "GOAL" and/or "BIBLE" files but others may be introduced on demand.

SYSTEM CONFIGURATION

Figure 9 illustrates the hardware configuration of the system which at present operates either through an on-line central processor (DEC 10 System) or in tandem with the local processing power of the Tektronix 4054 graphics system.

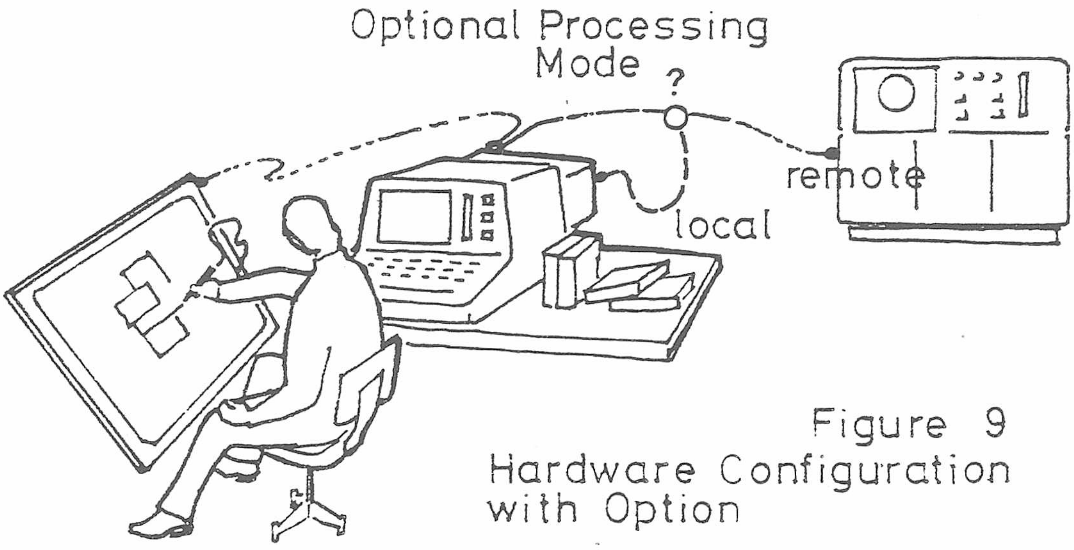


Figure 9
Hardware Configuration
with Option

The added advantages of using the Tektronix 4054 with refresh graphics potential are illustrated in Figure 10 and control of the software (under either configuration) is via a sequence of 3 menus of commands which are illustrated in Figure 11.

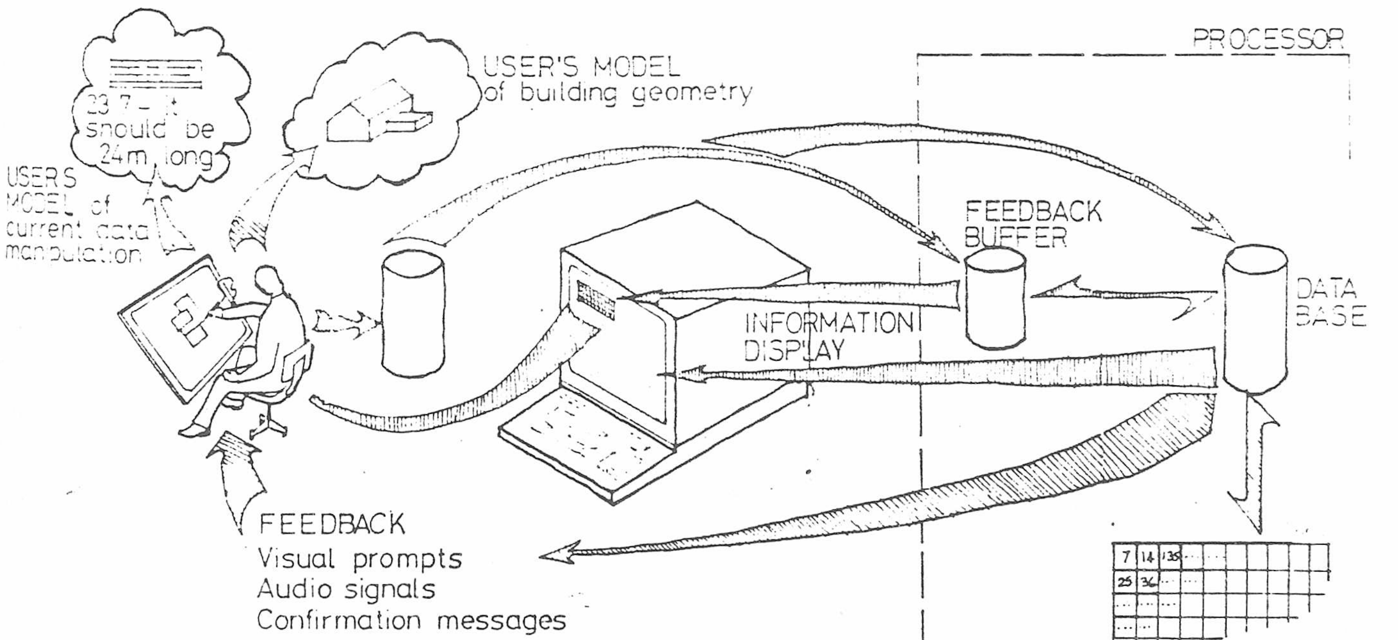


Figure 10 Graphical representation of componants involved in man/machine interaction utilising refresh graphics capabilities

Mainframe computing is carried out using FORTRAN but the refresh graphics option is programmed in BASIC.

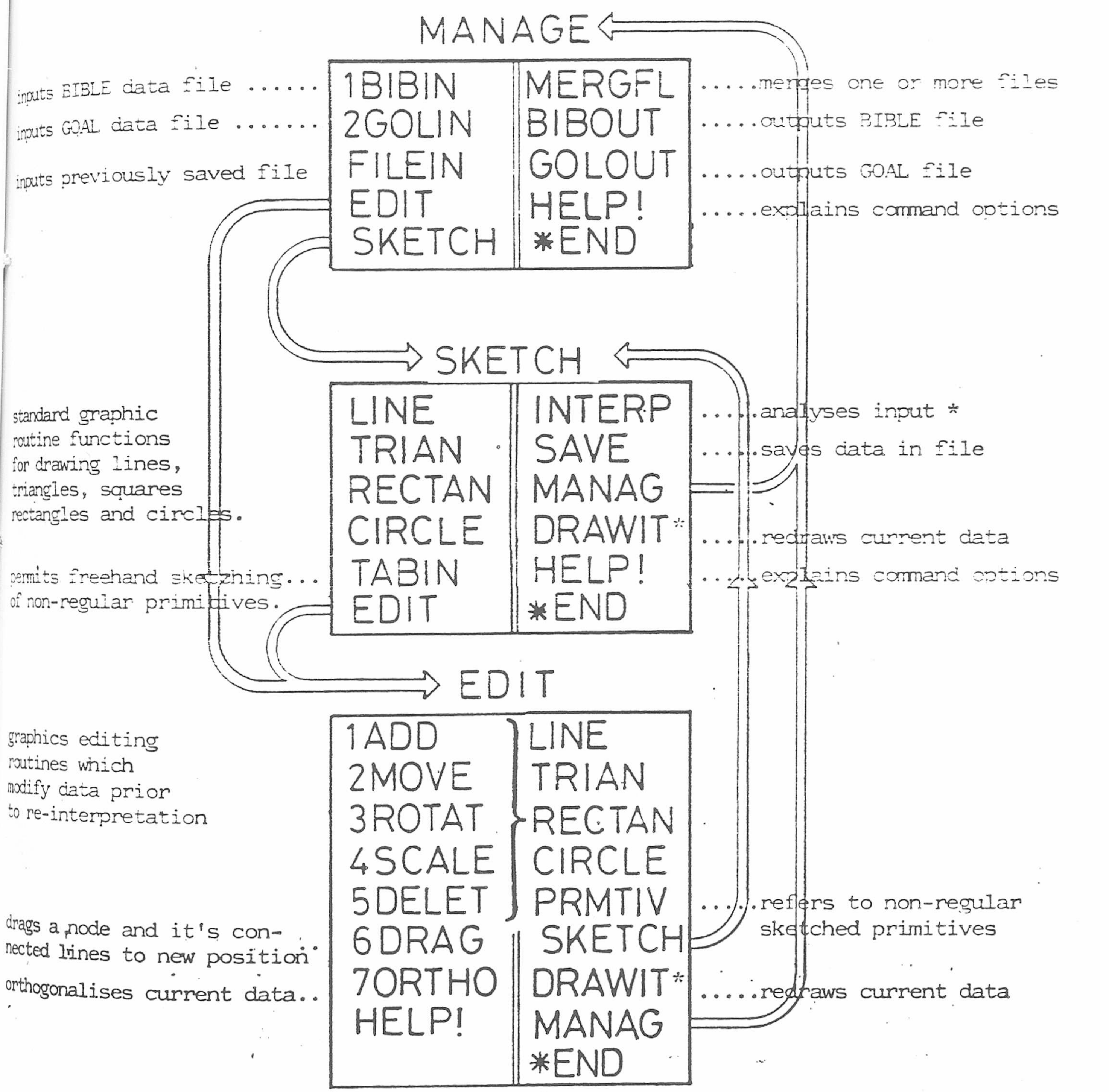


Figure 11 ENIGMA Menu Structure

* not required on TEKTRONIX 4054

SUMMARY

The system described here has the following properties ...

(a) *Easy for the user to operate requiring a minimum of computing knowledge.*

(b) *Closely emulates traditional working methods.*

(c) *The system feels and responds like a drawing board.*

This is because much of the data input is via continuous digitisation of freehand sketches.

(d) *It quickly encaptures large amounts of complex data.*

The interpretation algorithm overcomes loss of precision by ...

(e) *Replacing lost corners.*

(f) *Returning curves as a sequence of straight line segments.*

(g) *Maintains the integrity of each line and its relationship with others. during editing.*

The relational data structure used makes re-formatting straight-forward, thus

(h) *Crude first sketches can be quickly interpreted and re-formatted for interfacing with applications software*

as a consequence of which ...

(j) *At the earliest stage in the design process it becomes easy to obtain more instant feedback on the consequences of design decisions.*

REFERENCES

1. *Standing Committee on Computing and Data Co-ordination "Computing and Communication in the Building Industry" D.O.E. March 1979*

2. *C.I.C.A., Cambridge "Fact Sheet on the Use of Computers in the Construction Industry" C.I.C.A. Bulletin March 1982*

3. *ACM/SIGGRAPH Workshop, Battelle Seattle Conference Centre, June 1982 "Graphical Input Interaction Technique (GIIT)" Reported in Computer Graphics January 1983*

4. *Leifer, D., Scott Sutherland School of Architecture, Aberdeen "Man Before the Machine: An Issue in CAAD" BoCAAD, University of Strathclyde, Glasgow Vol 43-46 February 1983*

5. *Halasz, F. and Moran, T "Analogy Considered Harmful" Proc. Conference on Human Factors in Computer Systems Gaitherburg, Maryland 1982*

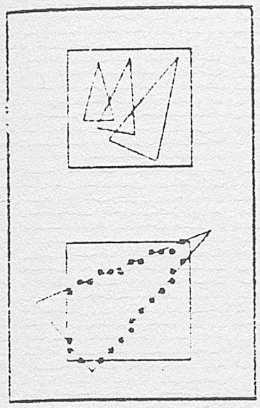
6. Bijl, A. et al Ed CAAD, University of Edinburgh
"Building Description Techniques to Support Computer Aided Design and Production"
Section 25 of SERC Grant Application December 1981

7. Bijl, A., Stone, D., Rosenthal, D., Ed CAAD, University of Edinburgh
"Integrated CAAD Systems"
Final Report to D.o.E. Project DRG 470/12 March 1979

8. Sussock, H., ABACUS, University of Strathclyde
"GOAL"
ABACUS Occasional Paper No 62 August 1982

9. Sussock, H. and Ross, J., ABACUS, University of Strathclyde
"BIBLE"
ABACUS User Manual No M11 December 1982

10. Williams, R. IFIP Congress, Stockholm
"On the Application of Relational Data Structures in Computer Graphics"
Information Processing 1974



APPENDIX D

CONTENTS

Occasional Paper No 14

Current CAAD Research at
Scott Sutherland School of Architecture

B Hammond (Jan 85)

Current CAAD Research at SSSA
by B Hammond

1.0 INTRODUCTION

Current CAAD research at Scott Sutherland School of Architecture, which is funded through an SERC studentship, is concerned with the development of a 'realistic' computer model for the analysis of room layouts. It is an extension of previous work done at this school by Langskog, combined, to a certain extent, with the ideas of the author and others(1) on man/machine communication.

2.0 HARDWARE

The research utilises the following hardware configuration - a Tektronix 4054 graphics terminal with refresh graphics, combined with a Tekronix 4907 file manager and disk drive. Peripherals, not essential to the research but available if required, include a Tektronix 4663 plotter and a 4631 hard copy unit. RS 232 communication with RGIT's DEC 2050 mainframe is also available for what it's worth!

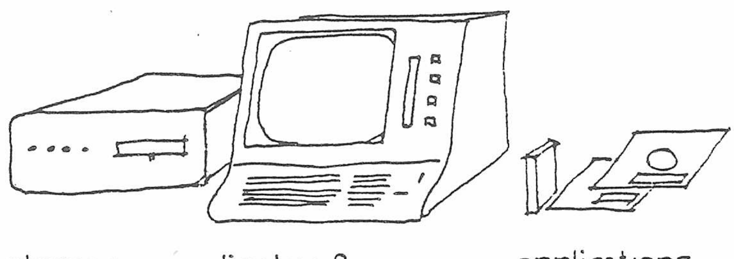


FIG 1 - Hardware Configuration

3.0 FOUNDATIONS OF THE RESEARCH

Langskog was perhaps the first to develope a technique which explicitly appraises the efficiency of floor area usage (2); although, he did state that certain more general graphical analysis techniques had been proposed previously (3,4,5).

Langskogs model was based on three different measures of layout efficiency:-

(i) A shape penalty
The more closely the room resembles a square, the more efficient it becomes, because it requires less perimeter wall length to bound the room area.

(ii) An area utilisation penalty
The greater the area of the room that is used, either explicitly by an item of furniture, or implicitly by the area required to use the furniture, compared to the total room area - the more efficient is the layout since there will be less wasted space.

(iii) An overlap penalty
Each furniture element has, as indicated above, an associated user space (the model actually uses two user spaces - an essential user space and a desirable user space). The degree to which user areas overlap can be measured, and the less the overlap area, the more efficient the room layout since the use of one element of furniture is less likely to interfere with the use of another.

The principals of this model provided the foundation for the current research.

4.0 DEFICIENCIES DISCOVERED IN LANGSKOG'S MODEL

Examination of Langskog's model identified three areas where it was thought an improvement in the model could be achieved.

(a) extend the model to encompass three dimensional spaces (Langskog's model only dealt with two dimensional plan images).

(b) extend the model to encompass non-orthogonal geometry.(Langskog's model was limited to orthogonal and rectangular geometry, primarally because of the software implementation).

(c) amend the model, so as to encompass an element of positive association such that one furniture element could be specifically associated with another element.

These thoughts provided a starting point for the research and delimited the scope of the new layout analysis model.

5.0 THE NEW LAYOUT ANALYSIS MODEL IS DEFINED

Examination of Langskog's work provided the basis for a new model. The new model had four different measures of efficiency.

(a) A room shape penalty. (SP)
(b) A volume utilisation penalty. (VU)
(c) A volume interpenetration penalty. (OP)
(d) An association penalty. (AP)

The first three measures were essentially those of Langskog, amended to take account of non-orthoganal, non-rectangular geometries and a degree of three dimensionality (extrusions of two dimensional shapes).

As is mentioned later, while the software was being implemented, it was noted that the volume interpenetration penalty occassionally gave unexpected results such that an increase in volume interpenetration could actually reduce the OP score.

5.1 THE VOLUME INTERPENETRATION PENALTY

To go into this in a bit more detail. The OP score, was based on the overlap of user volumes, and although changes were made to the new models method of describing user volumes, to ease software implemantation, the principal remained the same. The difference in the way that object outlines were described is shown in the diagram below.

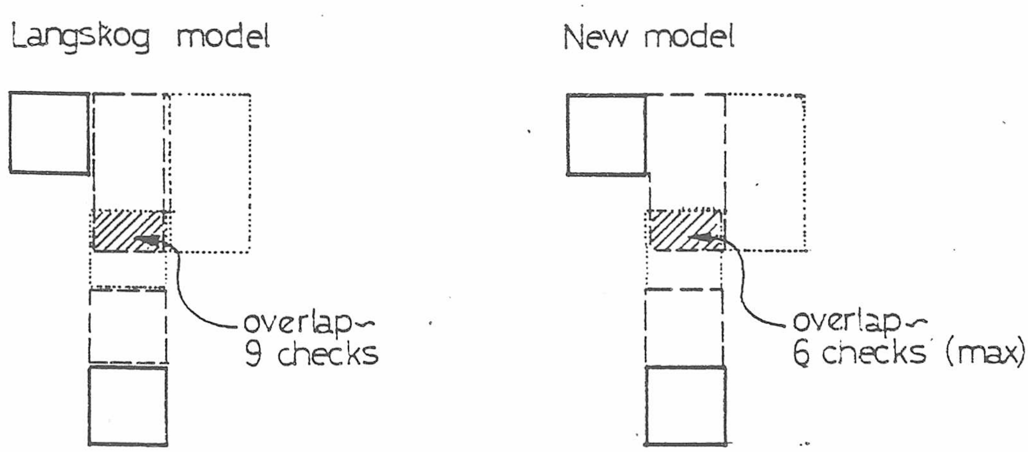


FIG 2 Comparison of models

As can be seen, if each user volume encloses the smaller or more essential volume, the number of overlap checks can be significantly reduced.

It was also felt that the effect of a single overlap instance should reduce as the problem became more complex (ie. as more furniture elements were added to the room)

Thus Langskog's OP equation changed from:-

$$OP = \frac{\sum_{1}^{n}(A \times W)}{\sum_{1}^{n} A}$$

to:-

$$OP = \frac{\sum_{1}^{n}(A \times W)}{\sum_{1}^{m} AT}$$

where:-

n = number of overlap instances.
m = number of elements in the room.
A = overlap volume of a single instance.
W = a penalty factor for that instance.
AT = largest user volume of each element in the room.

This change also ensured that any situation which caused a reduction in the total overlap volume, also caused a reduction in OP.

5.2 THE ASSOCIATION PENALTY

In certain circumstances, for example in a domestic kitchen, certain units of furniture are required to bear a particular relationship to other units of furniture. eg. a cooker should be between 1200 mm and 1800 mm from the sink, and between 1200 mm and 2400 mm from the food store(6).

Langskog's model had no way of dealing with this type of relationship.

The new model incorporated an association matrix of pairs of furniture elements and a min/max distance they should be apart. Furniture elements pairs that had such an association and failed the min/max test attracted a penalty based on the degree of failure, and a penalty factor for that failure. The penalty factor could be different for pairs that failed the min test as opposed to the max test.

6.0 SOFTWARE IMPLEMENTATION

As was indicated in section 2.0 the new model was implemented on a Tektronix 4054 with dynamic graphics. This tied in well with work being done at the time on 'user interfacing'(7,8) at this school and allowed a more friendly 'front-end' to be placed infront of the model.

The author was responsible for designing and implementing (including programing and debugging) all the software described below. The program suite convienently divides into five segments.

(a) a controller program (main menu and disk utilities)
(b) room creation - dynamically creating the basic room layout and positioning room elements such as doors, windows, floors and ceilings.
(c) furniture creation - defining furniture elements and their associated user volumes and graphic displays.
(d) menu creation - setting up a particular 'job file', incorporating a specific room and menu of furniture elements. Also setting up two penalty arrays dealing with overlap and association penalties.
(e) layout - dynamic manipulation of furniture elements within a specific room shell and also interpretation of specific geometries through the model.

The first four sections make quite extensive use of line 'rubber banding' and dynamic feedback as input techniques. Screen ergonomics of these sections was largely ignored, however, as it was not anticipated that anyone other than the author would be making use of them.

The last section allows the user to manipulate the 'job file' dynamically, using refresh graphics to position objects and as part of the feedback process.

The positioning routines use an auto-adjust mechanism to slide units of furniture against each other or against the external wall, if that is desired, so as to ensure that they do not overlap.

The modelling element is also contained in this section; as is those aspects analysing the output from the model. These functions are largely displayed in dynamic graphics.

Functions to save, recall and display partially completed jobs are also included in this section.

Clearly the use of a 64k micro introduced some limiting constraints mainly in terms of memory, and consequent speed of operation problems caused by the requirement to overlay, either data or program segments. This is particularlty evident in the final section where there is insufficient memory to hold display and geometry files and the programs to manipulate them simultaneously.

7.0 PRELIMINARY VALIDATION OF THE MODEL

A short exercise was carried out using a domestic kitchen as a test situation. It was intended primarilly as a means of checking the numerical accuracy and ergonomics of the computer program. This was extended to a preliminary validation exercise to ensure that for a clear-cut situation the computer assessment of most efficient room to least efficient room matched that of the designer (well two designers!)

The selection of kitchens used in the exercise (and I make no apology for the quality of their design - since that was not my main concern at the time) are included at the end of this paper as an example of the type of display given to the user whilst operating the system. It should be noted that most of the geometry and feedback sections of the display are actually held in refresh memory and change as the user operates the system.

Also note that the diagrams no longer have efficiency values shown, in case you feel tempted to rank order the diagrams yourself!

8.0 PRESENT POSITION OF RESEARCH

The research has reached the point where an attempt must be made to validate the model.

It is proposed that this be done in several stages.

(a) An attempt should be made to find a non-correlation result: ie. a selection of designs should be prepared which the designer finds very difficult to place in order of efficiency. It is thought that this will be discovered where changes in efficiency (according to the computer) are marginal, or where the room is an extremely unsuitable shape for the function resulting in all solutions being some way from the ideal.

(b) A larger selection of designs will be shown to several groups of designers, with different levels of experience. Hopefully this will show that the designers and the computer generally agree on which rooms are the more efficient, except where differences in efficiency are marginal. In those cases, it is expected that the more experienced designer will show a greater correlation to the computer because of his ability to discriminate between marginal designs.

(c) Inexperienced designers will be 'let loose' on the computer with the objective of improving the efficiency score of a given design (a pre-check will be required to ensure that the design can be improved). The view of these inexperienced designers should give some unbiased views on the ergonomic performance of the 'layout' screen and help to identify any problems there. The 'improved' and 'unimproved ' designs might then be presented to the same groups of designers as at (2) with the hope that all groups would correctly identify improved over non-improved designs, as the more efficient.

9.0 REFERENCES

(1) B. Hammond, D.' Leifer, 'A Graphics Interface to Complement Traditional Techniques', published and presented at PARC 83, Proceedings of the International Conference on Computers in Architecture, On-line Publications Ltd., Pinner, UK, 1983,pp 321-330

(2) K. Langskog, L. Laing, 'A Computer Aided Architectural Design Technique for the Appraisal of Domestic Activity Spaces', CAD 84, 6th. International Conference and Exhibition on Computers in Design Engineering, Butterworths, Surrey, UK, 1984, pp165-181

(3) R. Bjorkto, 'Vurdering av boligens bruksverdi - metodesporsinai' (Evaluation of the Usable Value of the Dwelling Plan), BD-ORIENTERING (REPORTS FROM THE NORWEGIAN DIRECTORATE OF HOUSING), no.3, 1963, pp18-33

(4) E. Svennar, 'Boligens planlosning (The Dwelling Plan) , HANDBOOK NO. 25, Oslo, Norwegian Building Research Institute, 1975

(5) A. Thiberg, ROOM LAYOUTS: REPORT R:41 1970. Stockholm. Swedish Research Institute, 1970

(6) Ed. P. Carolin, 'Conventional Planning', ARCHITECTS JOURNAL. 3 Oct. 1984, No.40, Vol.180, Architectural Press Ltd., London, pp49-79

(7) B. Hammond, 'Feedback Aids to Geometry Data Input', SSSA. Occassional Paper No.4, Unpublished

(8) L. Laing, D. Leifer, B. Hammond, 'Enigma: An Enhanced Interpretive Graphics Module for Architects', SSSA. Occassional Paper No10, Unpublished

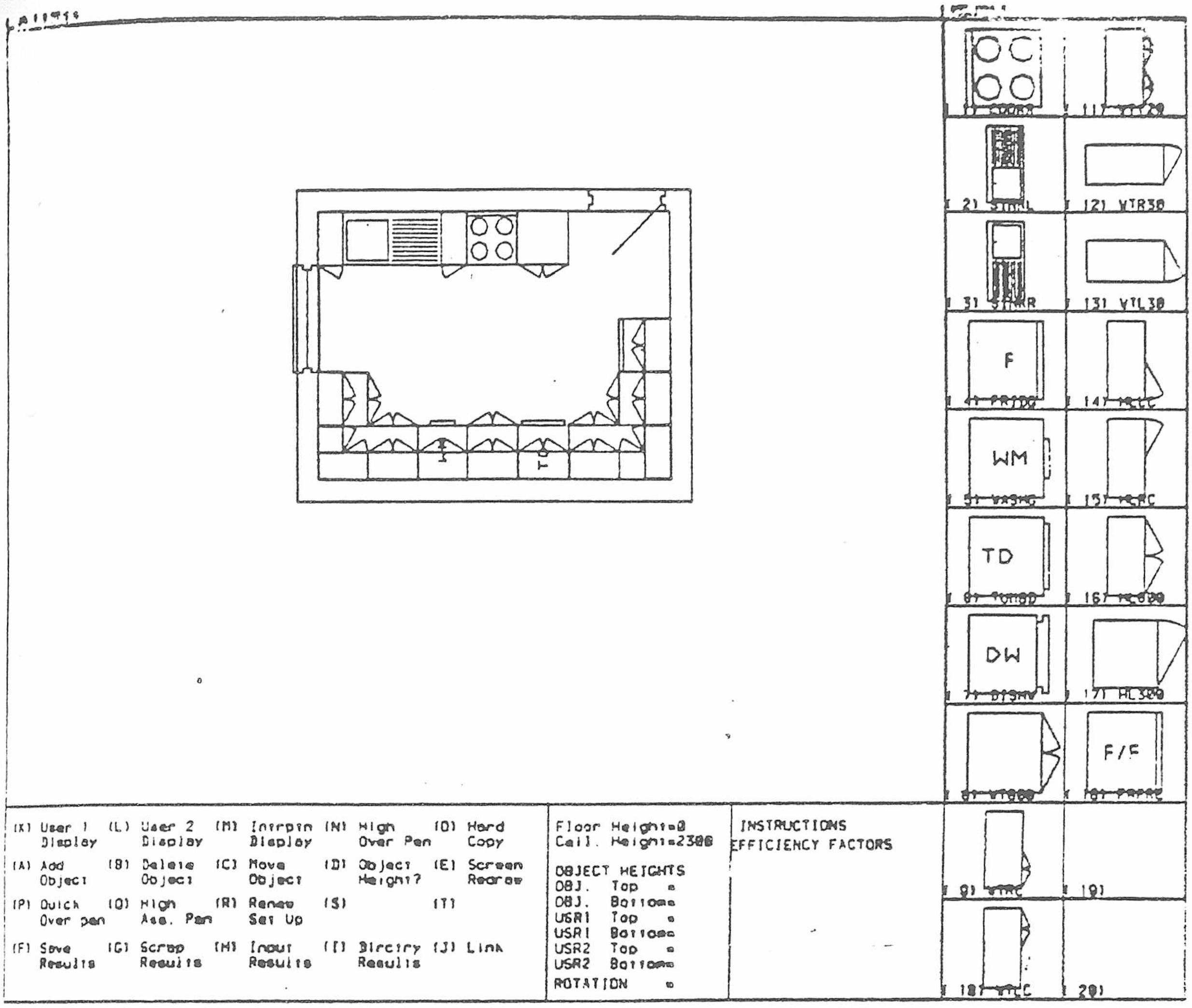


FIG 3 Sample Kitchen Layout

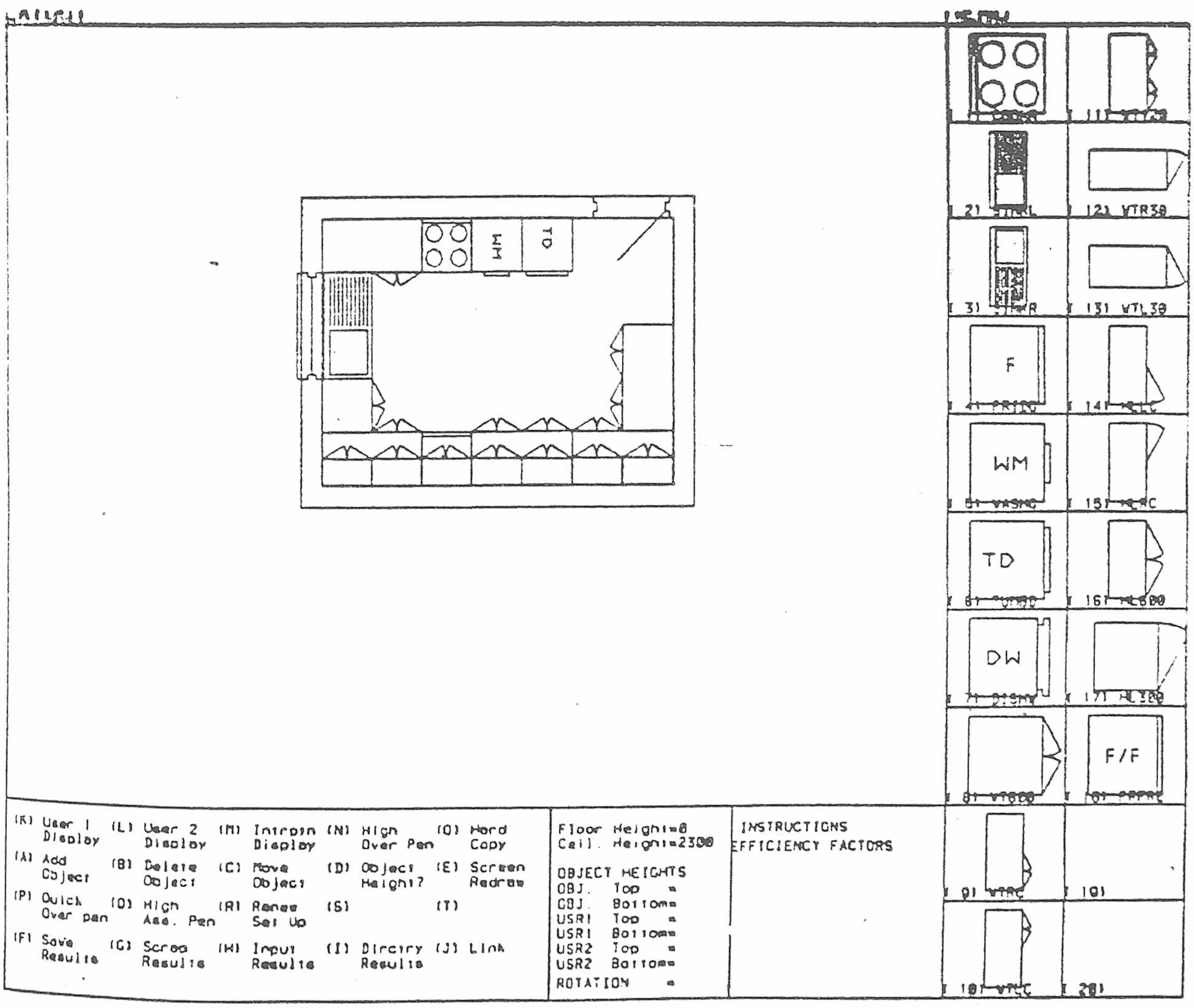


FIG 4 Sample Kitchen Layout

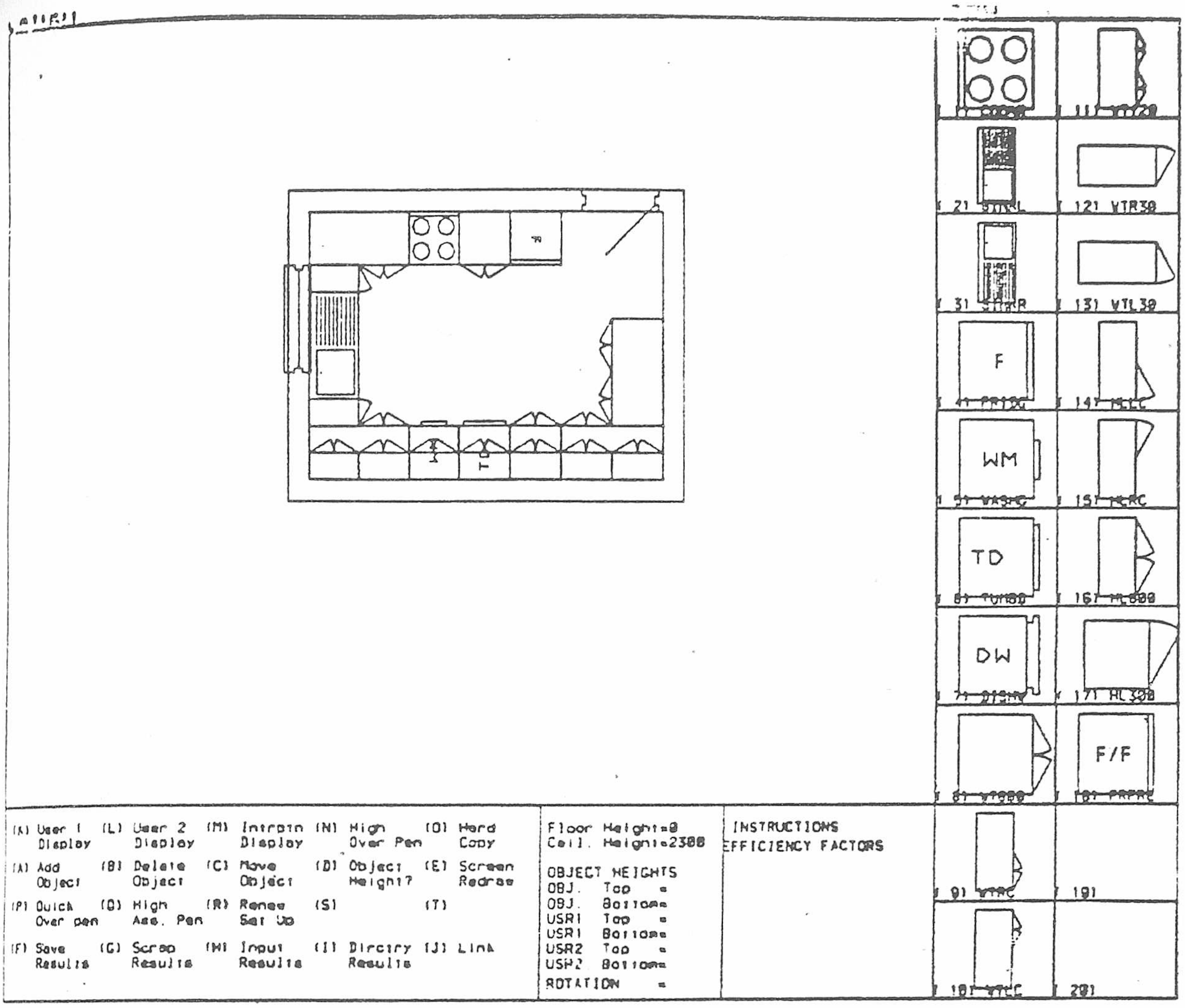


FIG 5 Sample Kitchen Layout

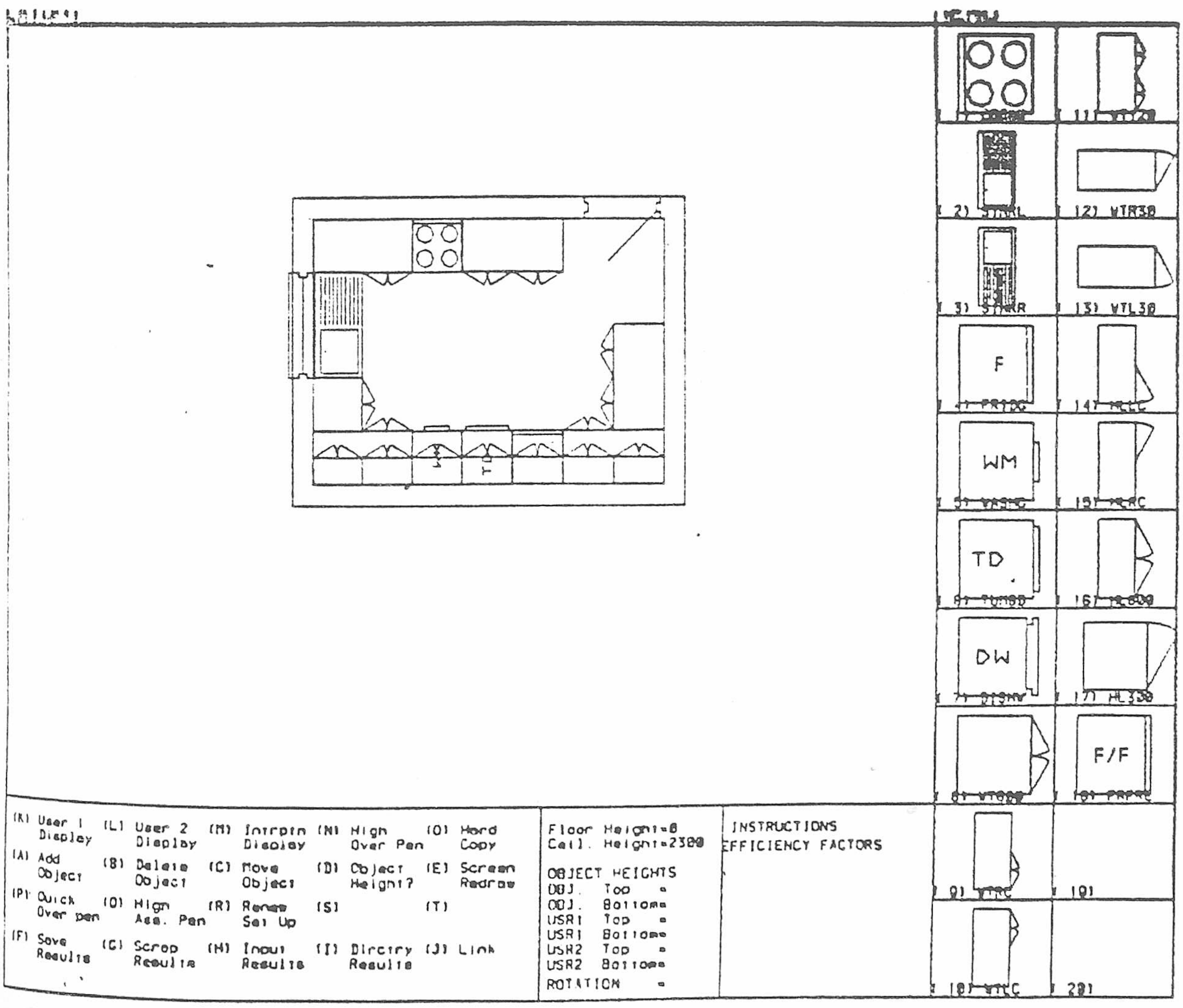


FIG 6 Sample Kitchen Layout

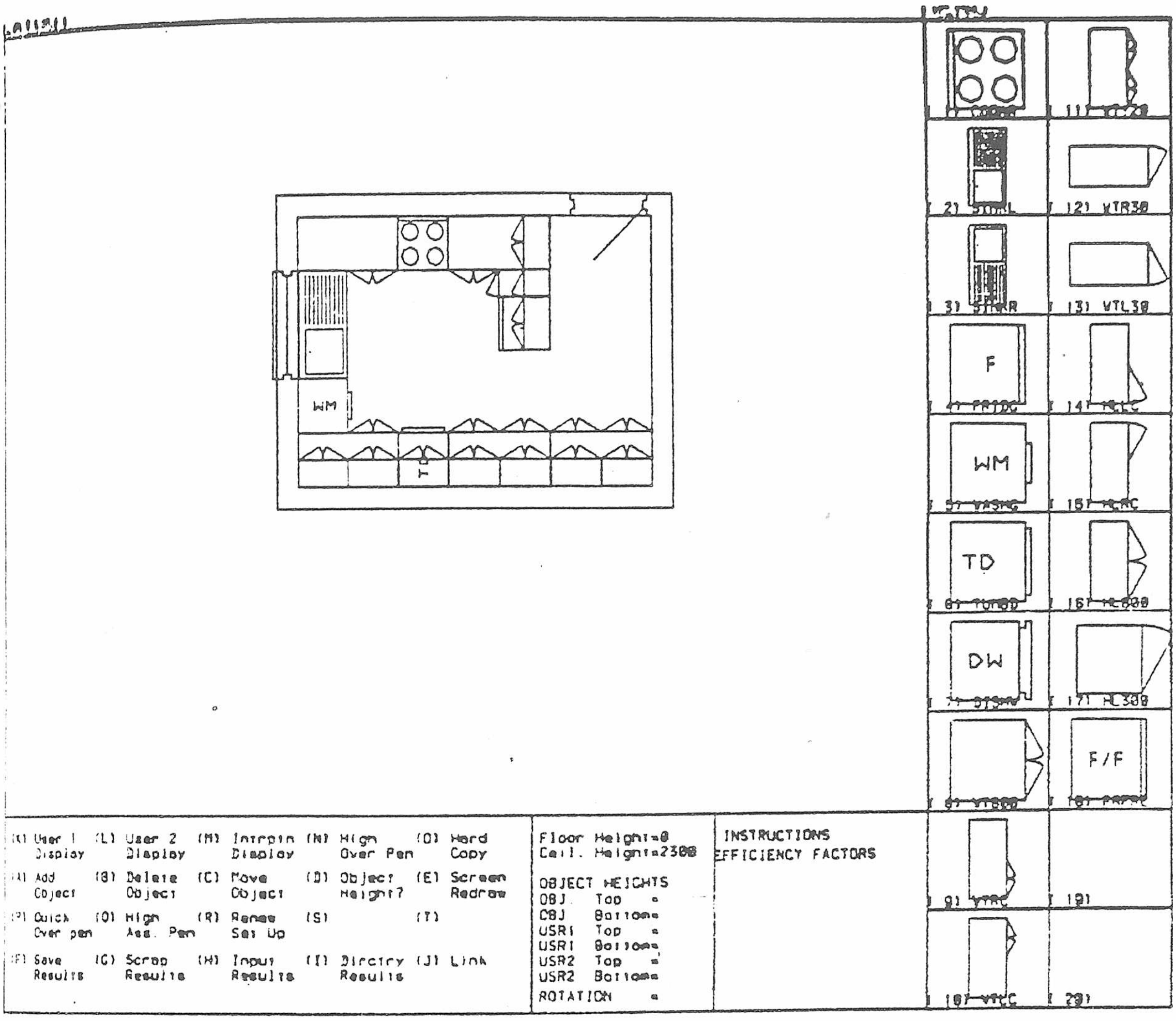
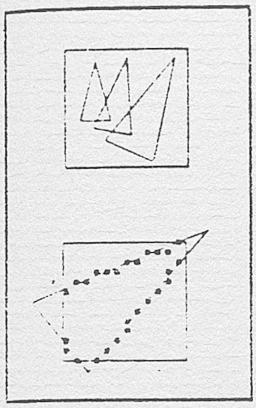


FIG 7 Sample Kitchen Layout



APPENDIX E

AREA DISSOCIATION
ALGORITHM

CONTENTS

E.1.0 INTRODUCTION

This appendix describes an algorithm for deriving ordered data sets, representing two dimensional space primatives. from unordered lists of line elements.

E.1.1 Origins

The algorithm originated during the earlier part of this research project, at a time when a method of giving the computer sufficient knowledge to derive additional input information, relevant to the architectural user, from the context of the geometry input was being considered (see appendixes A, B and C).

Specifically, the algorithm seeks to derive two dimensional space data from one dimensional line data. That is, if a building plan geometry were described in terms of lines the computer has sufficient intelligence to discard non-relevant lines and to construct a new data set consisting of sets of ordered co-ordinates, where each set of co-ordinates defines an individual room space.

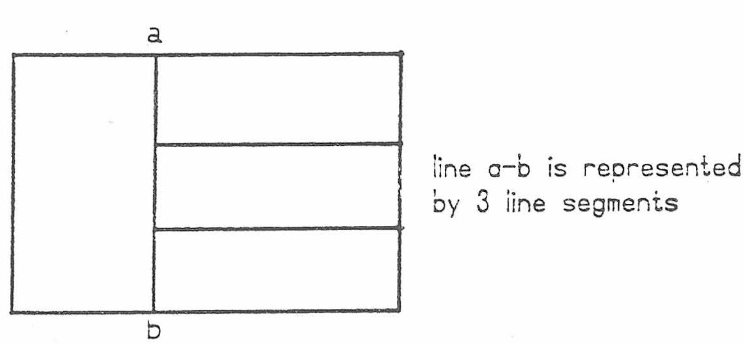
Although this algorithm was never incorporated in the methodology for room analysis, its development and implementation was important in the consideration of the algorithm for Area Overlap (discussed further in appendix F). This aspect is further discussed in section E.4.0.

E.1.2 Algorithm in brief

The algorithm is fully described in section E.3.0, but there follows a brief description of the concepts underlying the algorithm.

The algorithm works, as any algorithm must, by tracing round each individual polygon. To reduce tracing time, and the risk of polygon duplication or distortion, use is made of the fact that, in two dimensional space, no individual line segment may be a constituent part of more than two polygons, see fig.E.1 below. The author believes that it is the use of this property which makes the algorithm interesting in it's own right.

Fig. E.1 Definition of line segment.



E.2.0 PREREQUISITES

This section delineates the necessary background data structure for the successful use of the area dissociation algorithm. It also states the requirement for subsiduary routines necessary for some of the processing tasks within the algorithm. Also, some of the terms used in the description of the algorithm are defined.

E.2.1 The Data Structure

This sub-section describes the data structure used within the algorithm. It would be possible to use other data structures in the algorithm provided apppropriate changes were made to the algorithm.

The data structure utilised is defined below:-

1. The geometry description of each line segment is in the form of a start and a stop co-ordinate pair, which may not be co-incident.

2. Any line segment may join any other line segment only at it's start or stop point. See fig.E.1.

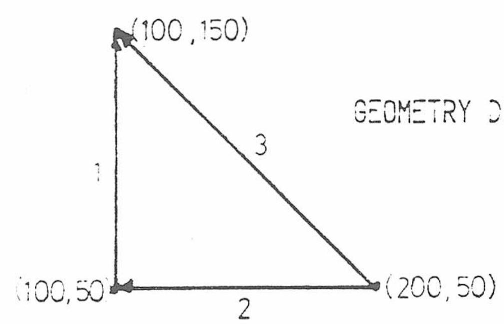
3. No line segment may be congruent with any other line segment.

4. Each description of a line segment is assigned an address for ease of subsequent location and recovery of the geometry description.

5. At the address of each line segment, data encoding the address of all other line segments that join that segment is noted together with a note as to whether they join the segment at it's start or end point.

The diagram below (fig.E.2) may assist in visualising the nature of the data structure.

Fig. E.2 Data structure for area dissociation algorithm



line address	geometry data: start co-ord. x	start co-ord. y	end co-ord. x	end co-ord. y	connection data				
1	100	50	100	150	2.1	3.2			
2	200	50	100	50	1.2	3.1			
3	200	50	100	150	2.1	1.2			

line address

start co-ord. x,y

end co-ord. x,y

list of joining lines in form n.m where:-
n=address of line being joined
m=1 when joined at start point
=2 when joined at end point.

E.2.2 Subsiduary Routines

This section indicates those routines which are neccessary to the main algorithm, but are so commonplace as to require no further explanation.

1. A proceedure for deriving the angle of a line segment relative to fixed base line e.g. the x-axis, is required.

2. The above proceedure is an important part of a second proceedure to determine the most acute angle subtended by two line segments, in either a positive or negative direction, relative to the direction of a base line segment in the pair of segments.

3. A proceedure to swop line segment elements within the data structure, whilst maintaining the integrity of the data structure, is required.

4. Similarly, a routine to turn a line segment around so that the start and stop co-ordinates are exchanged, along with end point connection data, will prove useful.

E.2.3 Nomenclature

There follows definitions of some words and phrases used in the description of the area dissociation algorithm.

Line segment - a line that joins a start and stop co-ordinate such that no other line in the data set meets that segment at a position other than it's start or stop point.

Line address - a tag attached to the description of a line segment so as to be able to indirectly reference either it's geometry or it's end connections.

Line connections - a note of addresses of all line segments that share the start or end co-ordinate of a particular line segment.

Line tab - a line segment within the data set which does not partly enclose a polygon.

Polygon tracing - following the line connections of line segments in a particular direction so as to determine those line segments which enclose a single polygon.

E.3.0 THE AREA DISSOCIATION ALGORITHM

This section examines, in detail, the mechanics of the algorithm.

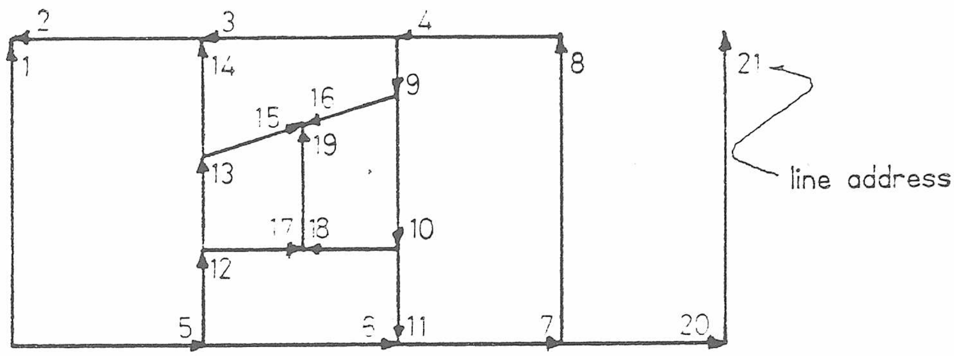
E.3.1 Implementation

The algorithm was implemented on a Tektronix 4054A graphics terminal with dynamic graphics and using Tektronix Graphics System Basic (A update version). This Basic supports CALL by name subroutines and IF...THEN...ELSE conditions. Implementation in this language does not preclude the implementation of the algorithm in other languages and on other machines.

E.3 2 Method of Detailing the Algorithm

The coding for the above implementation is long and complex and for that reason is not included here. Rather a verbal description of each sub-proceedure is given, with reference to an example data set, so as to give a firm indication of the logic employed. Abbreviated flow charts of each process are also given. The sample geometry is shown diagramatically below in fig.E.3.

Fig. E.3 Example geometry used to assist description of area dissociation algorithm



E.3.3 Preliminary Processing

The purpose of the preliminary processing is to discard all lines that cannot possibly form the bounding edge of a polygon; that is, the elimination of line tabs.

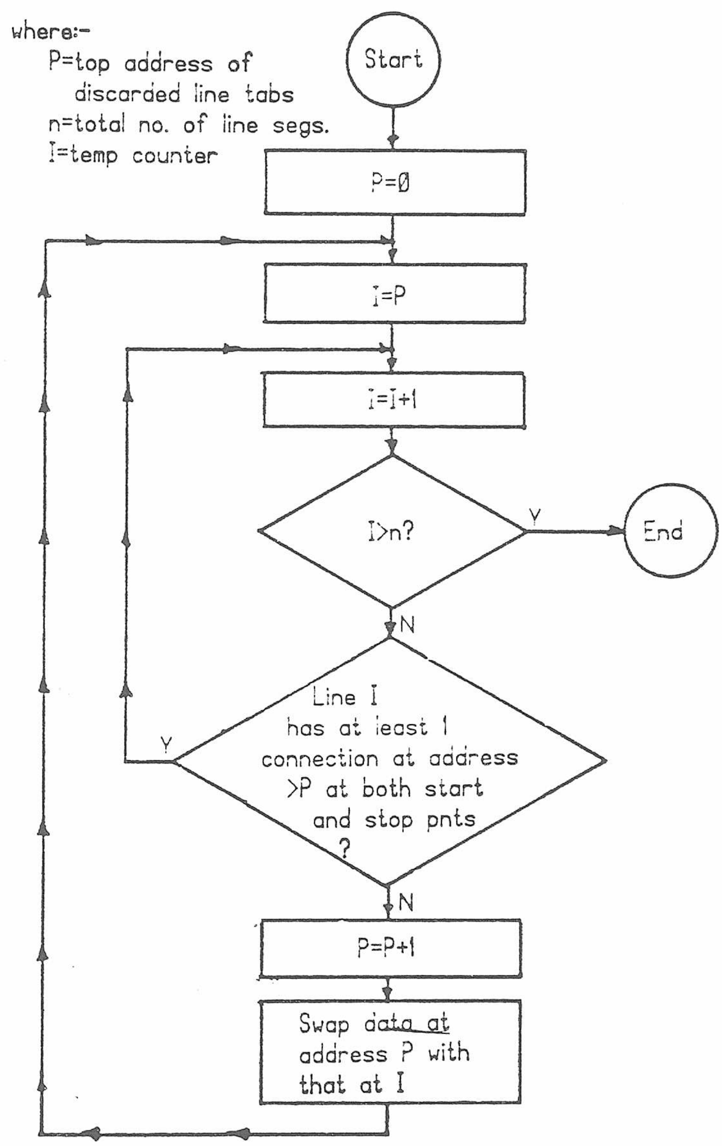
Conceptually, the preprocessing initially rejects all those line tabs which are unconnected to any other line at, at least, one end. Such as line 21 in fig.E.2.

Next all line segments that are only connected at either their start or their end point, to a line segment that has already been discarded, are, themselves dismissed.

The latter stage is iterative, and the final stage to the pre-processing, is a pass through the data set to confirm that no further line segments should be discarded.

Obviously, the actual coding of this pre-processing can be compressed by the combination of the first two stages as illustrated in the flow diagram E.4 below.

Fig. E.4 Flow chart showing tab filtering proceedure



E.3.4 Locating the First Polygon

The first step is to initialise several arrays and tags. Temporary arrays are required to hold the x,y co-ordinates of the discovered polygon. The other important piece of initialisation is to create a line tag array with one value for each line segment in the data set, with the exception of those line segments which have already been discarded by the pre-processing described in the previous sub-section.

Each location in the tag array is set to 2. As a line is included in a polygon, the corresponding tag array value is decremented by one. Since each line segment can only belong to a maximum of two polygons, no value in the tag array can be less than zero, and any line segment whose tag value has already reached zero need not be considered again.

The first pass through the data set, which usually uncovers most, if not all, of the polygons within the data set, takes as a starting point, the line with the lowest address and a tag value of two. Immediately

that tag value is set to one, and the end polygon co-ordinates are set to the start co-ordinate of that line segment.

A test is then applied to see if the end co-ordinates of the line segment equal the end polygon co-ordinates. (Given the data structure described earlier, the first two lines must fail this test). The line segment that passes this test completes the first polygon by having it's end co-ordinates written off to the next slot in the polygon holding arrays.

Line segments that fail this test have their end co-ordinates written off to the next slot in the polygon holding arrays and the search starts for the following line segment of the polygon.

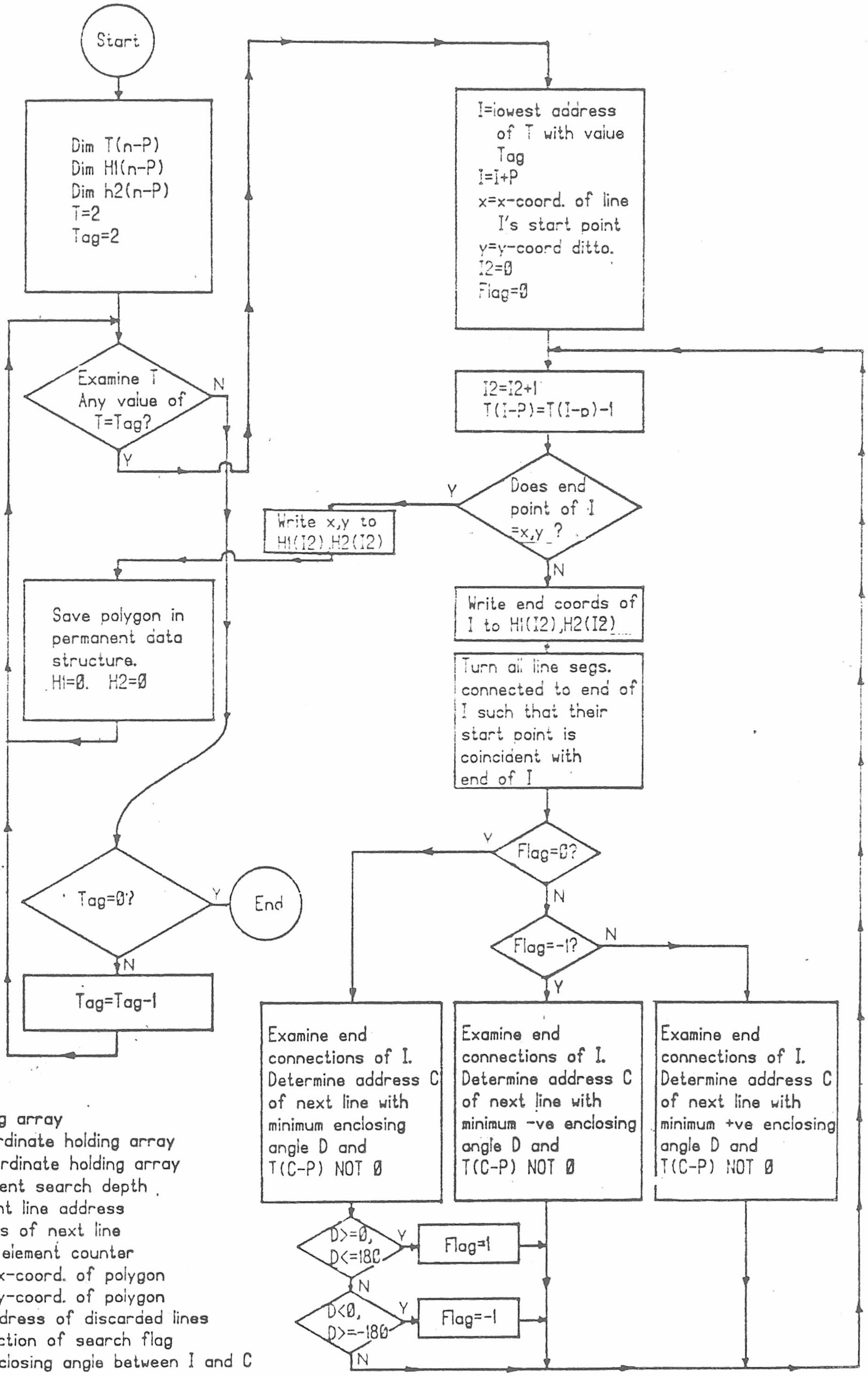
All lines connected to the end point of the first line segment are turned so that their start point is coincident with the end point of the original line segment.

A minimum positive enclosing angle, and a minimum negative enclosing angle are then calculated between the line under consideration, and all those potential next line segments. If the direction flag has already been set to either positive or negative, the appropriate line is selected and the trace continues from this new line.

If the direction flag is not set the minimum absolute enclosing angle sets the flag and determines which line is to be selected next. In the case of a tie between the absolute positive and the absolute negative values the positive direction is taken, and the direction flag is set appropriately, except when the absolute enclosing angle is 180 degrees, when the direction flag is left indeterminant.

This is illustrated graphically in fig.E.5. below.

Fig. E.5 Flow chart showing area dissociation mechanism

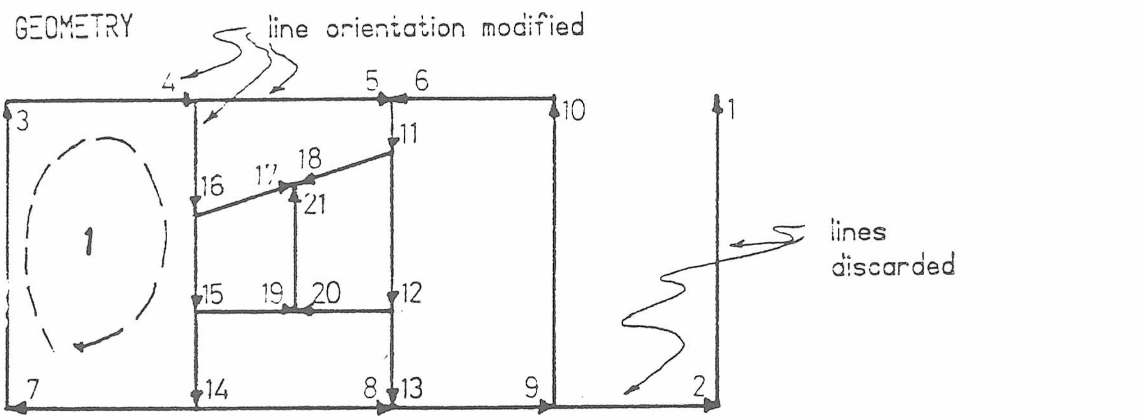


where:-

T=line tag array
H1=x-coordinate holding array
H2=y-coordinate holding array
Tag=current search depth
I= current line address
C=address of next line
I2=H1/H2 element counter
x=start x-coord. of polygon
y=start y-coord. of polygon
P=tag address of discarded lines
Flag=direction of search flag
D=min. enclosing angle between I and C

The diagram below (fig.E.6) shows the effect of the process on the example data set.

Fig. E.6 Illustration of changes to sample data set up to detection of first polygon



TAG ARRAY CONTENTS

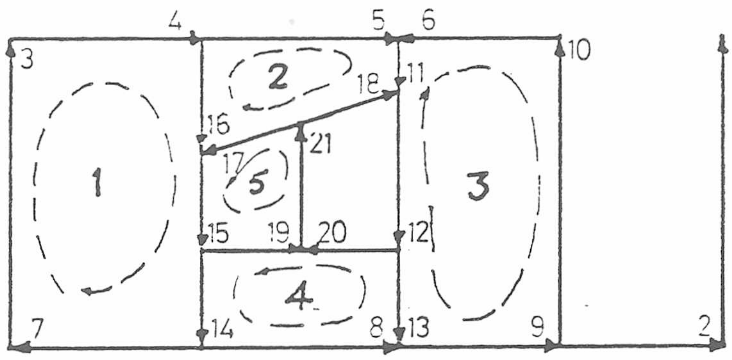
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	line address included for reference
1	1	2	2	1	2	2	2	2	2	2	1	1	1	2	2	2	2	2	tag value

E.3.5 Location of Subsequent Polygons

Further polygons are located in a similar manner, each time starting the polygon with the lowest addressed line with a tag value of two. Thus the next polygon would begin with line segment five. Fig.E.7 below, shows how the geometry and tag array contents have been changed after the first five polygons have been located, and no tag array value remains at two.

Fig. E.7 Sample data set changes whilst Flag=2

GEOMETRY



TAG ARRAY CONTENTS

3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	1	0	1	1

Notice that polygons 4 and 5 have been traced in an anti-clockwise manner, and that a 'hole' remains in the middle of the data set.

Had the line segments been ordered differently, it is highly likely that the hole either would not have existed, or been in a different location. Furthermore polygons 4 or 5 might well have been traced in a clockwise manner.

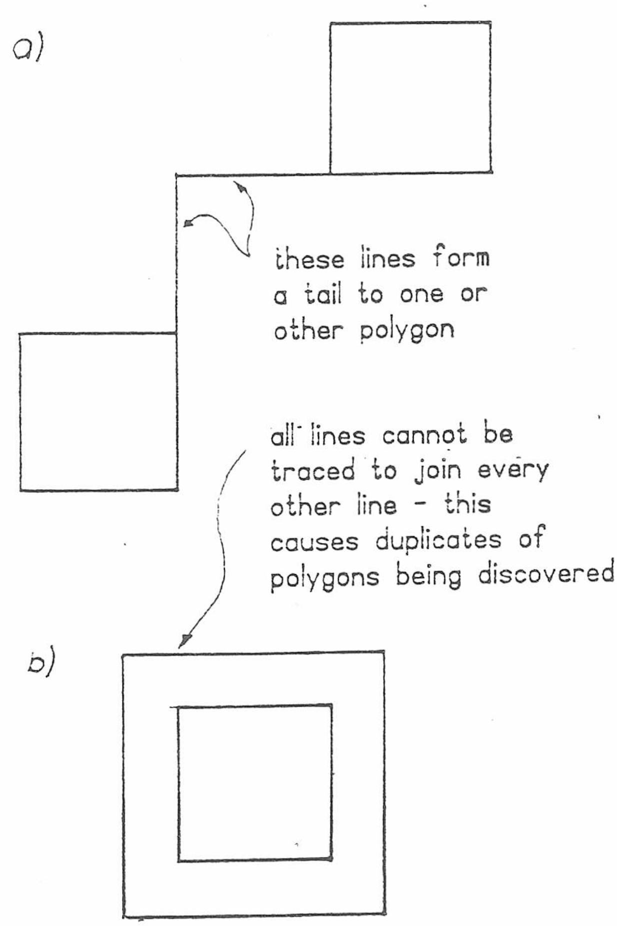
The last polygon (the 'hole' in the middle) and for that matter, the bounding polygon are uncovered in a manner similar to the first but starting with lines with a tag value of one.

When all tag array values are reduced to zero, all the polygons have been identified.

E.3.6 Exceptional Geometries

Although this algorithm, which was designed to assemble room geometries from line data, works satisfactorily for most architectural data sets, some geometries can cause the algorithm to produce invalid results. Fig.E.8 below illustrates.

Fig. E.8 Geometry types generating unexpected results for area dissociation algorithm



However, this problem should not be considered to be overly severe. The geometry type shown in (a) above is unlikely to occur in architectural data sets, and could be filtered out by some post-dissociation routine. Although (b) is a more likely problem in architectural data sets, the duplication it causes is much easier to filter out.

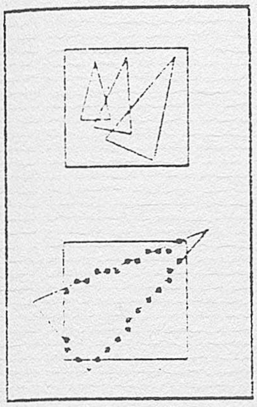
Algorithms to eliminate the problems inherent in geometries of type (a) and type (b) were not written, since the problems were comparatively easy to solve and contributed nothing to the conceptual core of the dissociation algorithm.

E.4.0 SIGNIFICANCE TO THIS RESEARCH PROJECT

Apart from the indirect benefit to the author's programming skills, this algorithm greatly influenced the author when designing the Area Overlap algorithm described in the next appendix.

This is essentially a trace algorithm and although it incorporates a tagging system and a relational data structure, it still requires a significant amount of processor time to discover the polygons. Without the benefit of a relational data structure, the time required to search a realistic line data base, would escalate unacceptably.

Thus experience of designing and implementing this trace mechanism prejudiced the author when considering how to create an Area Overlap algorithm, especially in the light of greater restrictions imposed on processor memory and processor speed.



APPENDIX F

POLYGON INTERSECTION ALGORITHM

CONTENTS

F.1.0 INTRODUCTION

This appendix starts by describing briefly, algorithms investigated to solve the problem of evaluating the area of overlap between two intersecting polygons. It then describes the algorithm devised by the author to solve that problem.

F.2.0 PUBLISHED ALGORITHMS

As far as the author is aware, no algorithm has been published which directly addresses the problem of determining the area of overlap between two intersecting polygons in vector mode.

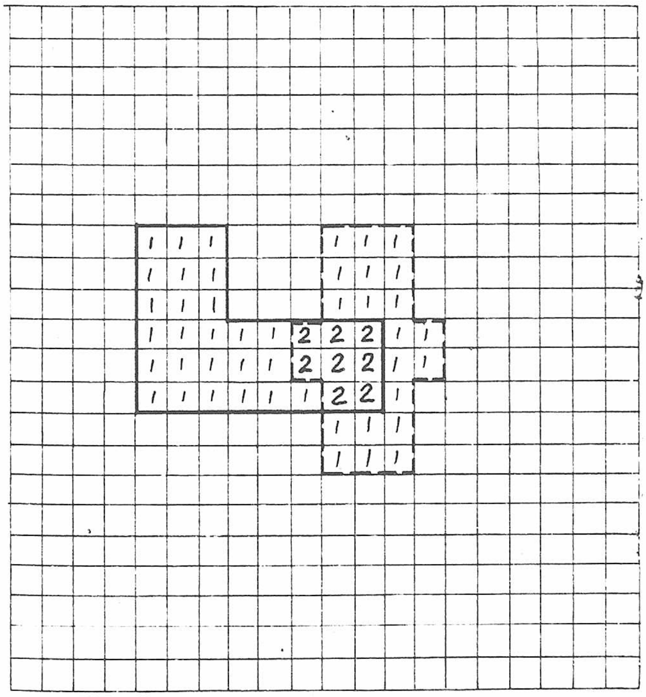
One algorithm addressing a similar problem in raster mode was discovered. However, the bulk of algorithms relating to the problem, were variations on hidden line/surface removal algorithms.

These algorithms are presented very briefly in the following pages.

F.2.1 Raster image handling

Scrivener (F.1), among others, has shown how to use a raster bit-map or framebuffer to extract partial images from the total picture. The output from his algorithm is remarkably similar to output from the area dissociation algorithm described in the previous appendix. However, the processing and structure of the input data is quite radically different.

Examining the fundamentals of his process, it was obvious that with modification the same type of process could be used to directly determine the area of overlap of two or more objects. Figure F.1 below indicates how the framebuffer concept could be applied to calculating areas of overlap.



Area of overlap can be calculated by counting the number of squares common to more than one object, multiplied by the area of a single square.

Fig.F.1 Raster Framebuffer concept used to calculate area of overlap

However, further examination of the practicalities of implementing this methodology on a small 64k vector display micro-computer uncovered several insuperable difficulties.

1 INCREASED COMPUTATION, due to reformating of vector data structures into raster data structures and back again.

2. MEMORY LIMITATIONS. To consider any room layout in any reasonable degree of accuracy (say to the nearest 50mm) would require very large arrays held in RAM. Furthermore, it would be quite difficult to decide an appropriate level of accuracy for any particular room layout.

3. NON RECTILINEAR FORMS. Depending on the algorithm used for rasterising a vector image, it is quite conceivable that two objects which do not infact overlap would attract a small overlap penalty due to the 'staircase' effects of rasterisation.

F.2.2 HIDDEN LINE/SURFACE ALGORITHMS

It is clear that determining the bounding edges of a polygon overlap in two dimensions has a great deal of similarity with the problem of removing hidden lines in a two dimensional projection of a three dimensional geometry.

Sutherland et al. (F.2) have studied hidden surface algorithms in considerable detail. A characterisation of ten such algorithms is presented below in Fig.F 2.

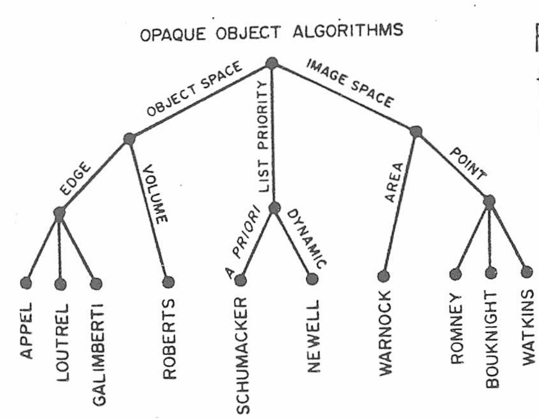


Fig.F 2 Characterisation of ten hidden surface algorithms (after Sutherland et al.)(F.2)

Those algorithms dependant on image space can be discarded, since they suffer from the criticisms leveled at raster image handling in the previous section. That leaves those that function, at least partially, in object space, and are concerned with determining the exact calculation of the picture display in object space uncorrupted by the resolution of the machine display.

In terms of the application under consideration, the object space algorithms make use of the coherence of the object space and also, generally, use a line tracing algorithm to clip one object against another.

However, since the algorithms all deal with three dimensional space, some of the idiosyncracies of the exact problem under consideration, (how to determine the area of overlap of two intersecting polygons in two dimensional space) remain unresolved,or at least unclear. For example, how do these algorithms deal with partial congruence of line or faces.

As a result of the above, it was determined that use would be made of the concept of coherence, mentioned above, in the search and sort operations of the new algorithm. For clipping of one object against another, machine dependant functions would be used (to calculate the intersection point of two lines), so as to reduce computation time in determining the degree of overlap between two polygons. The exact method employed is discussed more fully in the following section.

F.3.0 POLYGON INTERSECTION ALGORITHM

The following pages describe an algorithm for evaluating the outline of the polygon which encloses the area of overlap between two intersecting polygons.

The algorithm is also capable of determining if one of the overlapping polygons is completely inside the other.

F.3.1 Introduction

The geometry of each polygon is described by two lists of equal length - one of ordered x-coordinates, the other of the corresponding y-coordinates. The coordinates are ordered such that they trace out the outline of the polygon in either a clockwise or anti-clockwise direction. In addition the last coordinate is implicitly joined to the first.

F.3.2 Subsiduary algorithms

Two other subsiduary algorithms are used within the polygon intersection algorithm, but are not fully described in the following pages.

The first is the area of polygon, algorithm. The input of this algorithm is two ordered lists defining a polygon, as described above. The algorithm returns the area of the polygon as a positive number if the polygon is described in a clockwise direction and as a negative number if the polygon is described in an anti-clockwise direction.

The second is an inside polygon algorithm. The input is the polygon definition, as above, together with the x,y coordinates of a point. The algorithm tests the location of the point relative to the polygon and returns a check number with one of the following values:-

2-if the point is within the polygon.

1-if the point is on the boundary of the polygon.

0-if the point is outside the polygon.

F.3.3 Methodology in brief

The algorithm uses a coordinate sorting mechanism to make use of the implicit knowledge contained in the ordering of the coordinates of the input polygons.

If both polygons have the same direction of description, (clockwise or anti-clockwise) and both input polygons include all the relevant points of intersection, then the points describing the vertices of the intersection polygon can be determined and ordered by selecting the correct sub-sets of coordinates from the input polygons.

From the above, the discrete steps of the algorithm can be enumerated below:-

1. Min/max test to discard any input polygons that obviously do not intersect.

2. Inside/outside test to discover and evaluate any situation in which one input polygon is completely inside the other.

3. Having discarded obvious situations; both input polygons are copied to preserve their contents unmodified. Future references to input polygons refer to these copies.

4. The coordinate list/array for each input polygon is modified, if required, so that the coordinate sequence describes the polygon in a clockwise manner.

5. Each line of one input polygon is checked against each line of the other input polygon for relevant intersection points. Parallel lines and lines failing min/max tests need not be tested for intersection.

6. Relevant points of intersection are added to both input polygons, in the correct list position.

7. Each input polygon is marked by an integer descriptor donoting the position of each coordinate in that polygon list/array. Coordinates that are present in both polygons are identically marked.

8. If no coordinate is found which is present in both input polygons, then the polygons do not intersect.

9. An output polygon is formed from the input polygons - the exact method being described fully in the next section.

F.3.4 The algorithm

The algorithm is described in terms of the steps enumerated in the previous section. To differentiate between input polygons - one will be called the BASE polygon and the other, the TEST polygon.

1. Min/max test

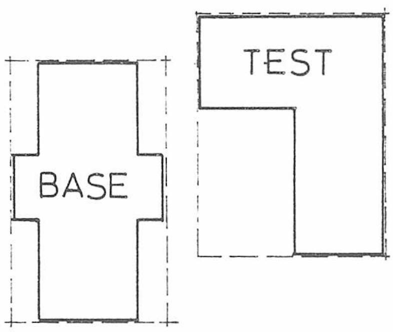


Fig.F.3 Min/Max test

Enclosing rectangles can be drawn round both polygons. If the enclosing rectangles do not intersect, then the polygons do not intersect.

In Fig.F.3 across, the polygons are rejected because:-

TEST min.X >= BASE max.X

2. Inside/outside test

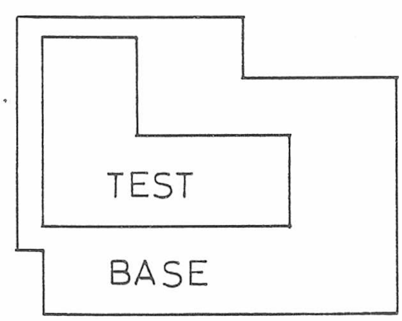


Fig.F.4 Inside - Outside test 1

If all the vertices of one polygon are inside the other polygon and all the vertices of the other polygon are outside the first polygon, then the first polygon is completely inside the second.

The polygons above (Fig.F.4) would pass this test and the output polygon description would be set to that of the TEST polygon description with no further calculation required.

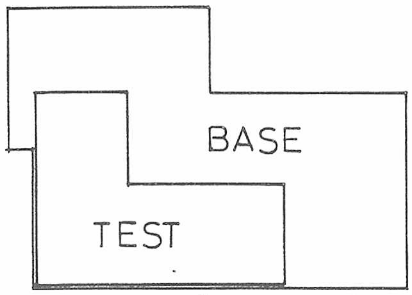


Fig.F.5 Inside - Outside test 2

The polygons in the diagram opposite (Fig.F.5) would fail the test, even although one polygon is inside the other, because they have two common edges.

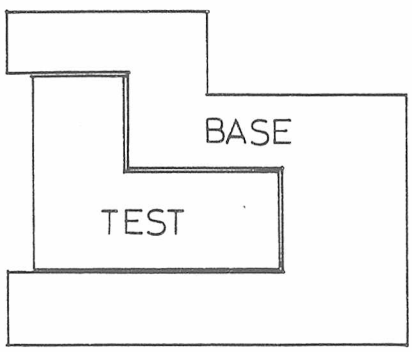


Fig.F.6 Inside - Outside test 3

The reasoning behind this becomes apparent when considering diagram (Fig.F.6) opposite. If (Fig.F.5) passed the test then so would (Fig.F.6) when it obviously shouldn't.

3. Input polygons copied

Each polygon description is copied into a temporary x,y coordinate list.

4. Sense checking

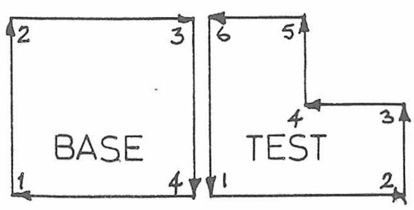
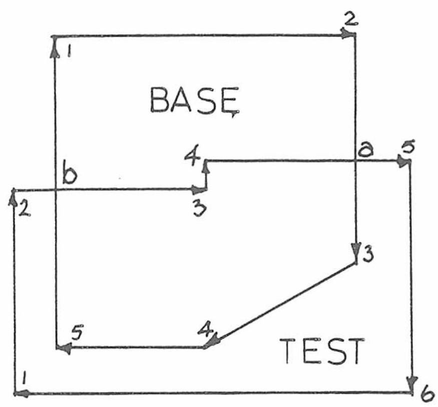


Fig.F.7 Sense checking

Using the area algorithm outlined in section F.3.2 both input polygons are checked for direction of description. Anti-clockwise polygons are re-ordered in a clockwise manner.

In this example the BASE object would be left as is; whilst the TEST polygon lists would be changed from 1,2,3,4,5,6 to 6,5,4,3,2,1.

5. Intersection discovery



BASE listing changes from
1,2,3,4,5 to 1,2,a,3,4,5,b
TEST listing changes from
1,2,3,4,5,6 to 1,2,b,3,4,a,5,6

Fig.F.8 Intersection discovery

In this part of the algorithm, the necessary 'new' information is calculated by checking each line in one polygon against each line in the other.

An intersection point is valid if the BASE line intersects the TEST line within both the BASE and the TEST lines' length, but not if the intersection point is either of the BASE line's end points.

The algorithm requires the role of BASE and TEST polygons to be transposed to uncover all the coordinates of intersection for each polygon. Notice that it is possible for each of the input polygons to acquire different additional intersection points, for instance, where an intersection point for one polygon is congruent with one of the vertices of the other polygon.

6. Polygon marking

The algorithm now marks each of the vertices of both input

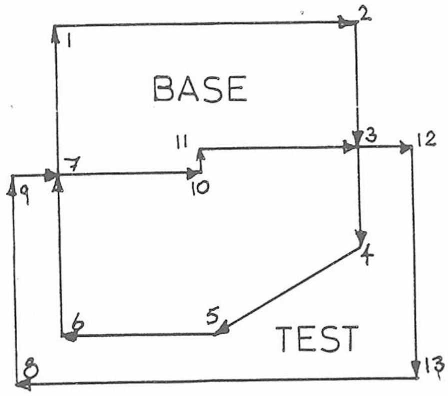


Fig.F.9 Polygon marking

polygons, by creating a reference list for each polygon.

This allows the reference list for each polygon to be manipulated rather than the actual polygon coordinates.

The diagram for the previous example would acquire the following marked lists:-

BASE 1,2,3,4,5,6,7

TEST 8,9,7,10,11,3,12,13

7. No interaction test

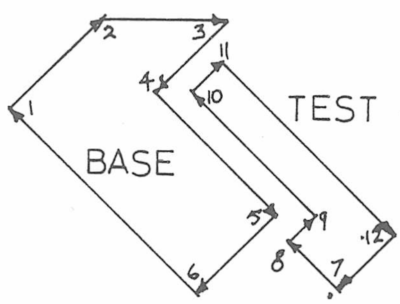
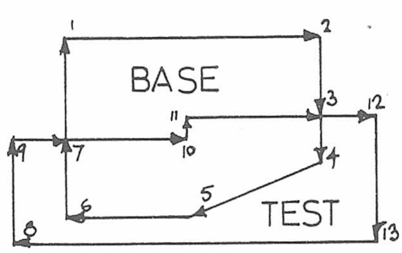


Fig.F.10 No interaction test

If the marked BASE and TEST lists have no number in common, then the polygons do not intersect. To reach this stage, they must of course have passed the min-max test.

8. Output polygon assembly



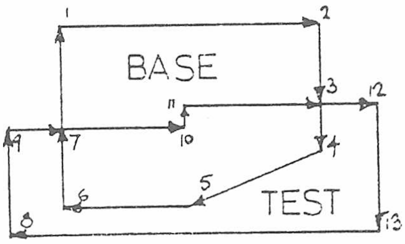
From the figure

BASE 1,2,3,4,5,6,7

TEST 8,9,7,10,11,3,12,13

Fig.F.11 Output polygon assembly - 1

The object of this stage is to assemble a marked list of the output polygon. This is then used to assemble the coordinates of the output polygon.

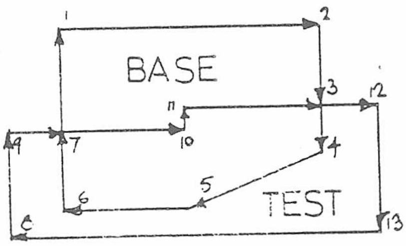


BASE 3,4,5,6,7

TEST 7,10,11,3

Fig.F.12 Output polygon assembly - 2

1. FIRST STEP. The first step is to discard from the marked lists any coordinate that is not inside or on the other polygon, or is not adjacent to an area common to both input polygons.

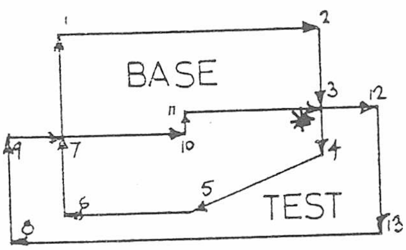


BASE 3 4,5,6,7,Last

TEST 3,7,10,11,Last

Fig.F.13 Output polygon assembly - 3

2. SECOND STEP. The second step is to re-order both marked lists such that the lowest common number is at the head of the list. Also the end of each list is marked.



From the figure

BASE 4,5,6,7,L,3

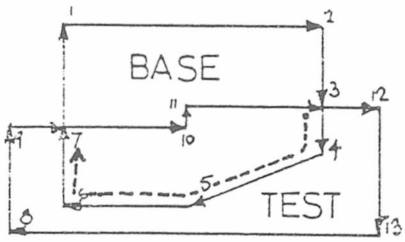
TEST 7,10,11,L,3

OUTPUT 3

Fig F.14 Output polygon assembly - 4

3. THIRD STEP. The third step is to assemble the output marked list.

If the head of both BASE and TEST lists are common, then the output number is that common number. Both BASE and TEST lists are rotated forwards by one element.



From the figure

BASE L,3,4,5,6,7

TEST 10,11,L,3,7

OUTPUT 3,4,5,6,7

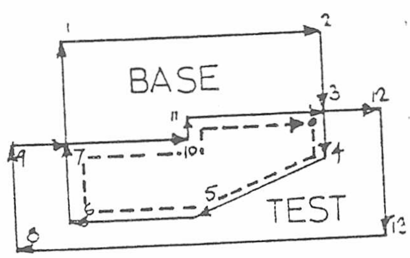
Fig.F.15 Output polygon assembly - 5

Considering the BASE list, if the number at the head of that list is not contained in the TEST list then that number is added to the output list and the BASE list is rotated forward by one element.

If the number at the head of the BASE list is contained within the TEST list, then the number at the head of the TEST list is added to the output list and the TEST list is rotated by one element.

If the last number has been reached in the BASE list, then the remainder of the TEST list is added to the end of the output list.

If the last number has been reached in the TEST list, then the remainder of the BASE list is added to the output list.



From the figure

BASE L,3,4,5,6,7

TEST L,3,7,10,11

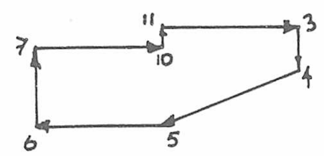
OUTPUT 3,4,5,6,7,10,11

Fig.F.16 Output polygon assembly - 6

4. FOURTH STEP. The fourth step is to assemble the actual coordinates from the marked output list.

5. LAST STEP. The last step is to use the area algorithm to check the sense of the output polygon. If the output polygon has a clockwise sense it is valid. If it has zero area or an anti-clockwise sense it is not a valid polygon.

Fig.F.17 Output polygon assembly - 7



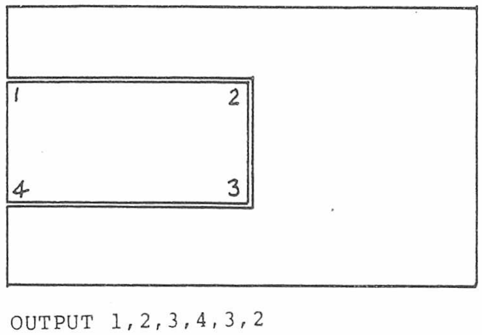
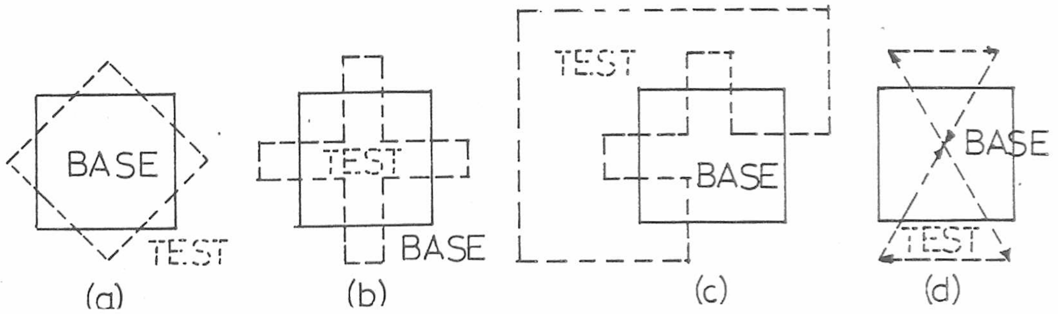


Fig.F.18 Output polygon assembly - 8

F.3.5 Last remarks

The algorithm can return the polygon that describes the bounding edges of the overlap area of all non-self-intersecting polygons. ie. it will return the correct polygon for the classes of shapes shown below in Fig.F.19.

Fig.F.19 Successful calculation of area of overlap



However, it will almost certainly get the shape below incorrect, because BASE is self intersecting. A preliminary test to resolve self-intersecting polygons into non-self-intersecting polygons could be devised relatively easily. This was not done by the author since it was not required for the problem application to which the software is addressed.

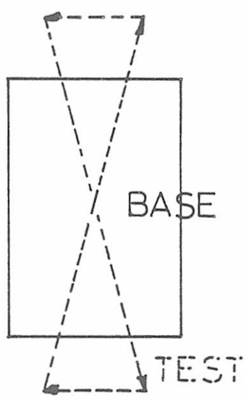
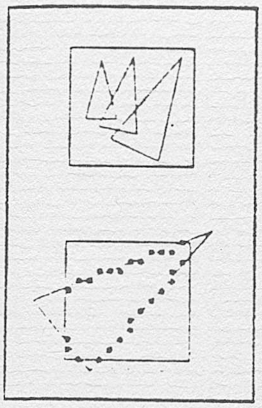


Fig.F.20 Unsuccessful calculation of area of overlap

F.4.0 REFERENCES

(F.1) Scrivener, S. 'How to handle an image', Design Studies, Vol.14, No.1, Jan.1983, pp.35-40, Butterworth Scientific Ltd., in cooperation with the Design Research Society.

(F.2) Sutherland, I.E, Sproull, R.F, Schumaker, R.A, 'A Characterisation of Ten Hidden-Surface Algorithms', Computing Surveys, The Survey and Tutorial Journal of the ACM, Vol.6, No.1, March 1974, pp.1-45, Association for Computing Machinery.



APPENDIX G

GRAPHS, TABLES AND MISCELLANEOUS INFORMATION PERTAINING TO THE EVALUATIVE EXPERIMENTS

CONTENTS

G.1.0 INTRODUCTION

This appendix forms a supplement to chapter six, and as such it should be read in conjunction with that chapter. It consists of material excluded from the main body of the text for brevity and clarity.

As far as possible, section numbers match the appropriate section in chapter six.

G.2.0 EXPERIMENT 1

This section contains addenda to the first experiment.

G.2.1 Questionnaire design

Below is a list of the names and addresses of the kitchen designers contacted with a request for assistance.

Aberdeen Kitchen
Design Ltd.,
Summerhill Court,
Summerhill Road,
ABERDEEN

+ Albyn Kitchens
14a, Bon Accord Crescent,
ABERDEEN

* Continental Kitchens
of Perth,
3, King Street,
PERTH

* Elgin Kitchen Centre
164a, High Street,
ELGIN

+ Ensign Kitchen
(Installations) Ltd.,
24, Market Street,
ABERDEEN

Home Decor Centre,
Castle Street,
TURRIFF

* Fraserburgh Kitchen
Centre,
14, Cross Street,
FRASERBURGH

+ Colin Hunter Kitchens
Devanna House,
Riverside Drive,
ABERDEEN

Grampian Kitchens
24, Carmelite Street,
BANFF

+ Kitchen Installation,
44, Craigpark Place,
ELLON

+ James Chivas,
93, Victoria Road,
Torry,
ABERDEEN

++ Rational Kitchen Studio,
The Village,
John Street,
ABERDEEN

+ Deeside Design Kitchens
and Bathrooms,
Aboyne Business Centre,
ABOYNE

Ellington Forbes Kitchen
Studio,
20, Great Western Road,
ABERDEEN

Lager Kitchens
(Scotland) Ltd.,
20, Greenhole Place,
Bridge of Don,
ABERDEEN

* First Studio
281, Rosemount Place,
ABERDEEN

+ Buchan Schreiber
Furniture Centre,
Union Street,
ELLON

Gordon Forbes
Kitchen Studio,
Edgar Road,
ELGIN

** In-toto Ltd.,
92, Rosemount Place,
ABERDEEN

+ Ski Specialised
Kitchen Installation,
25, Balgownie Place,
Bridge of Don,
ABERDEEN

* Shirras Laing Ltd.,
46/52, Schoolhill,
ABERDEEN

Victoria Kitchens,
The Village,
John Street,
ABERDEEN

** The Kitchen Studio,
123, Rosemount Place,
ABERDEEN

* Upstairs Downstairs
Kitchen Design Studio,
82, Walker Road,
Torry,
ABERDEEN

Valiso Ltd.,
61, Constitution Street,
ABERDEEN

+ Paterson Oldmeldrum,
Eavern,
Colpy Road,
OLDMELDRUM

** School of Home Economics,
RGIT
Queens Road,
ABERDEEN

*	returned one or more kitchen designs.
**	returned one or more kitchen designs and received one or more interviews.
+	unobtainable or ceased trading.
++	same business as Victoria Kitchens - trading under a different name due to two dealerships.

Below is a copy of the questionaire issued as part of the first experiment.

KITCHEN LAYOUT SURVEY/QUESTIONNAIRE

Your assistance is requested with this questionnaire. There is less to it than the number of pages imply. If you are a student, rest assured that the questionnaire forms no part of your coursework and will not be 'assessed' by any member of staff.

If you've got down here, you must have volunteered - so thanks for the help!

What you should do

A selection of 21 domestic kitchen layouts are presented to you in the following pages. Using the page of answer boxes provided overleaf, please rank the layouts from 1st to 21st, for the following two criteria.

(i) firstly, for the smallest room where 1 is the smallest room and 21 is the largest room. Please do this by eye - do not use a ruler or any other measuring device.

(ii) secondly, for the room where best use is made of available space, (bearing in mind it is possible to overfill a room with units as well as to underprovide) where 1 is the room that makes best use of available space, and 21 is the worst.

Note: If you find it impossible to discriminate between two or more rooms, mark them as equal. For example if you believe Room H and Room P are equally small, and smaller than all others mark them both as 1st=; the next smallest room would then be 3rd.

Finally: On the last sheet you are asked to comment on how you decided a kitchen made better use of available space than another.

For your information only

(a) Each room layout is to the same space (approx 1:45 ; this odd scale is due to the vagaries of reproduction).

(b) All windows are set with a sill height of 1050 mm and a head height of 2100 mm. Doors are full length and have a head height of 2100 mm.

(c) Most kitchen appliances (or their housings) are 600 x 600 by 900 mm high.

(d) Exceptions are fridge freezers (F/F), Oven housings (O) and tall cupboards (600 x 600, marked by a diagonal line) which are 2000 mm high.

(e) Most work top covered base units are 600 x 600, though some rooms also have 600 x 500 and 600 x 300 units - these are easily recognised on plan. All have a height of 900 mm.

(f) High level units (marked by a diagonal line running through them) are generally 300 x 600, though again 300 x 500 and 300 x 300 units have been used in some rooms. All high level units are 650 mm high and are set at 1350 mm from ground level.

ANSWER BOXES

1. You are (delete as applicable) staff/student/other.

2. If answer to 1 is 'other' please describe ....

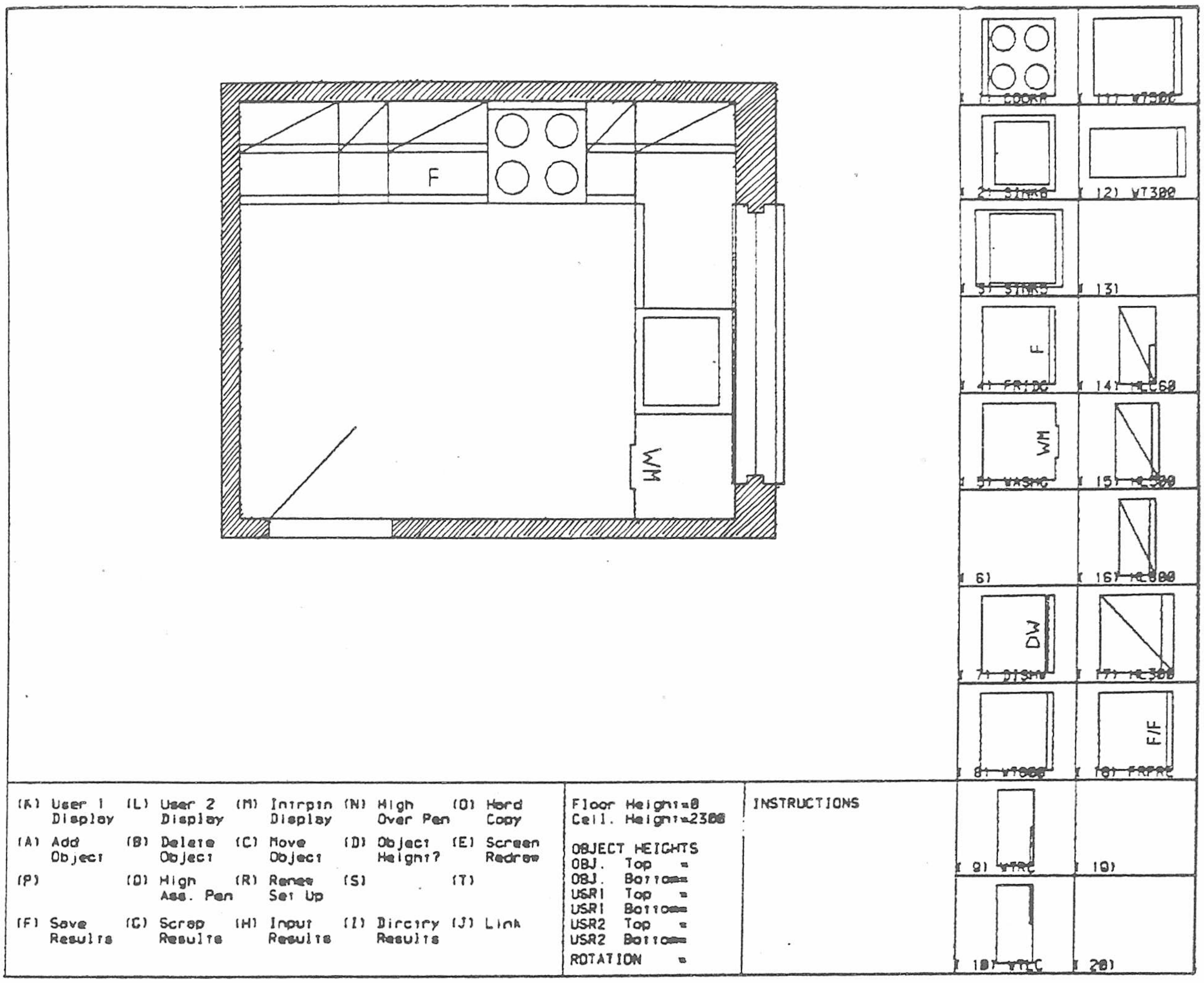
3. Rank rooms from 1 - 21 in the boxes provided below.

Name	Area	Space
Kitchen A		
Kitchen B		
Kitchen C		
Kitchen D		
Kitchen E		
Kitchen F		
Kitchen G		
Kitchen H		
Kitchen I		
Kitchen J		
Kitchen K		
Kitchen L		
Kitchen M		
Kitchen N		
Kitchen O		
Kitchen P		
Kitchen Q		
Kitchen R		
Kitchen S		
Kitchen T		
Kitchen U		

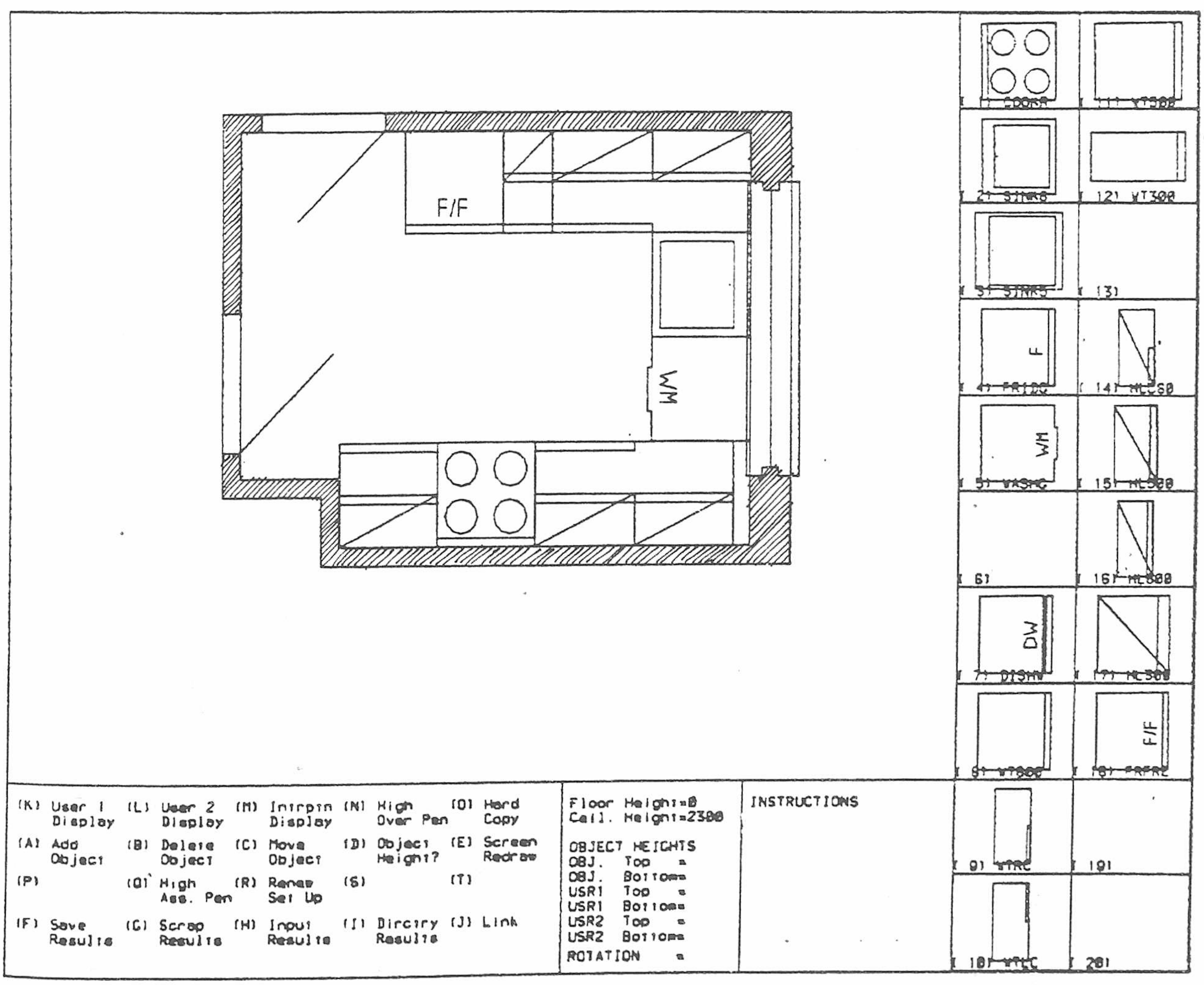
Reminder - for Area 1 = room with least floor area.
for Space 1 = room making best use of available space.

4. Don't forget to fill out comment section on last page.

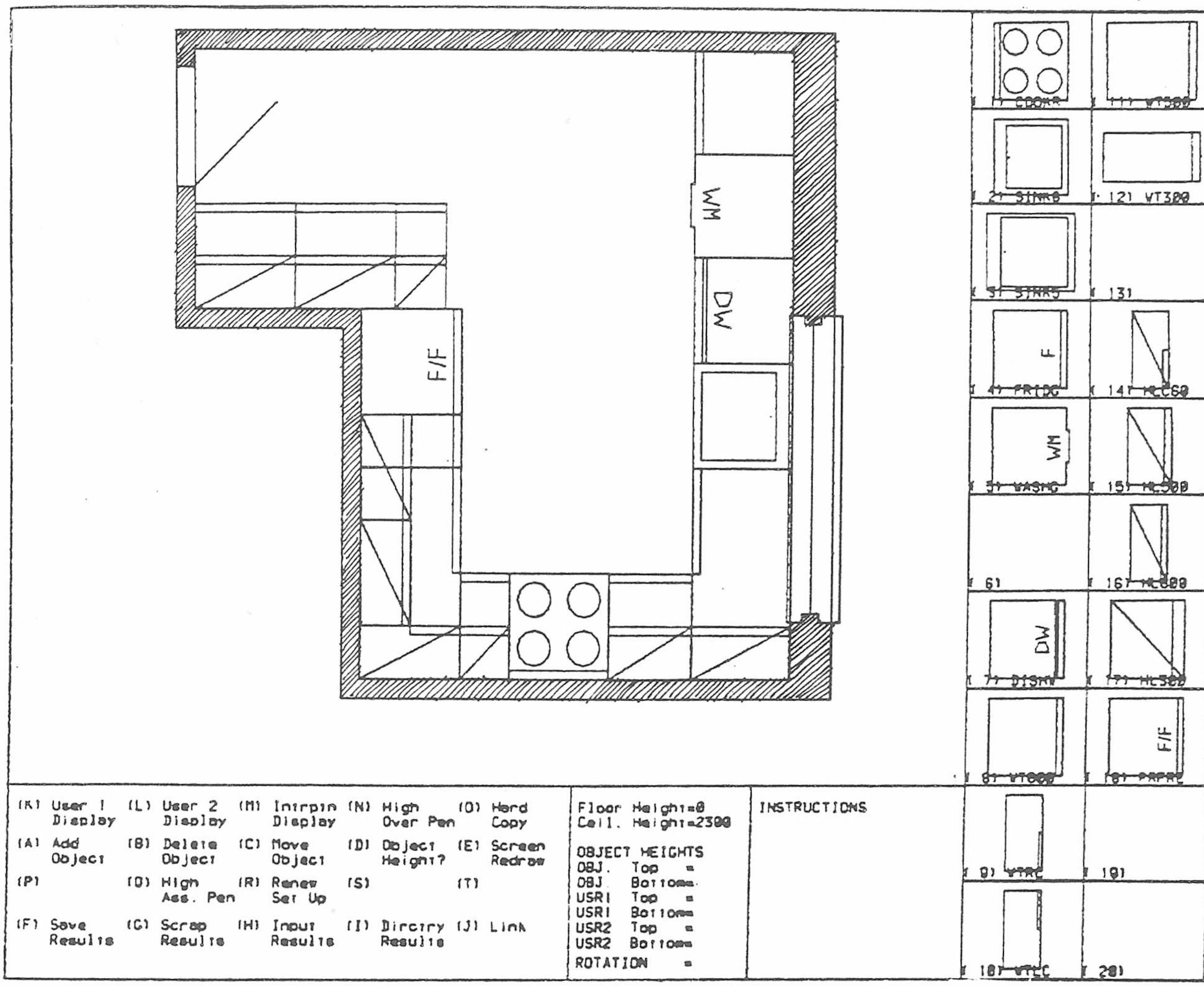
A


F
WM
F
WM
12) WT300
13)
6)
DW
F/F
9)
19)
20)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

B

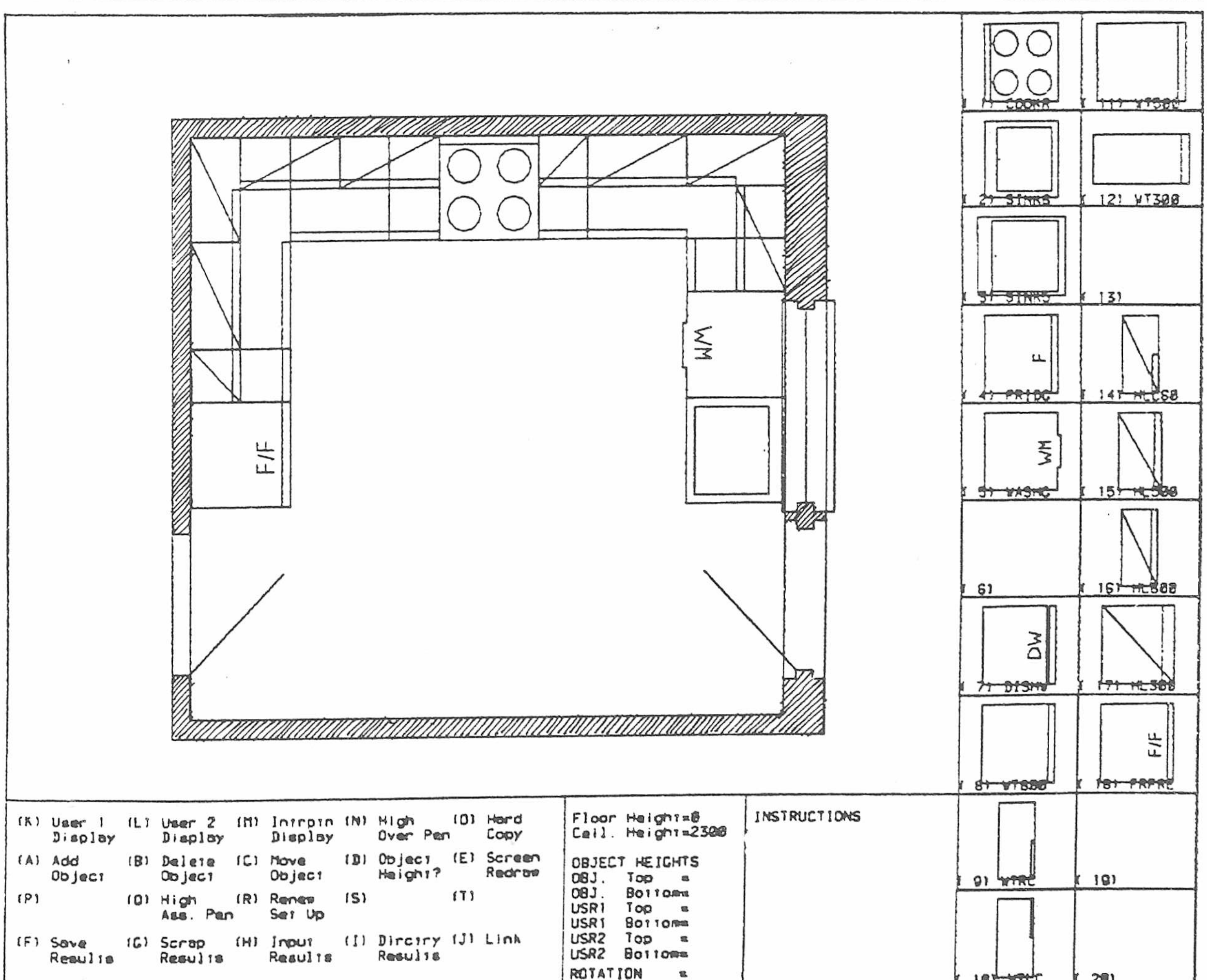

F/F
WM
F
WM
12) WT300
13)
6)
DW
F/F
9)
19)
20)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

C



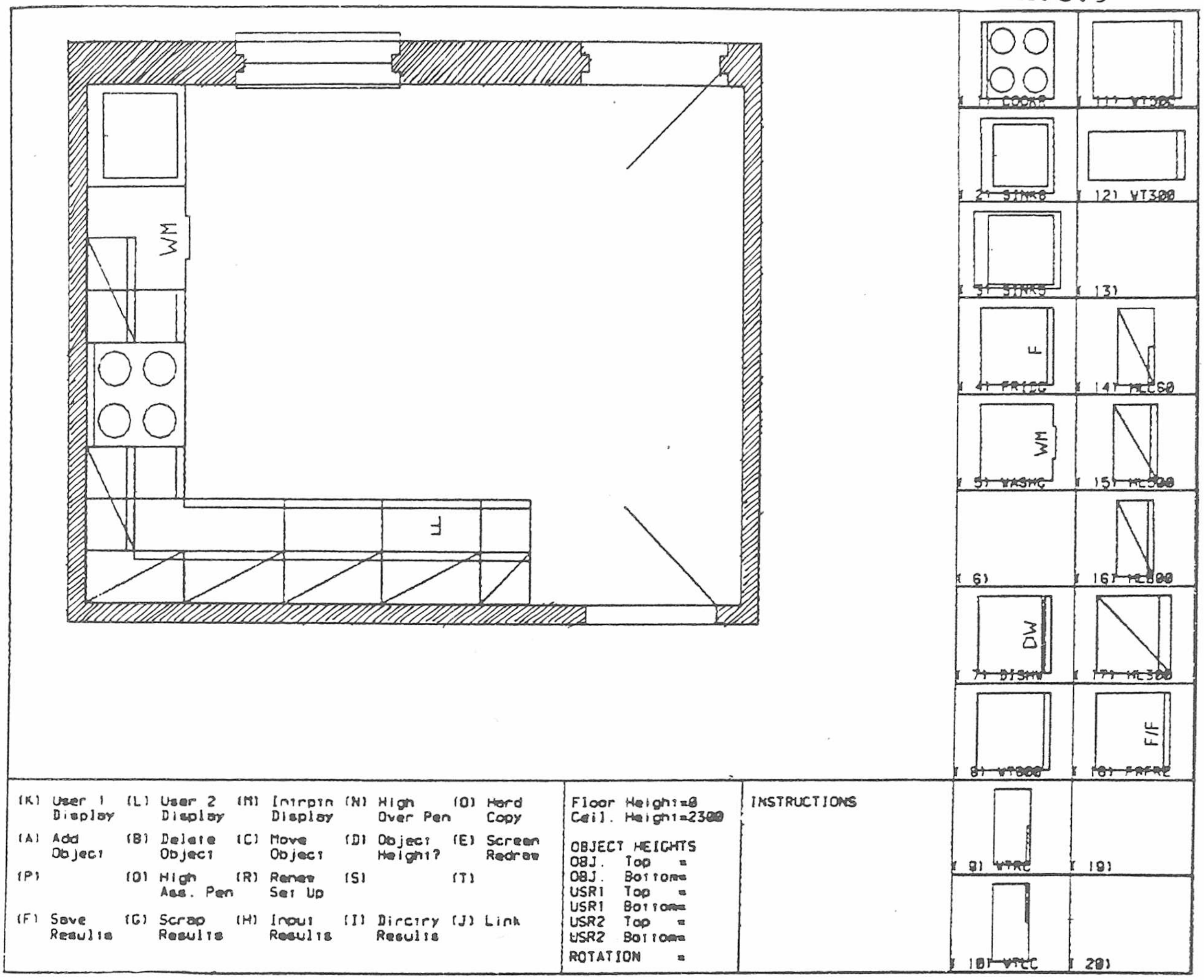
WM
DW
F/F
(K) User 1 Display (L) User 2 Display (M) Intrpin Display (N) High Over Pen (O) Hard Copy
(A) Add Object (B) Delete Object (C) Move Object (D) Object Height? (E) Screen Redraw
(P) (Q) High Ass. Pen (R) Renew Set Up (S) (T)
(F) Save Results (G) Scrap Results (H) Input Results (I) Dirctry Results (J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
12) VT300
13)
F
WM
6)
DW
F/F
19)
20)

D

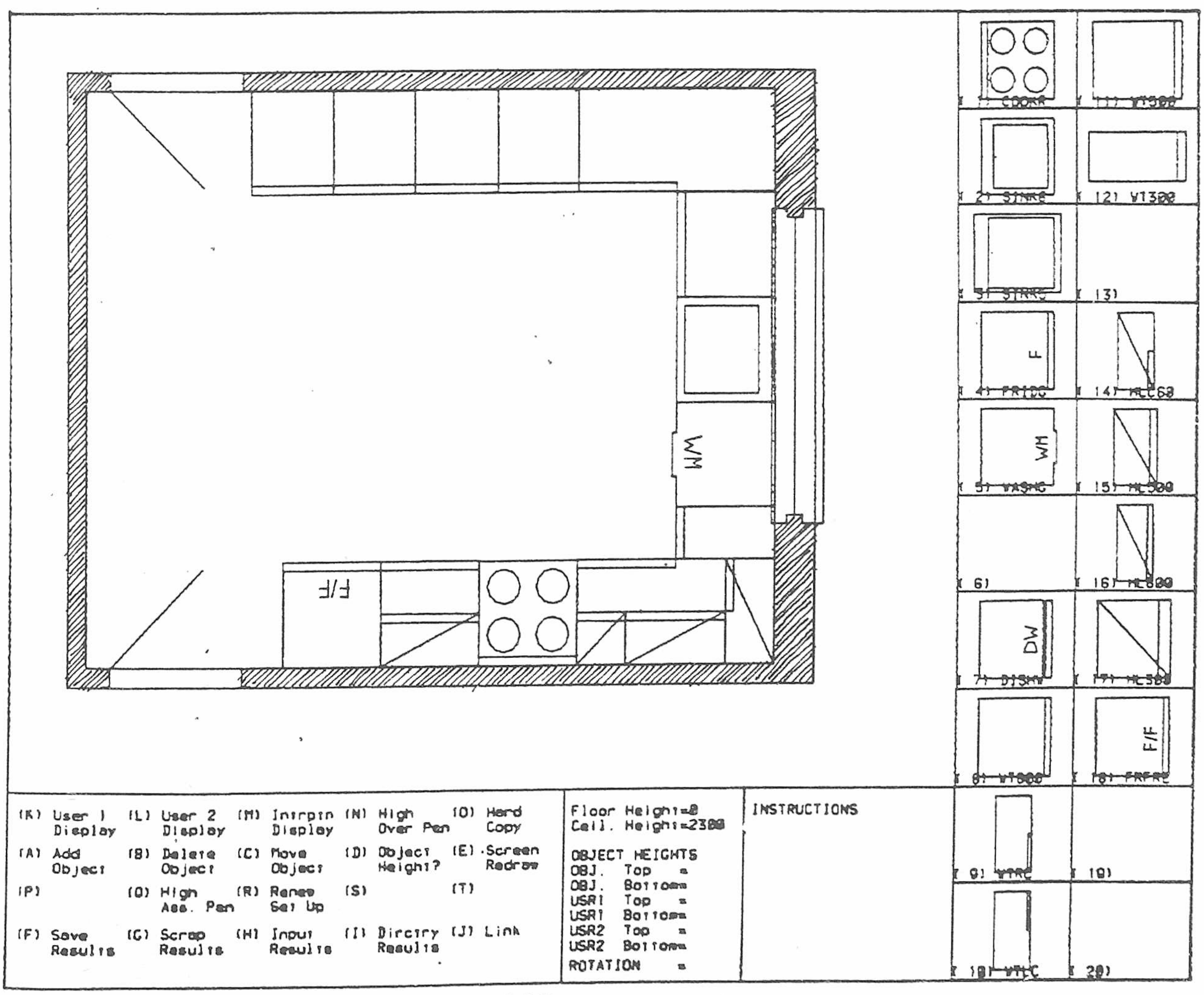


WM
F/F
(K) User 1 Display (L) User 2 Display (M) Intrpin Display (N) High Over Pen (O) Hard Copy
(A) Add Object (B) Delete Object (C) Move Object (D) Object Height? (E) Screen Redraw
(P) (Q) High Ass. Pen (R) Renew Set Up (S) (T)
(F) Save Results (G) Scrap Results (H) Input Results (I) Dirctry Results (J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
12) VT300
13)
F
WM
6)
DW
F/F
19)
20)

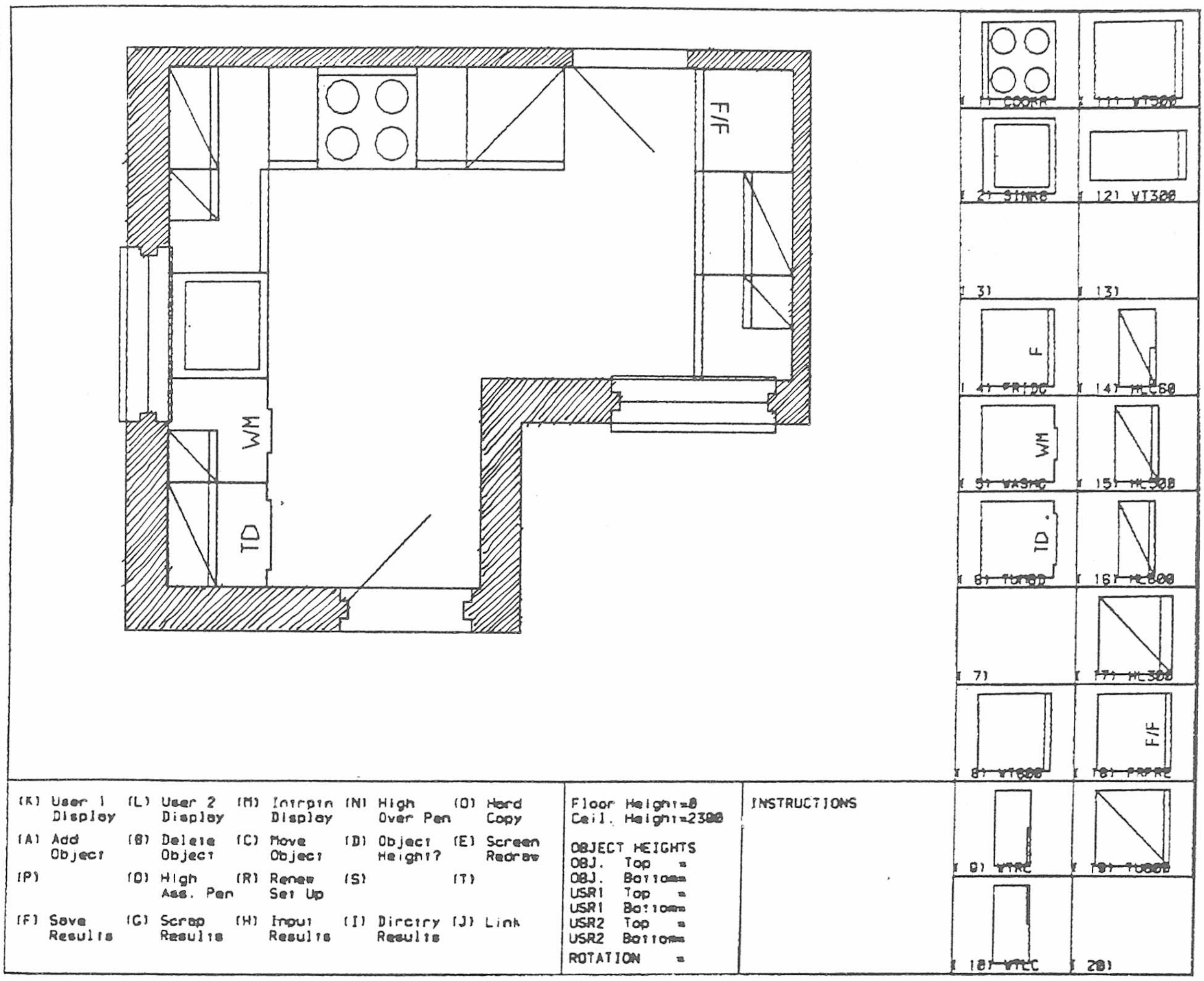
E


WM
F
DW
F/F
12) VT300
13)
6)
19)
20)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

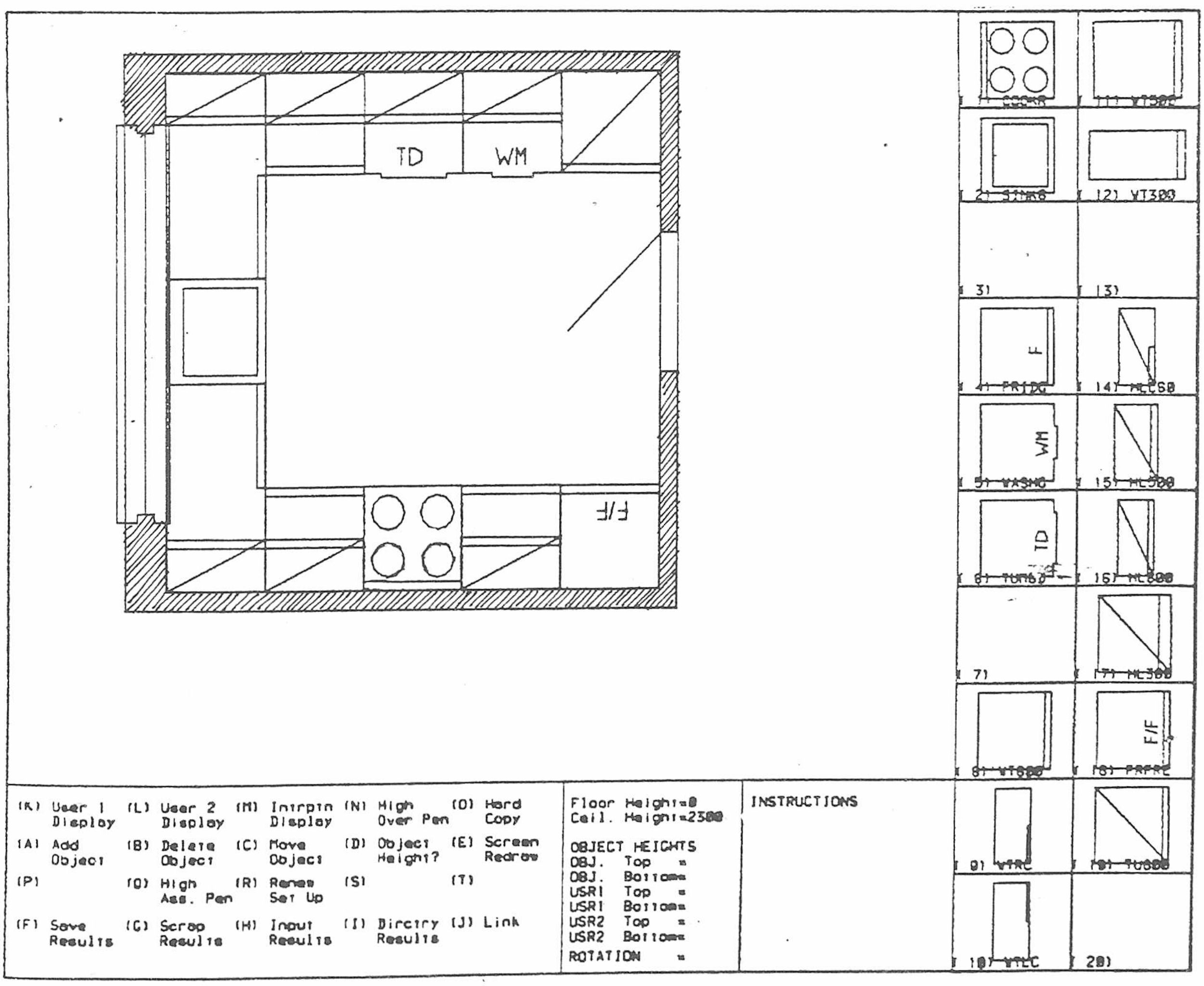
F


WM
F/F
F
DW
12) V1300
13)
6)
19)
20)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

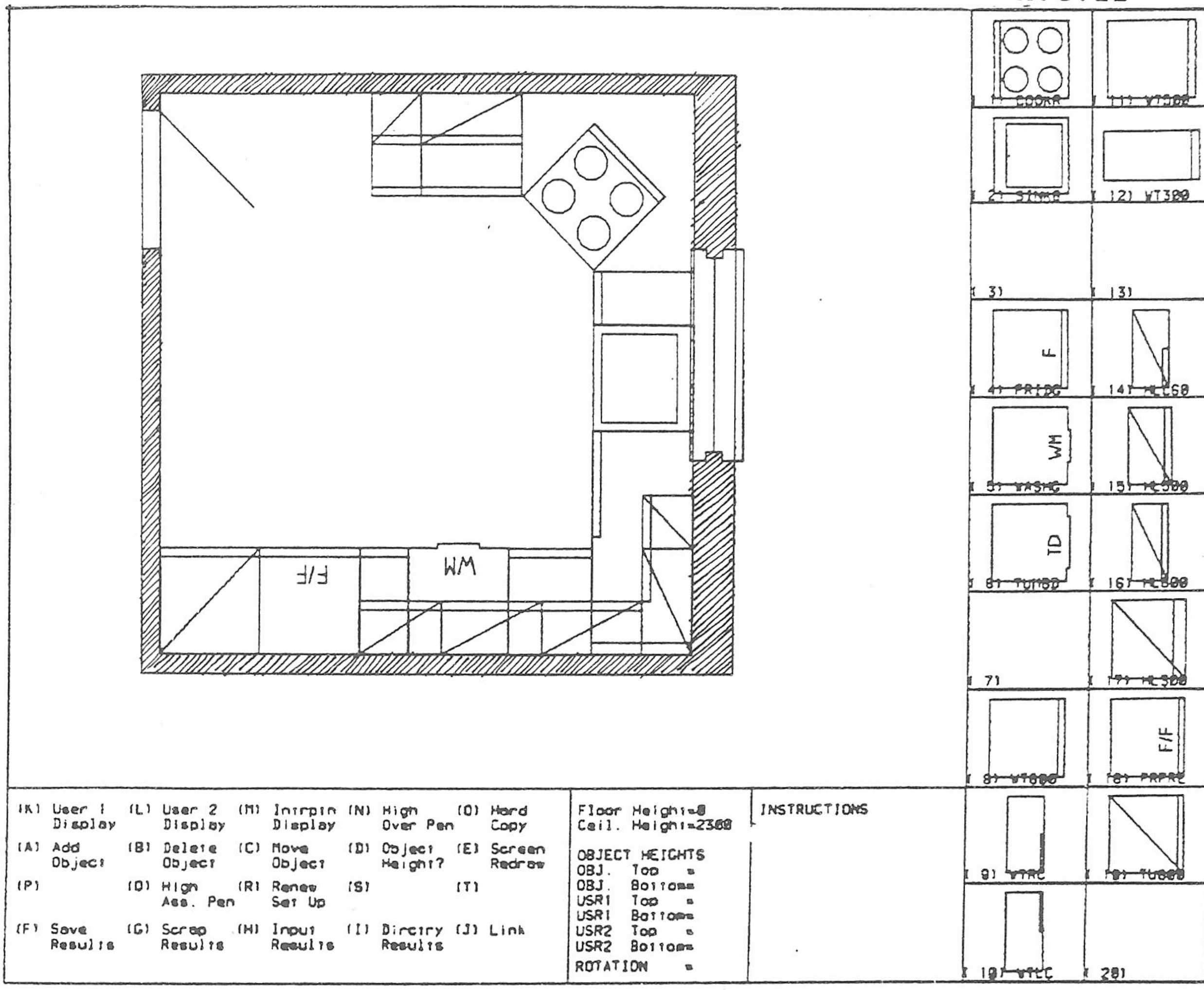
G


F/F
WM
TD
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
F
WM
TD
F/F
12) VT300
3)
13)
7)
20)

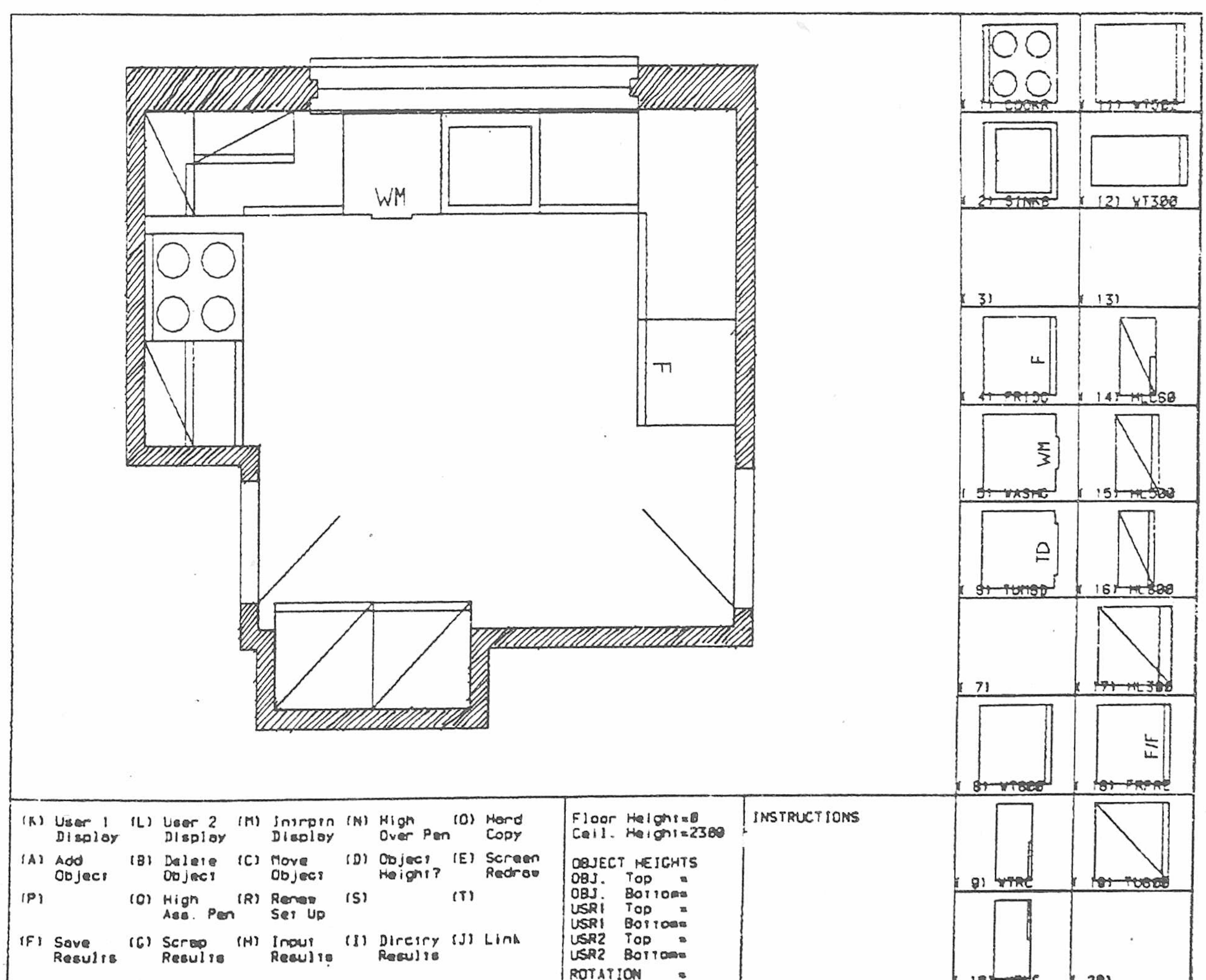
H


TD
WM
F/F
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
F
WM
TD
F/F
12) VT300
3)
13)
7)
20)

I

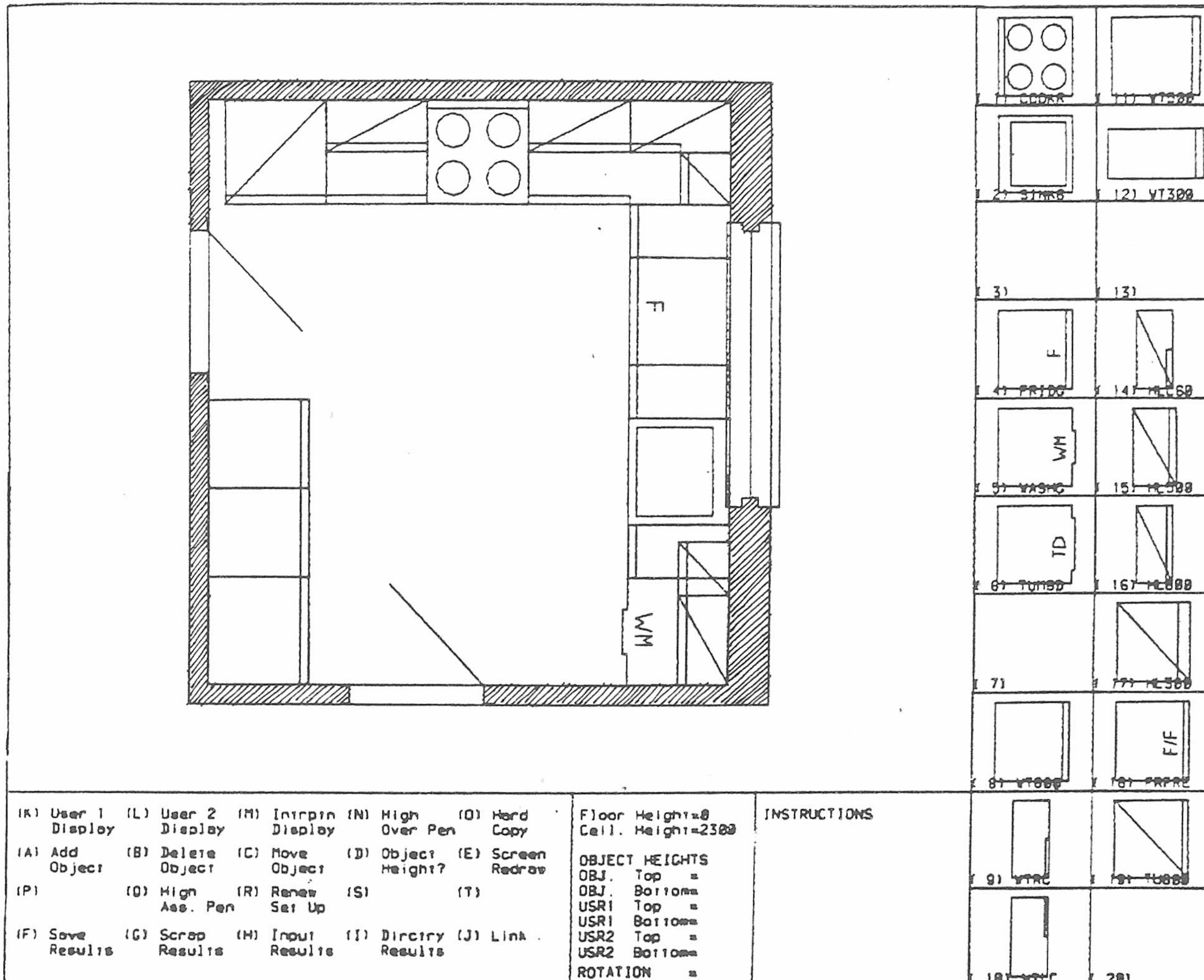

F/F
WM
1) COOKR
11) WT300
2) SINKR
12) WT300
3)
13)
F
4) FRIDG
14) HLC50
WM
5) WASHG
15) HL500
TD
6) TUMBD
16) HL600
7)
17) HL300
8) WT600
F/F
18) FRFRZ
9) WTRC
19) TU600
10) WTLC
20)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

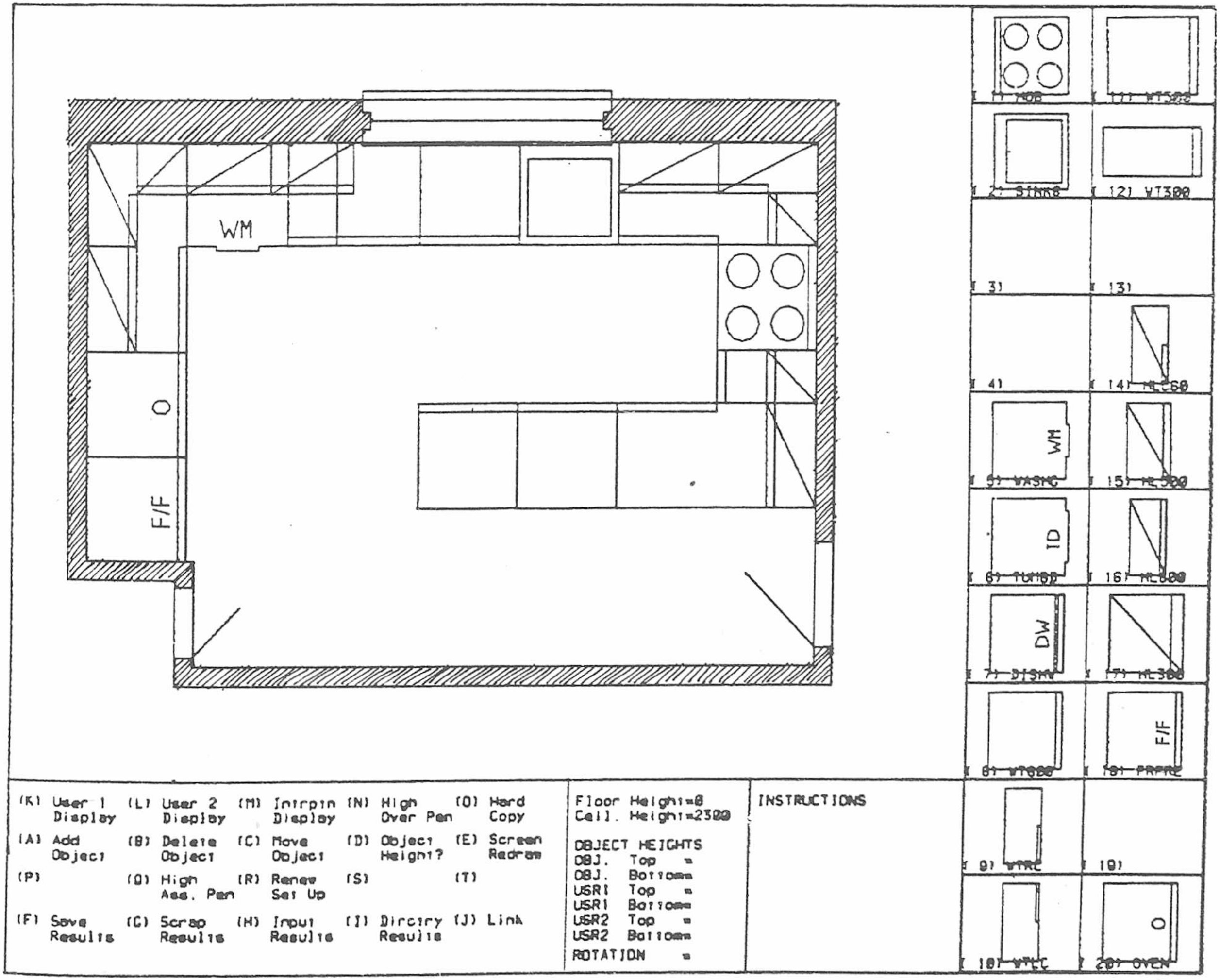
J


WM
F
1) COOKR
11) WT600
2) SINKR
12) WT300
3)
13)
F
4) FRIDG
14) HLC50
WM
5) WASHG
15) HL500
TD
6) TUMBD
16) HL600
7)
17) HL300
8) WT600
F/F
18) FRFRZ
9) WTRC
19) TU600
10) WTLC
20)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

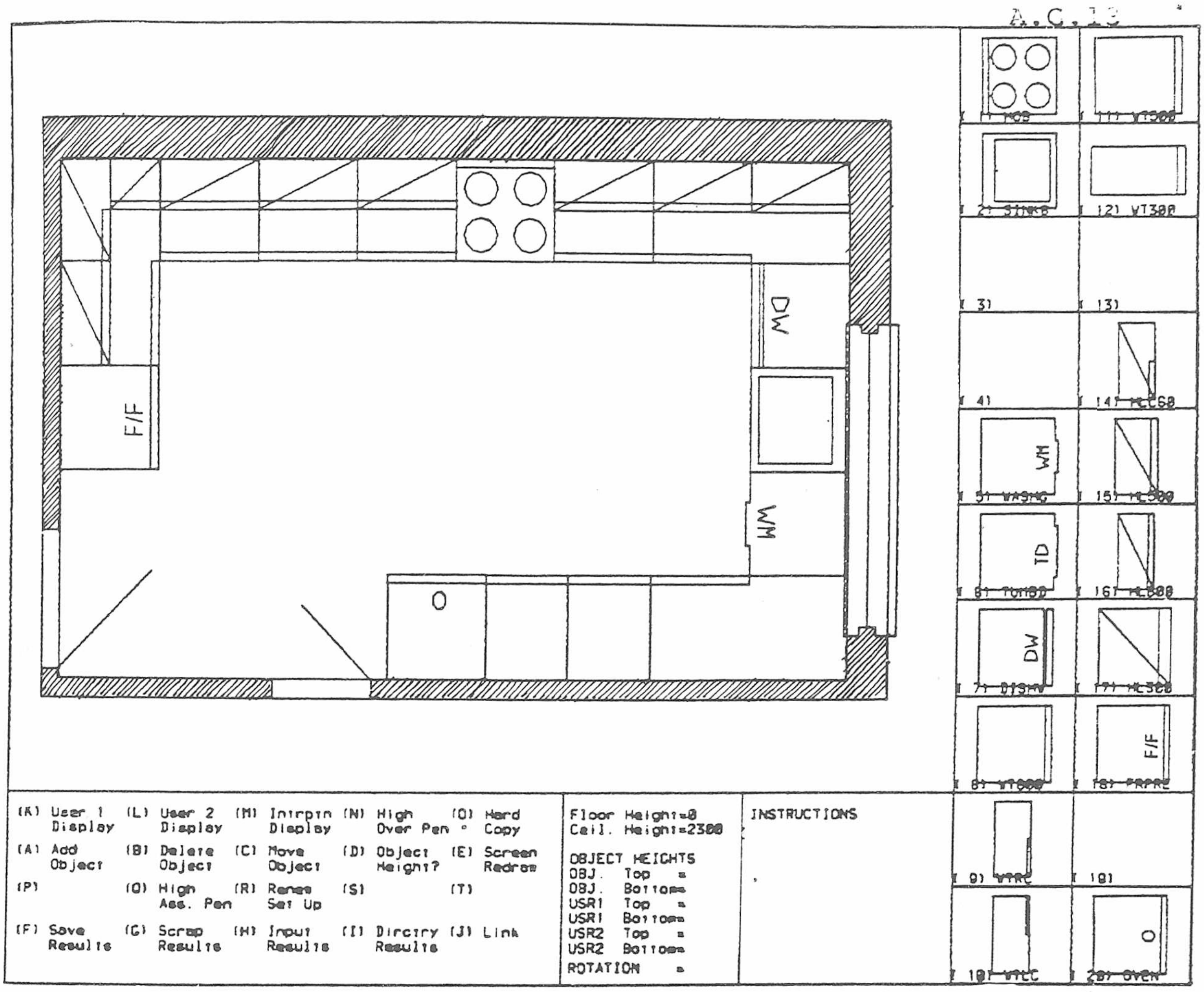
A.G.12

K

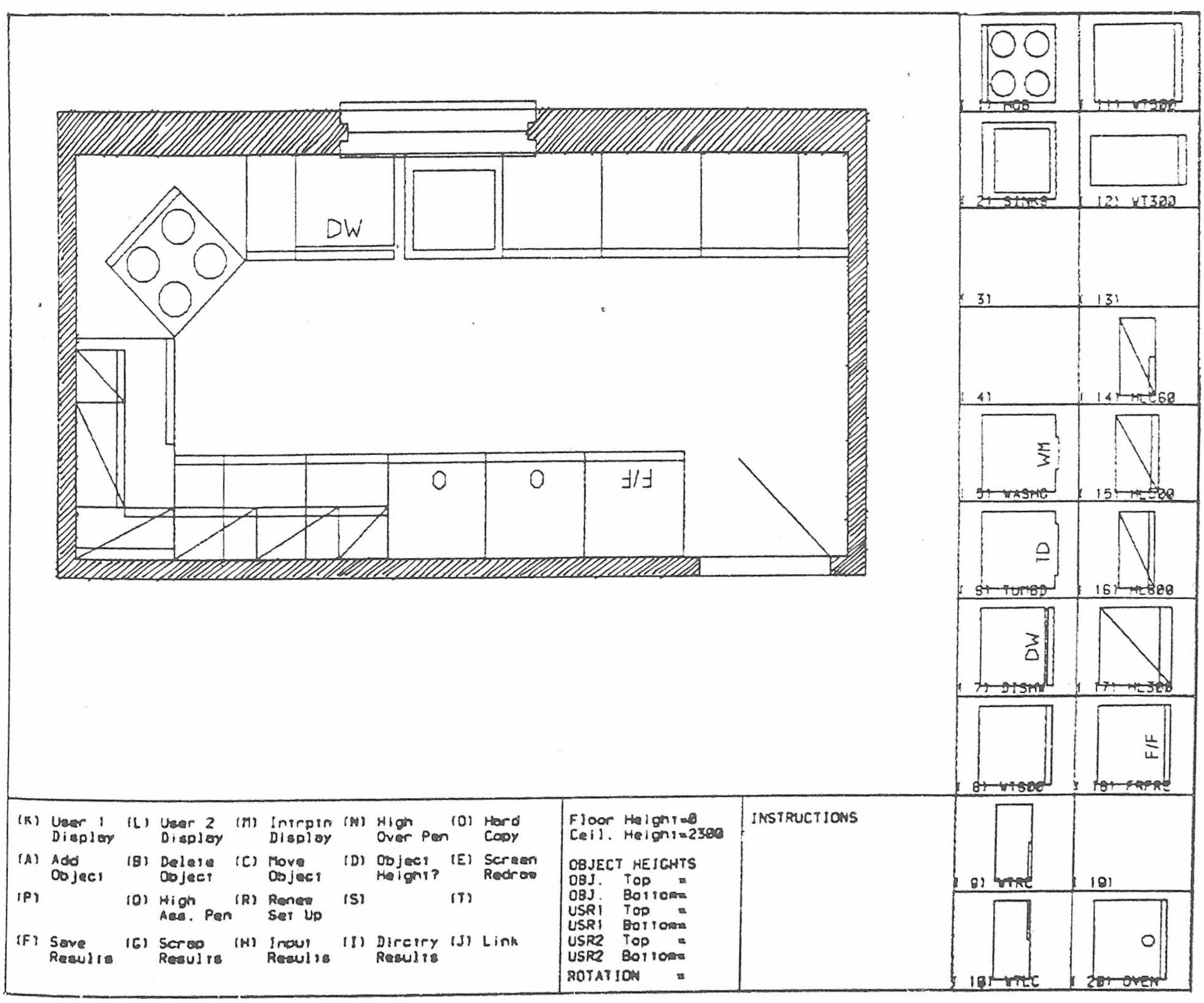

F
WM
F
WM
ID
F/F
12) VT300
3)
13)
7)
28)
(K) User 1 Display
(L) User 2 Display
(M) Intrpln Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS


WM
O
F/F
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
3)
13)
4)
WM
ID
DW
F/F
12) VT300
19)
O

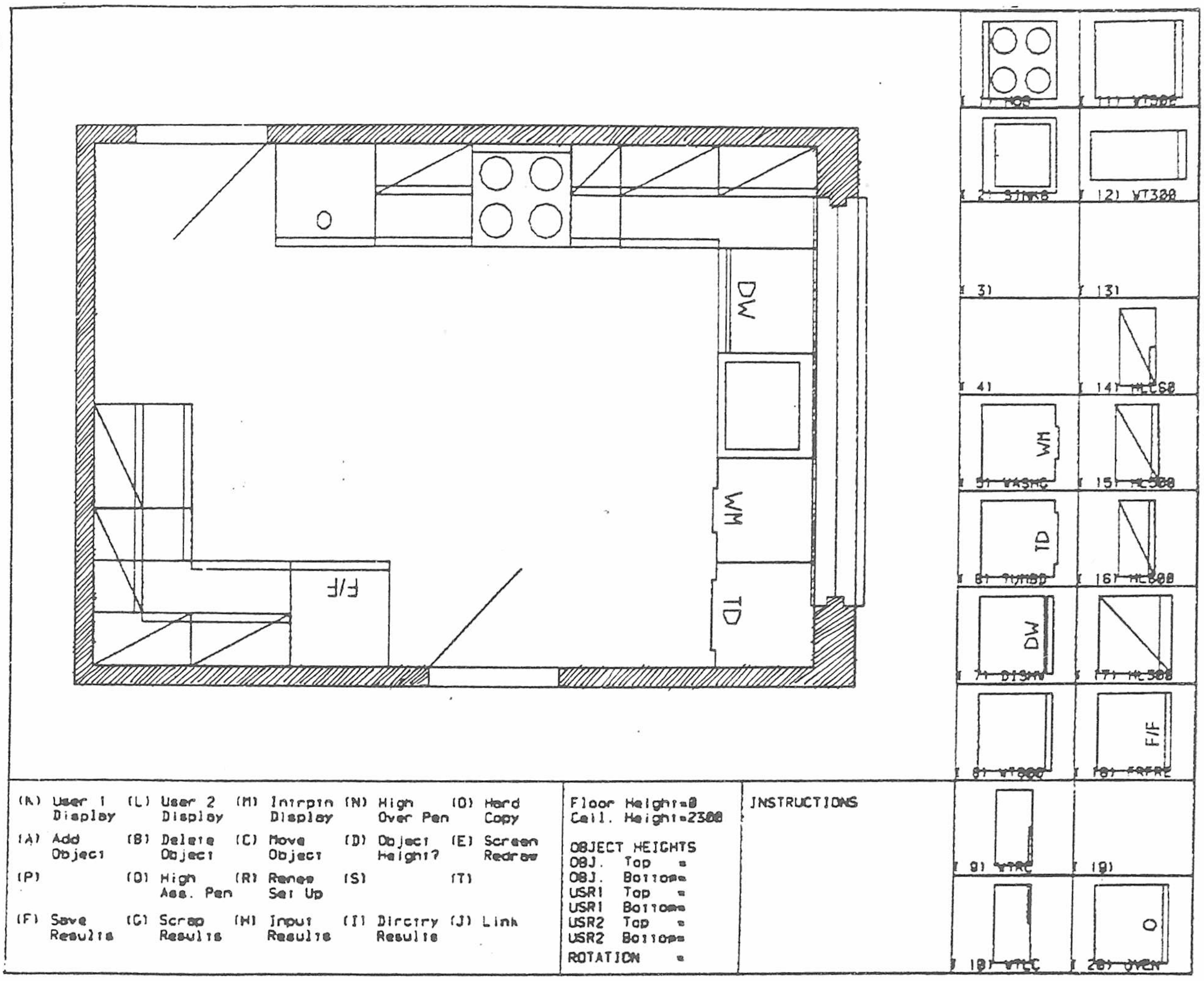
M


F/F
DW
WM
O
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Abs. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
WM
ID
DW
F/F
O
12) VT300

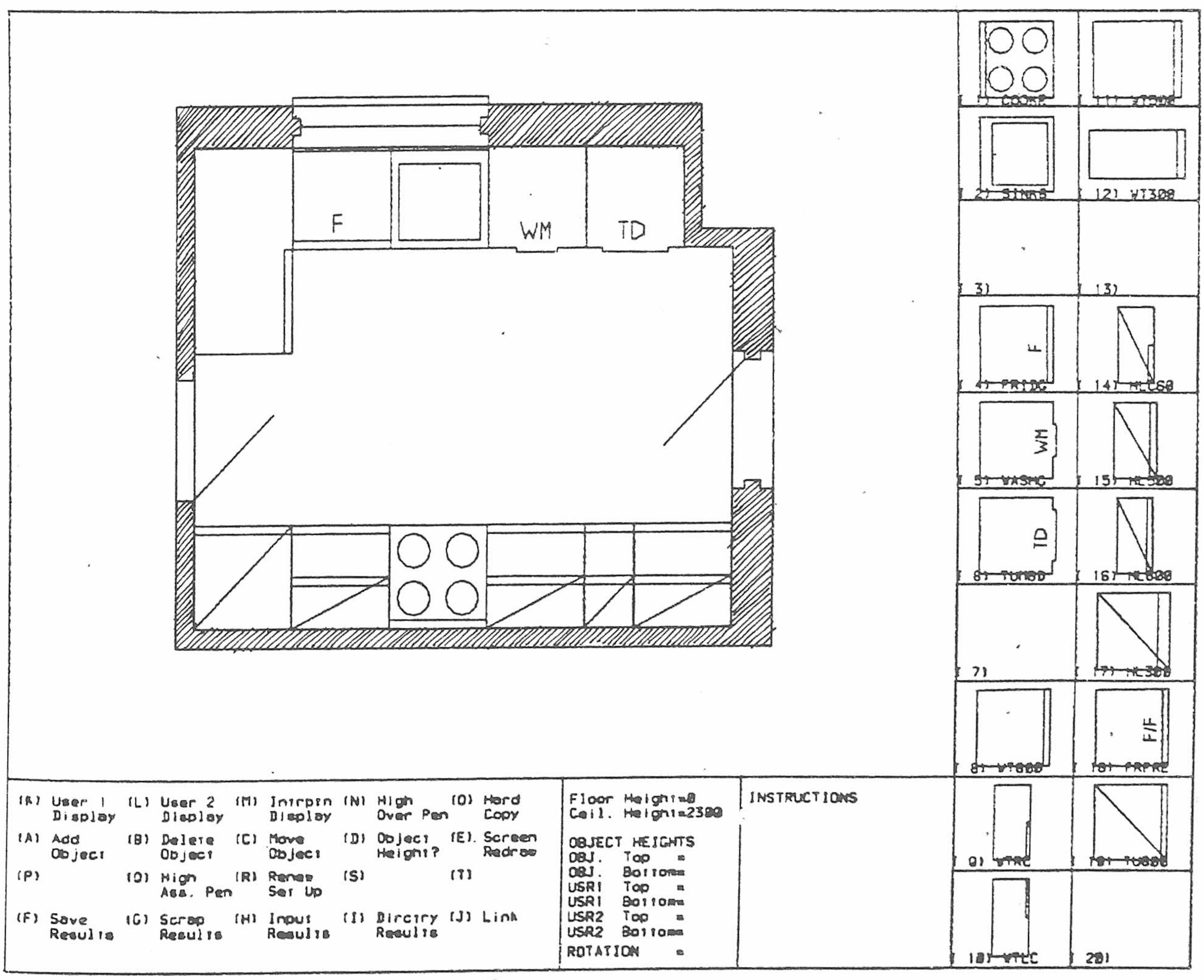
N


DW
O
O
F/F
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Abs. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
WM
ID
DW
F/F
O
12) VT300

O

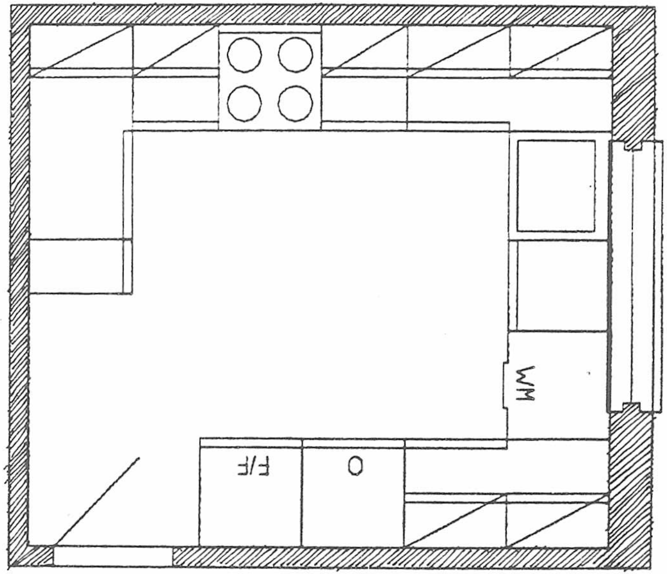

F/F
DW
WM
TD
1) HOB
11) VT50F
2) SINK8
12) VT300
3)
13)
4)
14) HLC60
5) WASHC
15) HL500
6) TUMBD
16) HL600
7) DISHW
17) HL300
8) VT600
18) FRFRE
9) VTRC
19)
10) VTLC
20) OVEN
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

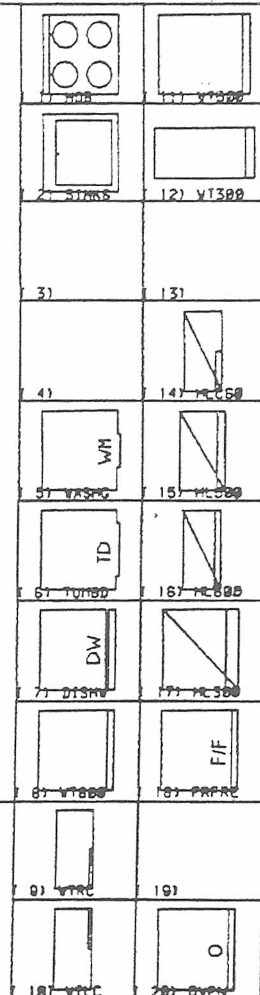
P


F
WM
TD
1) COOKP
11) VT50F
2) SINK8
12) VT300
3)
13)
4) FRIDG
14) HLC60
5) WASHC
15) HL500
6) TUMBD
16) HL600
7)
17) HL300
8) VT600
18) FRFRE
9) VTRC
19) TU600
10) VTLC
20)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

A.G.15

Q





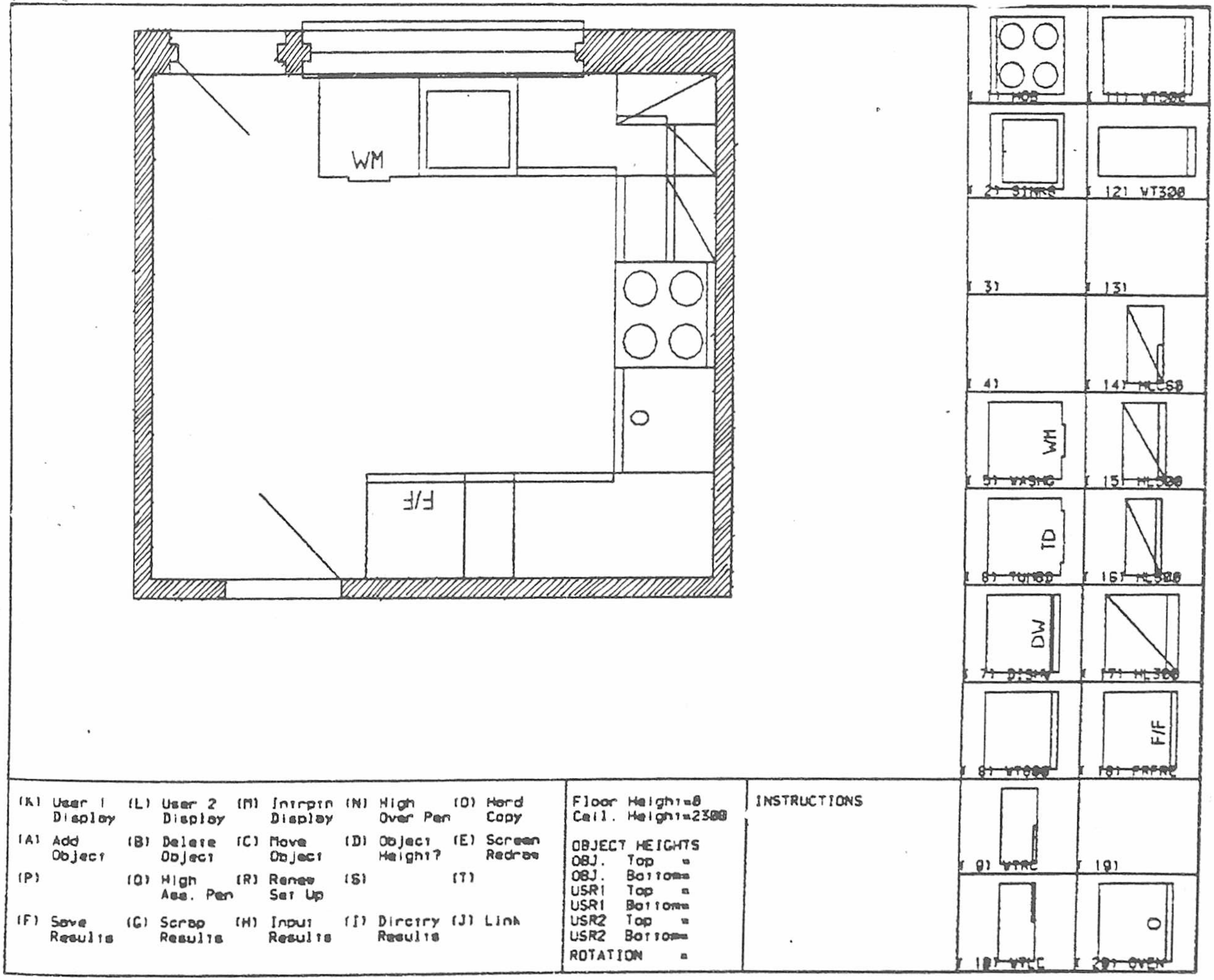
(K) User 1 Display	(L) User 2 Display	(M) Intrptn Display	(N) High Over Pen	(O) Hard Copy
(A) Add Object	(B) Delete Object	(C) Move Object	(D) Object Height?	(E) Screen Redraw
(P)	(Q) High Ass. Pen	(R) Renew Set Up	(S)	(T)
(F) Save Results	(G) Scrap Results	(H) Input Results	(I) Dirctry Results	(J) Link

Floor Height=0
Ceil. Height=2300

OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =

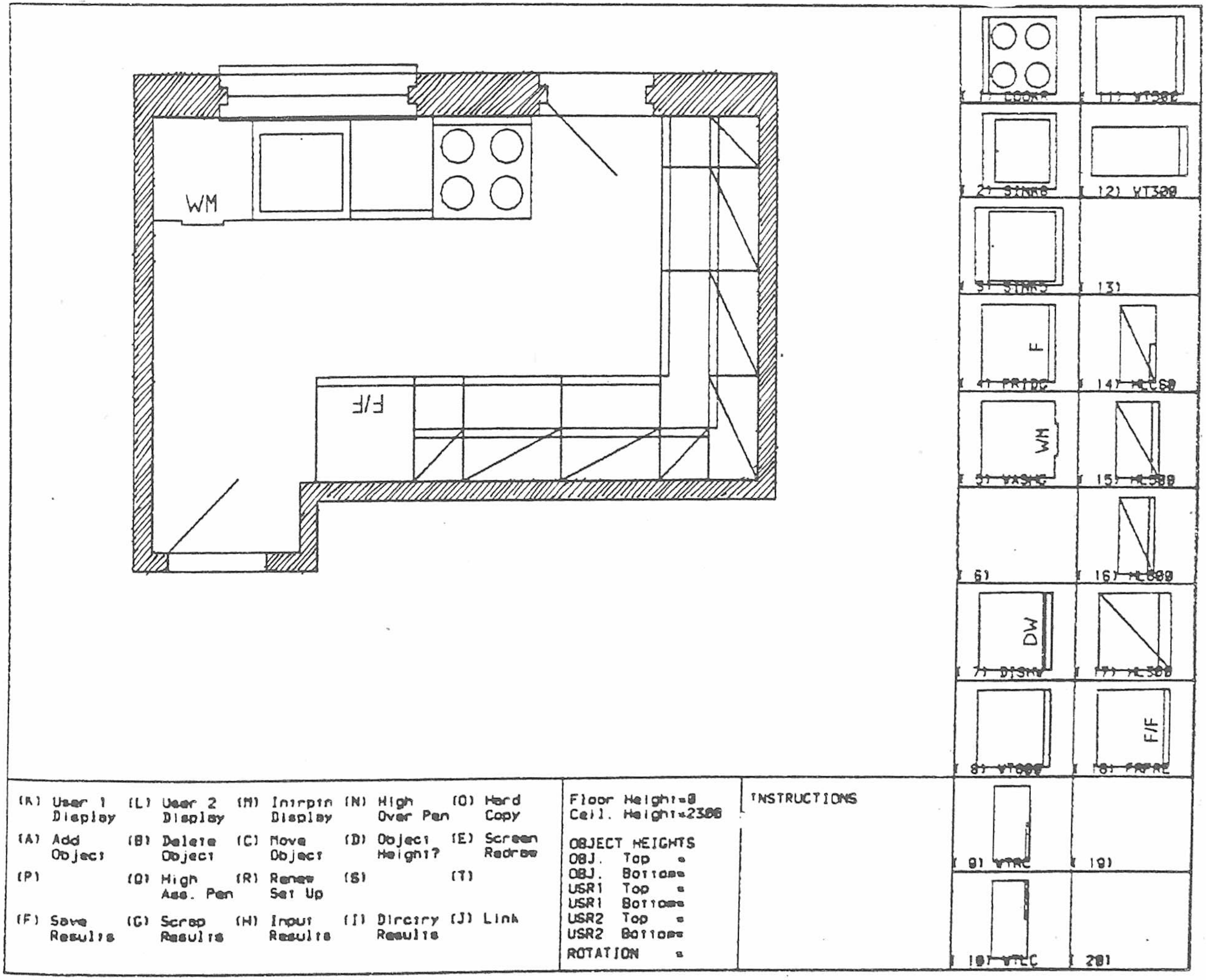
INSTRUCTIONS

R



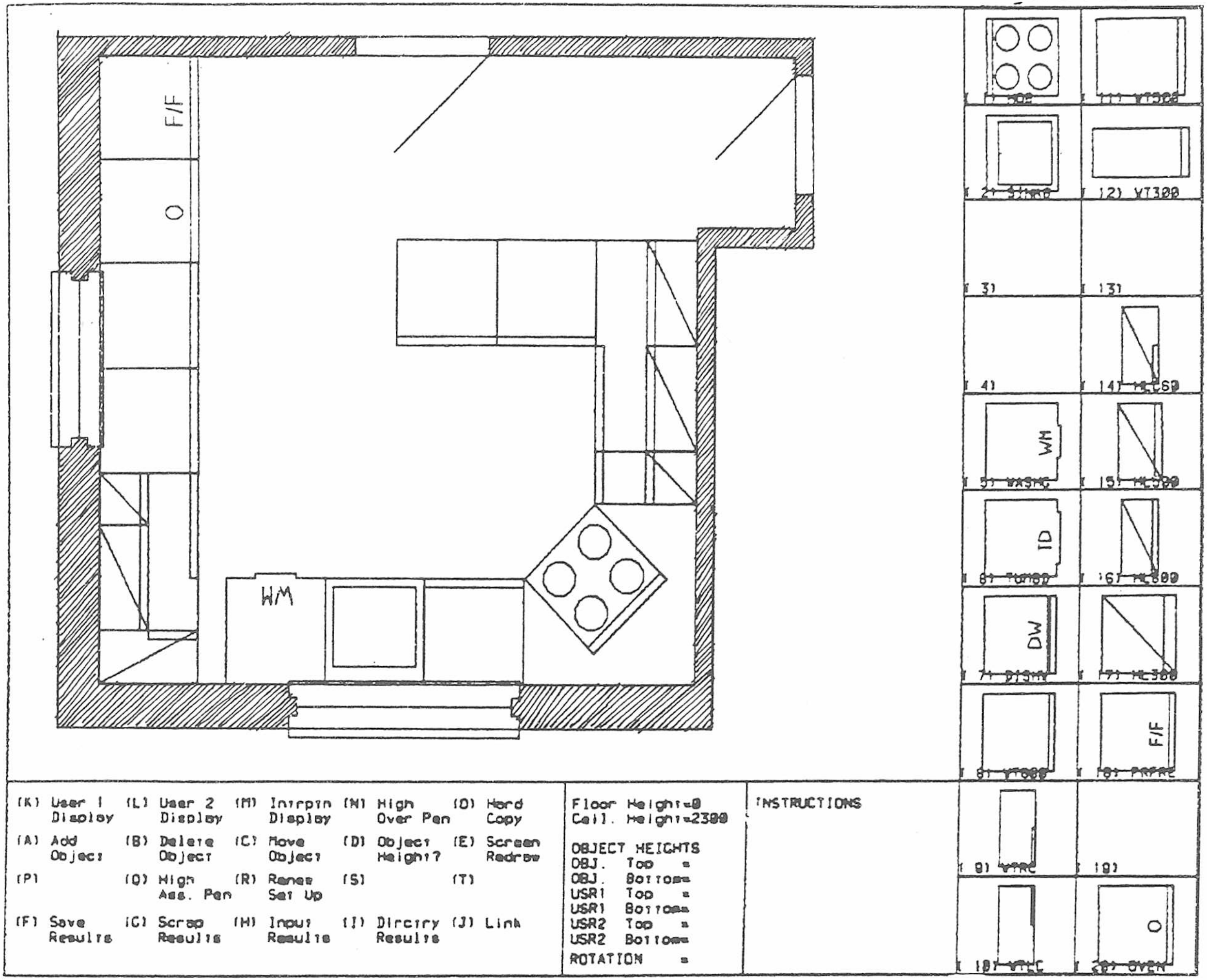
WM
F/F
O
1) HOB
11) WT300
2) SINKS
12) WT300
3)
13)
4)
14) HL
5) WASHG
WM
15) HL300
6) TUMBD
TD
16) HL300
7) DISHW
DW
17) HL300
8) WT600
18) FRFRE
F/F
9) WTRC
19)
10) WTLC
20) OVEN
O
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

T


WM
F/F
1) COOKR
11) VT300
2) SINKB
12) VT300
3) SINKS
13)
4) FRIDG
F
14) HL600
5) WASHG
WM
15) HL300
6)
16) HL600
7) DISHW
DW
17) HL300
8) VT600
18) FRFRE
F/F
9) VTRC
19)
10) VTLC
20)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

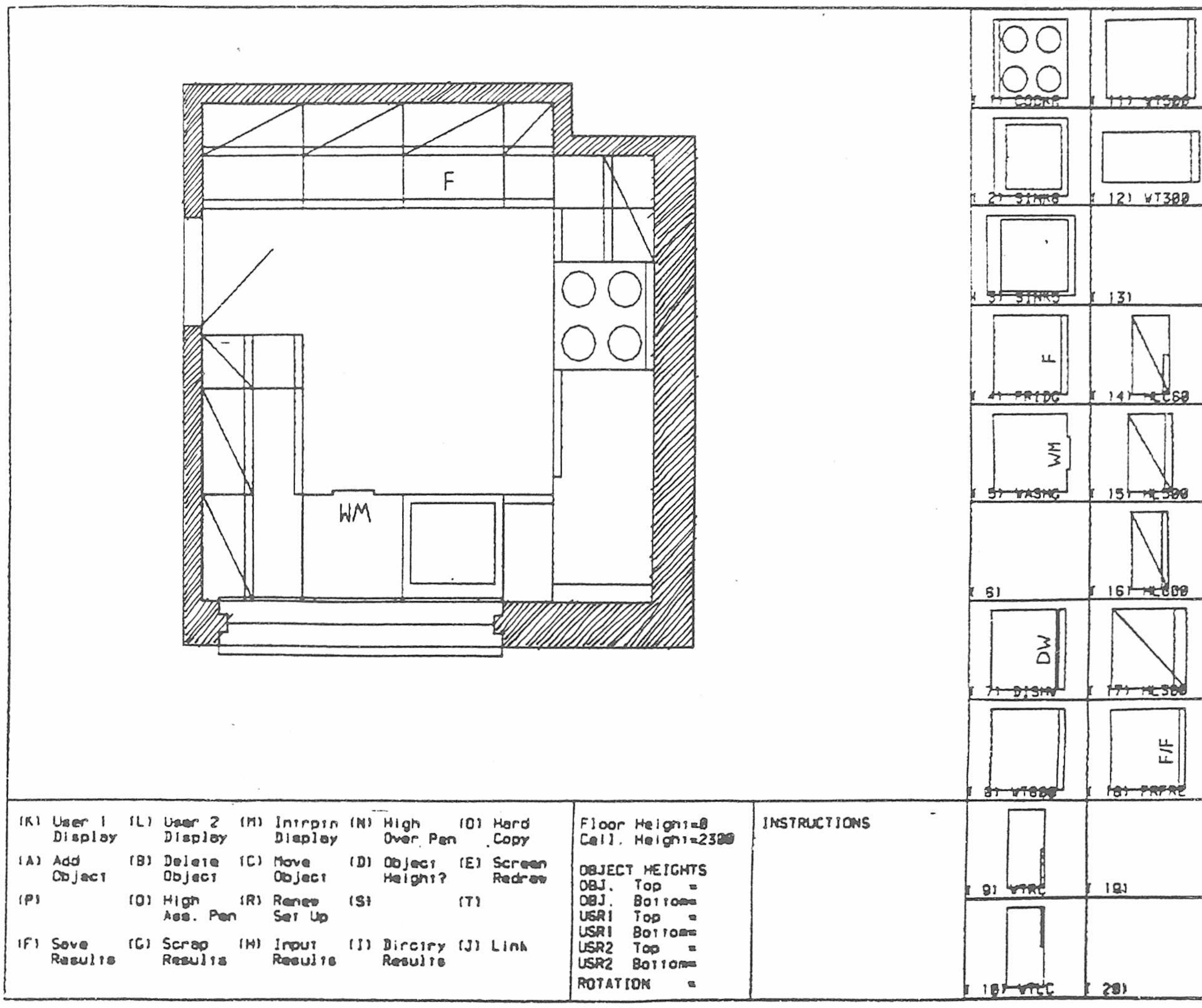
S

A.G.16



F/F
O
WM
DW
ID
WM
F/F
O
(2) YT300
(3)
(13)
(4)
(K) User 1 Display
(L) User 2 Display
(M) Intrpin Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P)
(Q) High Ass. Pen
(R) Renes Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
(9)
(19)

U



In the space below, please describe as best you can, the strategy you adopted in deciding which layout made best use of available space ...

....

....

....

G.2.2 Results and evaluation

There follows tables of results and scatterplots pertaining to this experiment.

Numeric & Physical Results.	SP penalty	VU penalty	AP penalty	OP penalty	ER penalty	Area (m^2)	Major Dimension (M)	Minor Dimension (M)	M. run of appliances	M. run of worktop	M. run of high level units	M. run of Doors & Windows
Kitchen A	1.004	1.135	0.101	1.518	0.947	7.20	3.0	2.4	2.4	3.0	2.4	2.45
Kitchen B	1.016	0.858	0.000	2.527	1.100	7.20	3.1	2.4	2.4	3.4	3.3	3.45
Kitchen C	1.054	1.007	0.000	2.279	1.085	10.86	3.6	3.6	3.0	5.6	3.8	2.55
Kitchen D	1.001	1.240	0.023	1.900	1.041	11.55	3.5	3.3	2.4	4.2	2.4	3.00
Kitchen E	1.006	1.521	0.097	1.335	0.990	12.00	4.0	3.0	2.4	3.3	3.6	3.00
Kitchen F	1.004	1.342	0.006	1.464	0.954	13.86	4.2	3.3	2.4	4.7	1.8	3.60
Kitchen G	1.076	0.960	0.207	1.894	1.034	9.12	3.8	3.0	3.0	3.6	3.3	3.60
Kitchen H	1.000	0.365	0.085	2.049	1.000	9.00	3.0	3.0	3.0	4.2	5.4	3.20
Kitchen I	1.000	1.102	0.107	1.757	1.011	9.92	3.2	3.1	2.4	3.3	3.8	2.10
Kitchen J	1.049	1.132	0.051	1.730	0.991	10.52	3.6	3.0	2.4	3.6	2.7	3.70
Kitchen K	1.000	1.032	0.040	1.706	0.945	10.23	3.3	3.1	2.4	4.9	3.3	3.40
Kitchen L	1.019	1.120	0.471	2.843	1.363	12.80	4.4	3.0	3.0	6.3	4.3	2.70
Kitchen M	1.015	1.079	0.203	2.103	1.100	14.40	4.8	3.0	3.6	6.4	4.8	3.40
Kitchen N	1.038	0.984	0.192	3.013	1.307	10.81	4.7	2.3	3.6	4.4	2.5	2.10
Kitchen O	1.010	1.108	0.022	1.938	1.019	13.20	4.4	3.0	4.8	5.4	4.8	4.20
Kitchen P	1.008	0.993	0.101	1.813	0.979	9.06	3.3	2.8	3.0	5.1	2.1	2.90
Kitchen Q	1.002	0.972	0.098	2.015	1.022	9.86	3.4	1.9	3.0	4.2	4.0	2.30
Kitchen R	1.002	1.251	0.101	2.072	1.107	9.86	3.4	2.9	3.0	2.6	1.1	3.20
Kitchen S	1.033	1.213	0.551	1.836	1.158	13.56	3.6	3.6	3.0	4.3	2.7	4.10
Kitchen T	1.054	0.904	0.138	2.577	1.168	8.13	3.7	2.1	2.4	4.1	3.9	2.80
Kitchen U	1.008	0.849	0.000	3.290	1.287	7.38	2.8	2.7	2.4	4.9	4.2	3.40

Fig. G.1 Table of computer and numerical evaluations of 21 kitchen designs

2nd Year student ranking of areas	student 1	student 2	student 3	student 4	student 5	student 6	student 7	student 8	student 9		measured area
Kitchen A	1	3	3	2	2	3	6	2	1		1=
Kitchen B	2	6=	1	1	1	4	1	1	2		1=
Kitchen C	13	13	11	18	14	11	11	14	12		14
Kitchen D	16	7	12	14	16	16	16	16	16		15
Kitchen E	15	15	15	15	15	20	15	15	17		16
Kitchen F	19	19	17	17	21	21	18	20	18		20
Kitchen G	6	9	9=	11	7	8	9	10	10		7
Kitchen H	5	2	5	8	5	9	8	5	6		5
Kitchen I	10	6=	6	9=	11	10	12	8	11		10
Kitchen J	14	11	13	13	13	14	13	13	13		12
Kitchen K	11	8	10	9=	9	12	10	11	9		11
Kitchen L	17	20	18	16	18	13	19	17	14		17
Kitchen M	20=	21	20	21	19	19	20	21	19		21
Kitchen N	12	14	14	12	10	15	14	12	15		13
Kitchen O	18	17	16	20	17	18	17	18	20		18
Kitchen P	7	12	4	6	6	5	7	6	5		6
Kitchen Q	8	5	7	5	12	6	5	7	7		8=
Kitchen R	9	10	9=	7	8	7	4	9	8		8=
Kitchen S	20=	18	19	19	20	17	21	19	21		19
Kitchen T	3	4	8	4	4	2	3	4	4		4
Kitchen U	4	1	2	3	3	1	2	3	3		3

Fig. G.2 Table of student results (area rankings)

2nd Year student ranking of spaces.	student 1	student 2	student 3	student 4	student 5	student 6	student 7	student 8	student 9		computer ER.
Kitchen A	16	15=	15	14	12	7	19	9	6		2
Kitchen B	4	12	6	7	2	1	2	8	18		14=
Kitchen C	13	11	12	17	14	14	3	17	16		13
Kitchen D	10	20	5	6	19	13	17	19	2		12
Kitchen E	21	21	21	4	20	19	20=	18	7		5
Kitchen F	20	9	3	3	21	17	20=	21	1		3
Kitchen G	9	10	13	9	4	10	7	15	17		11
Kitchen H	1	3	16	16	10	8	8	4	4		7
Kitchen I	11	3=	10	19	17	6	13	13	11		8
Kitchen J	12	18	4	12	6	12	11	2	20		6
Kitchen K	5	15=	9	20	9	9	12	3	13		1
Kitchen L	17	19	1	1	18	20	14	6	10		21
Kitchen M	3	17	8	5	13	16	15	1	5		14=
Kitchen N	7	7	18	18	8	5	5	16	9		20
Kitchen O	15	6	20	8	11	11	18	7	15		9
Kitchen P	14	2	19	10	3	18	4	12	19		4
Kitchen Q	6	14	14	13	5	2	9	5	3		10
Kitchen R	2	3=	2	2	16	15	10	14	8		16
Kitchen S	19	13	7	21	15	21	16	20	12		17
Kitchen T	18	1	17	11	7	4	6	10	21		18
Kitchen U	8	5	11	15	1	3	1	11	14		19

Fig. G.3 Table of student results (efficiency rankings)

Fig. G.4 Table of staff results (efficiency rankings)

Staff ranking of Spaces.	Staff 1	Staff 2
Kitchen A	6	12=
Kitchen B	13	8=
Kitchen C	5	18=
Kitchen D	17	2=
Kitchen E	21	20=
Kitchen F	15	20=
Kitchen G	7	14=
Kitchen H	18	1
Kitchen I	19	11
Kitchen J	20	14=
Kitchen K	1	4=
Kitchen L	14	6
Kitchen M	2	4=
Kitchen N	3	8=
Kitchen O	10	8=
Kitchen P	4	12=
Kitchen Q	11	4=
Kitchen R	8	2=
Kitchen S	16	7=
Kitchen T	12	18=
Kitchen U	9	4=

Fig. G.5 Table of computer and numerical evaluations (rankings)

Ranking of above.	least SP	least VC	least AP	least OP	least ER.	least area	least major dimension (m)	least minor dimension (m)	least m. run of appliances	least m. run of worktops	least m. run of H/L units	least m. run of doors & windows
Kitchen A	7=	15	12=	3	2	1=	2=	4=	1=	2	4	4
Kitchen B	14	2	1=	17	14=	1=	4	4=	1=	5	8=	16
Kitchen C	19=	9	1=	16	13	14	11=	20=	11=	19	12=	5
Kitchen D	4	18	6	10	12	15	10	18=	1=	9=	16=	9=
Kitchen E	9	21	10	1	5	16	16	10=	1=	3=	11	9=.
Kitchen F	7=	20	4	2	3	20	17	18=	1=	14	2	17=
Kitchen G	21	5	19	9	11	7	15	10=	11=	6=	8=	17=
Kitchen H	1=	3	9	13	7	5	2=	10=	11=	9=	21	11=
Kitchen I	1=	16	15	6	8	10	5	16=	1=	3=	12=	1=
Kitchen J	18	14	8	5	6	12	11=	9	1=	6=	6=	19
Kitchen K	1=	16	7	4	1	11	6=	16=	1=	15=	8=	13=
Kitchen L	15	13	20	19	21	17	18=	10=	11=	20	18	6
Kitchen M	13	11	18	15	14=	21	21	10=	19=	21	19=	13=
Kitchen N	17	7	17	20	20	13	20	3	19=	13	5	1=
Kitchen O	12	12	5	11	9	18	18	10=	21	18	19=	21
Kitchen P	10=	8	12=	7	4	6	6=	7	11=	17	3	8
Kitchen Q	5=	6	11	12	10	8=	8=	1	11=	9=	15	3
Kitchen R	5=	19	12=	14	16	8=	8=	3	11=	1	1	11=
Kitchen S	16	17	21	8	17	19	11=	20=	11=	12	6=	20
Kitchen T	19=	4	16	18	18	4	14	2	1=	8	14	7
Kitchen U	10=	1	1=	21	19	3	1	6	1=	15=	16=	13=

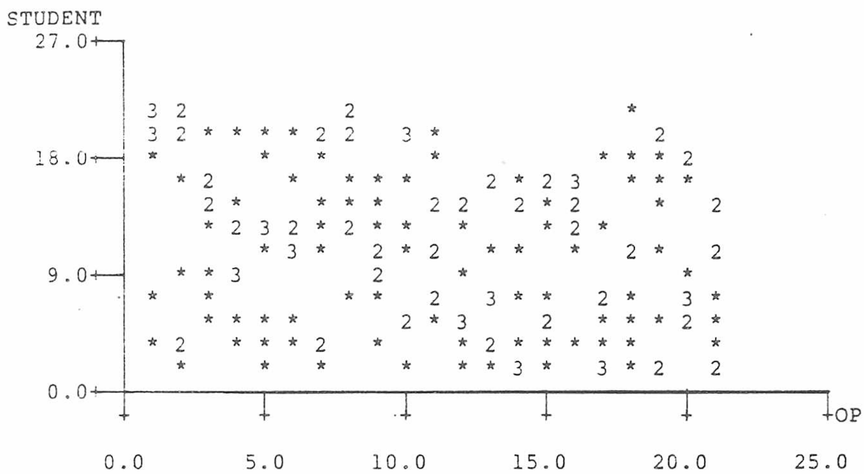


Fig. G.6 Scatterplot of student results against ranked overlap penalty values

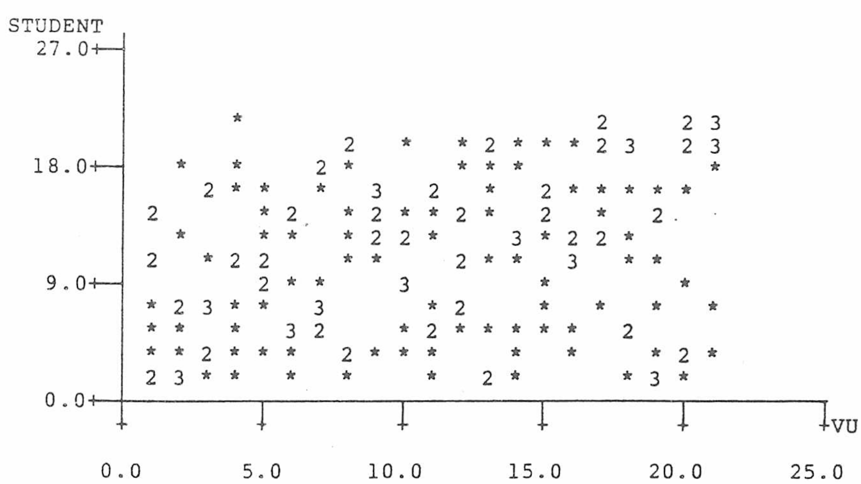


Fig. G.7 Scatterplot of student results against ranked volume utilisation values

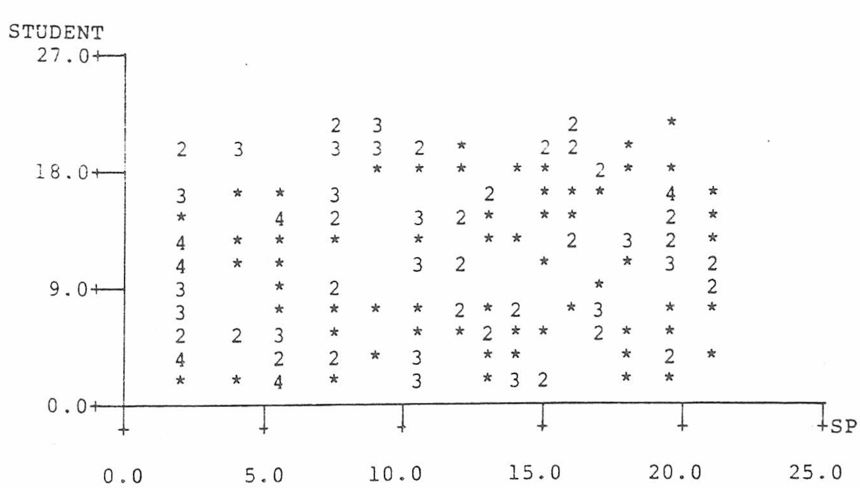


Fig. G. 8 Scatterplot of student results against ranked space penalty values

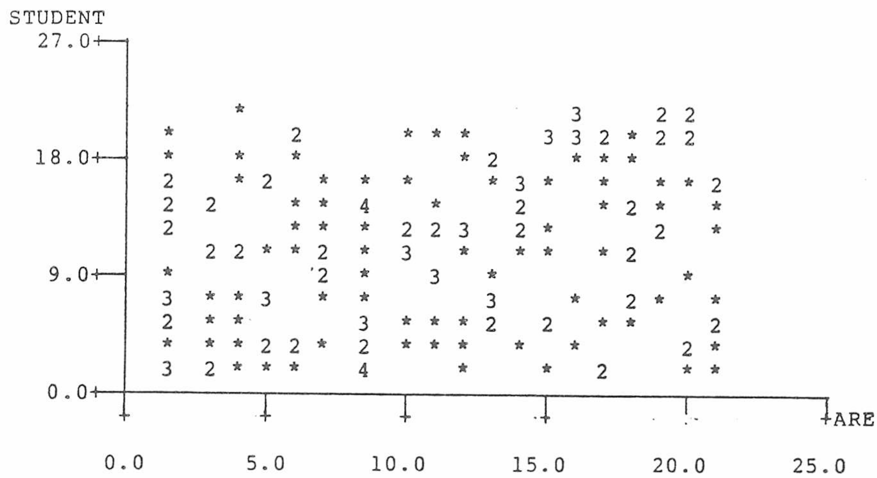


Fig. G.9 Scatterplot of student results against ranked floor area values

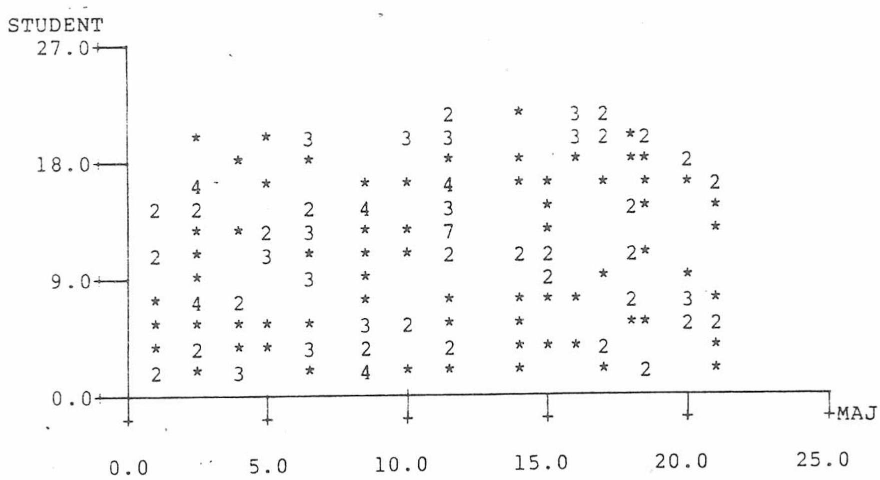


Fig. G.10 Scatterplot of student results against ranked major dimension of room value

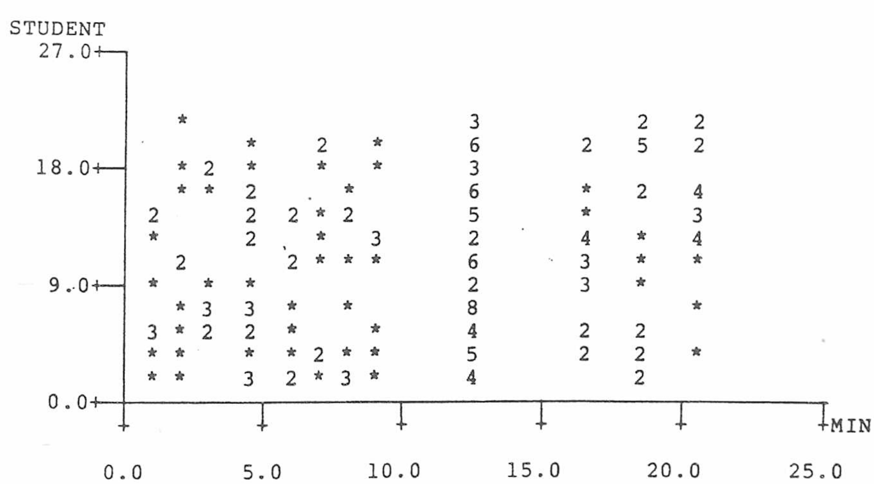


Fig. G.11 Scatterplot of student results against ranked minor dimension of room value

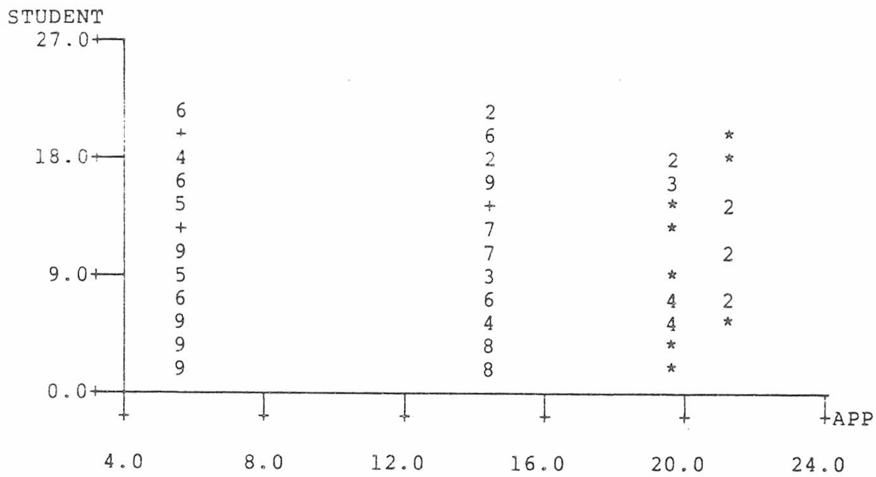


Fig. G.12 Scatterplot of student results against ranked m. run of appliances value

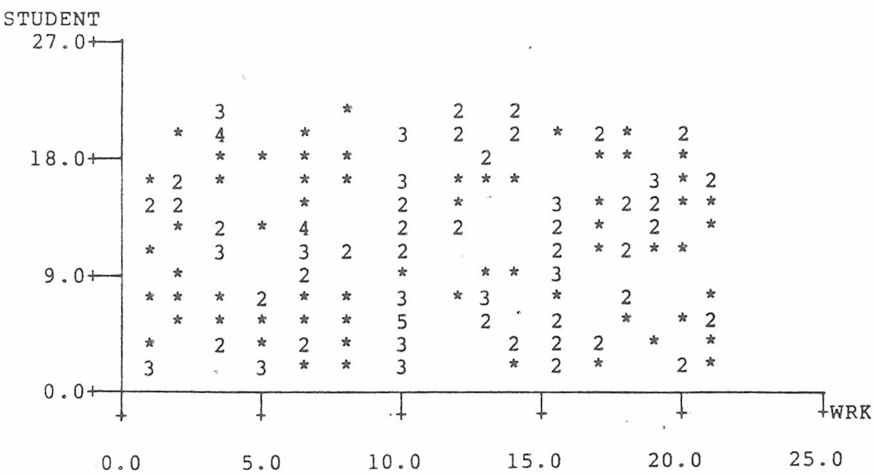


Fig. G.13 Scatterplot of student results against ranked m.run of worktop value

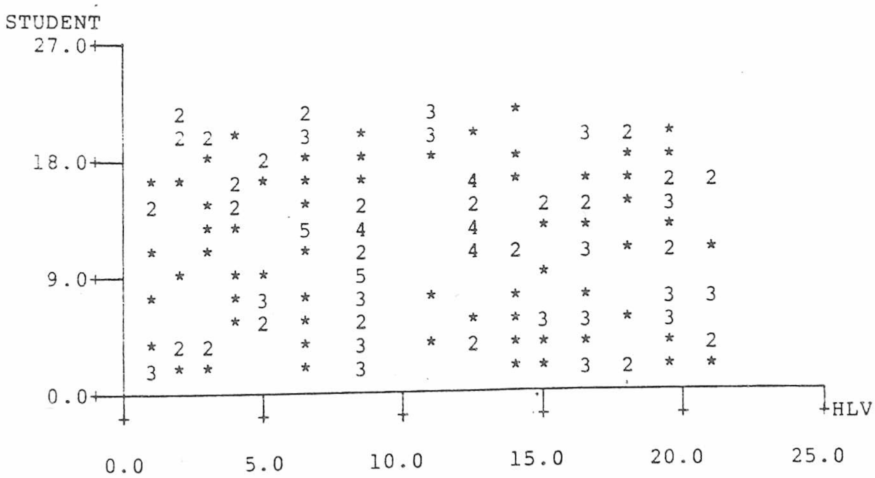


Fig. G.14 Scatterplot of student results against ranked m.run of high level units value

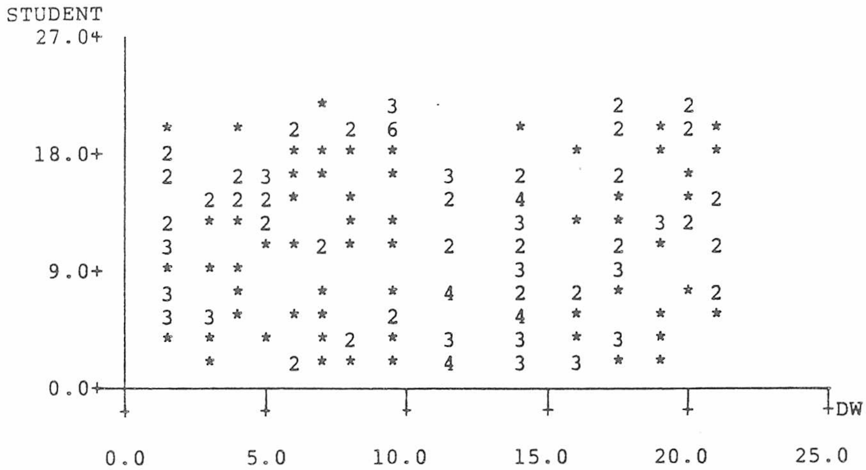


Fig. G. 15 Scatterplot of student results against ranked m. run of doors and windows value

G.3.0 EXPERIMENT 2

This section contains addenda to the second experiment.

G.3.1 Questionnaire design

Below is a sample of the questionnaire used in this experiment.

KITCHEN LAYOUT SURVEY/QUESTIONNAIRE

Your assistance is requested with this questionaire/survey. There is a great deal less work involved than the number of pages imply. If you are a student rest assured that the questionaire forms no part of your coursework and will not be 'assessed' by any member of staff. In any case, all completed questionaires will be treated in the strictest confidence.

If you've got down to here you must have volunteered - so thanks for the help.

WHAT YOU SHOULD DO.

A selection of ten pairs of kitchen layouts are presented to you in the following pages.

Please mark with a large tick, which kitchen of the pair presented on each page, makes BETTER use of the available space. In each case the room shape is identical and the number and type of furniture units is the same.

Also, please write your name on the front cover of the questionaire before returning it. Thank you.

FOR YOUR INFORMATION ONLY.

1. Each room layout is to the same scale (approx. 1:45; the odd scale is due to the vagaries of reproduction).

2. All windows are set with a sill height of 1050mm and a head height of 2100mm. Doors are full length and have a head height of 2100mm.

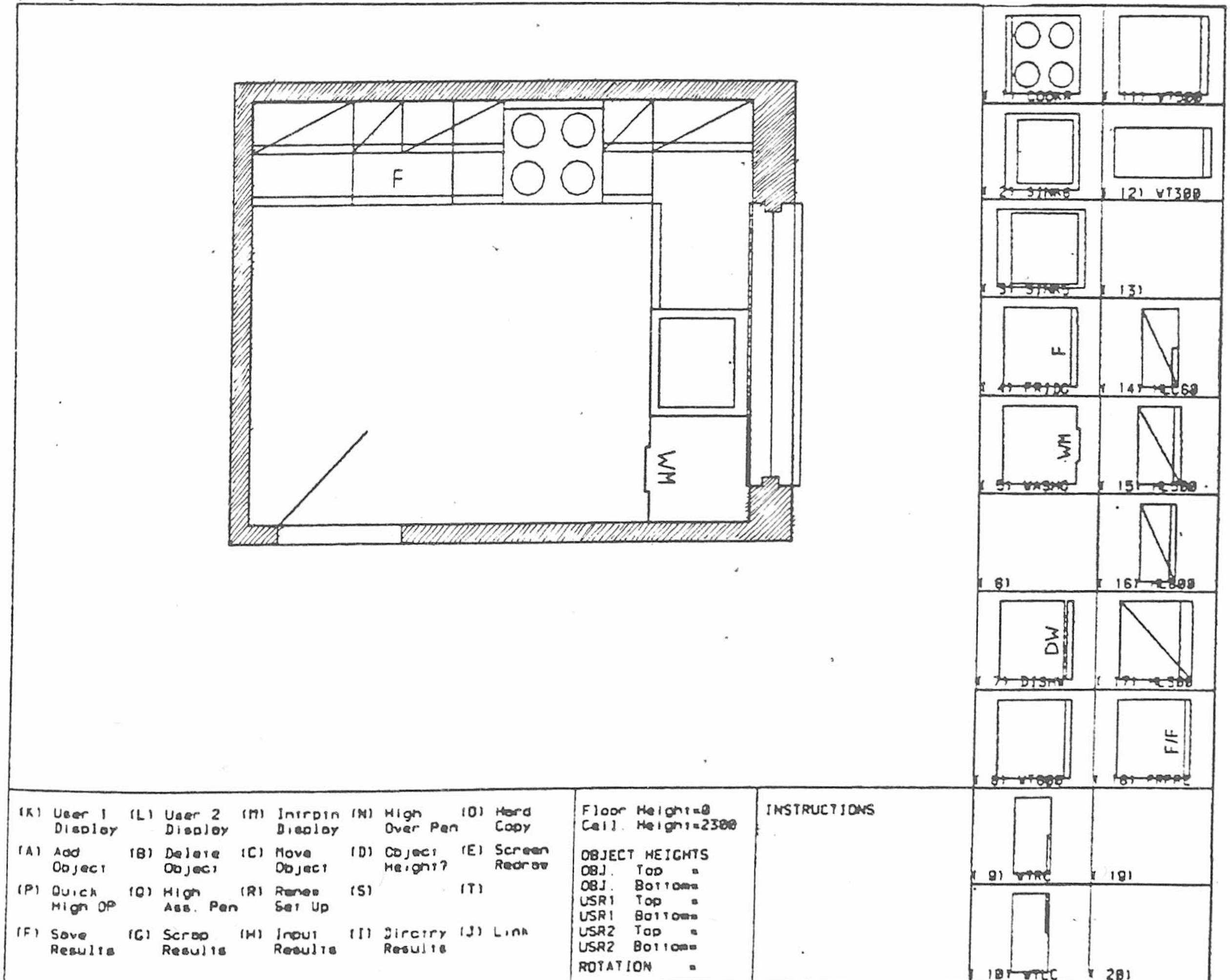
3. Most kitchen appliances (or their housings) are 600x600x900mm.

4. Exceptions are fridge/freezers (F/F), oven housings (O) and tall cupboards (600x600 and marked with a diagonal line) which are 2000mm high.

5. Most work top covered base units are 600x600, though some rooms also have 600x500 and 600x300 units. These are easily recognised on plan. All have a height of 900mm.

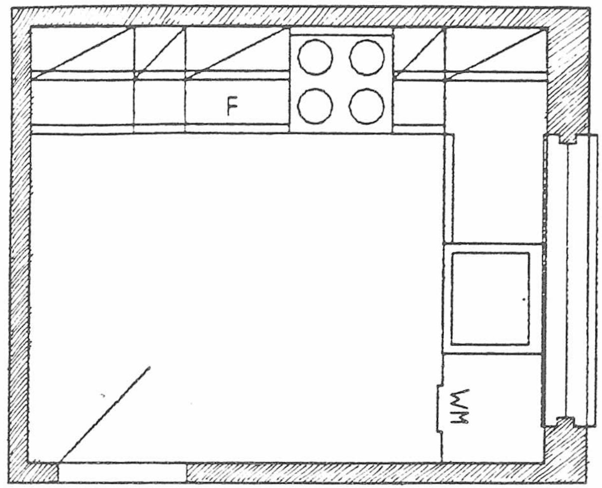
6. High level units (marked by a diagonal line running through them) are generally 300x600, though again, 300x500 and 300x300 units have been used in some rooms. All high level units are 650mm high and are set at 1350mm from ground level.

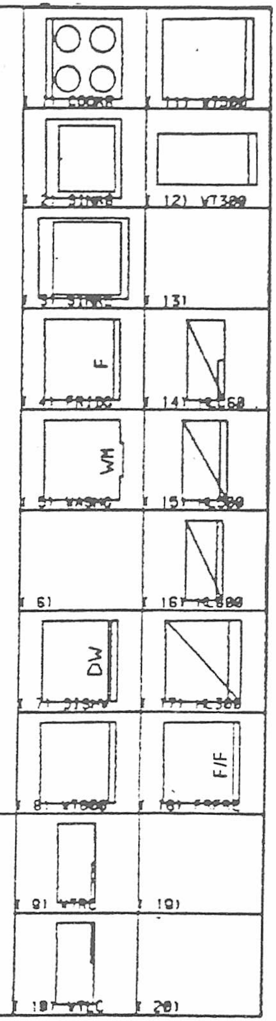
1b



F
WM
DW
F/F
COOKR
(1) VT300
(2) SINKB
(2) VT300
(3) SINKS
(3)
(4) FRIDG
(14)
(5) WASHG
(15)
(6)
(16)
(7) DISHW
(17)
(8)
(9) VTRC
(19)
(10) VTCC
(20)
(K) User 1 Display
(L) User 2 Display
(M) Intrpin Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renee Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

1a





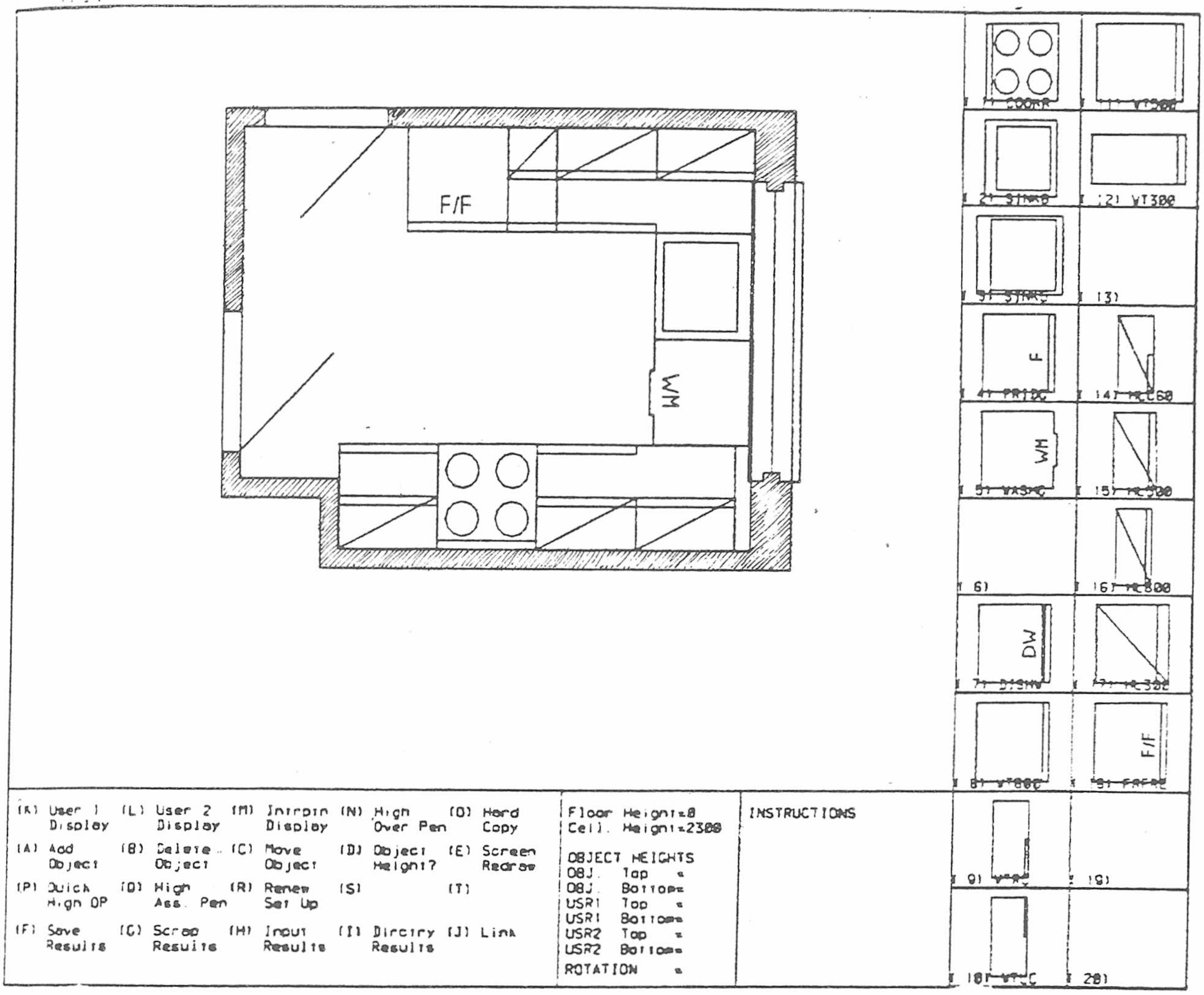
(K) User 1 Display	(L) User 2 Display	(M) Intrpin Display	(N) High Over Pen	(O) Hard Copy
(A) Add Object	(B) Delete Object	(C) Move Object	(D) Object Height?	(E) Screen Redraw
(P)	(Q) High Ass. Pen	(R) Renew Set Up	(S)	(T)
(F) Save Results	(G) Scrap Results	(H) Input Results	(I) Dirctry Results	(J) Link

Floor Height=0
Ceil. Height=2300

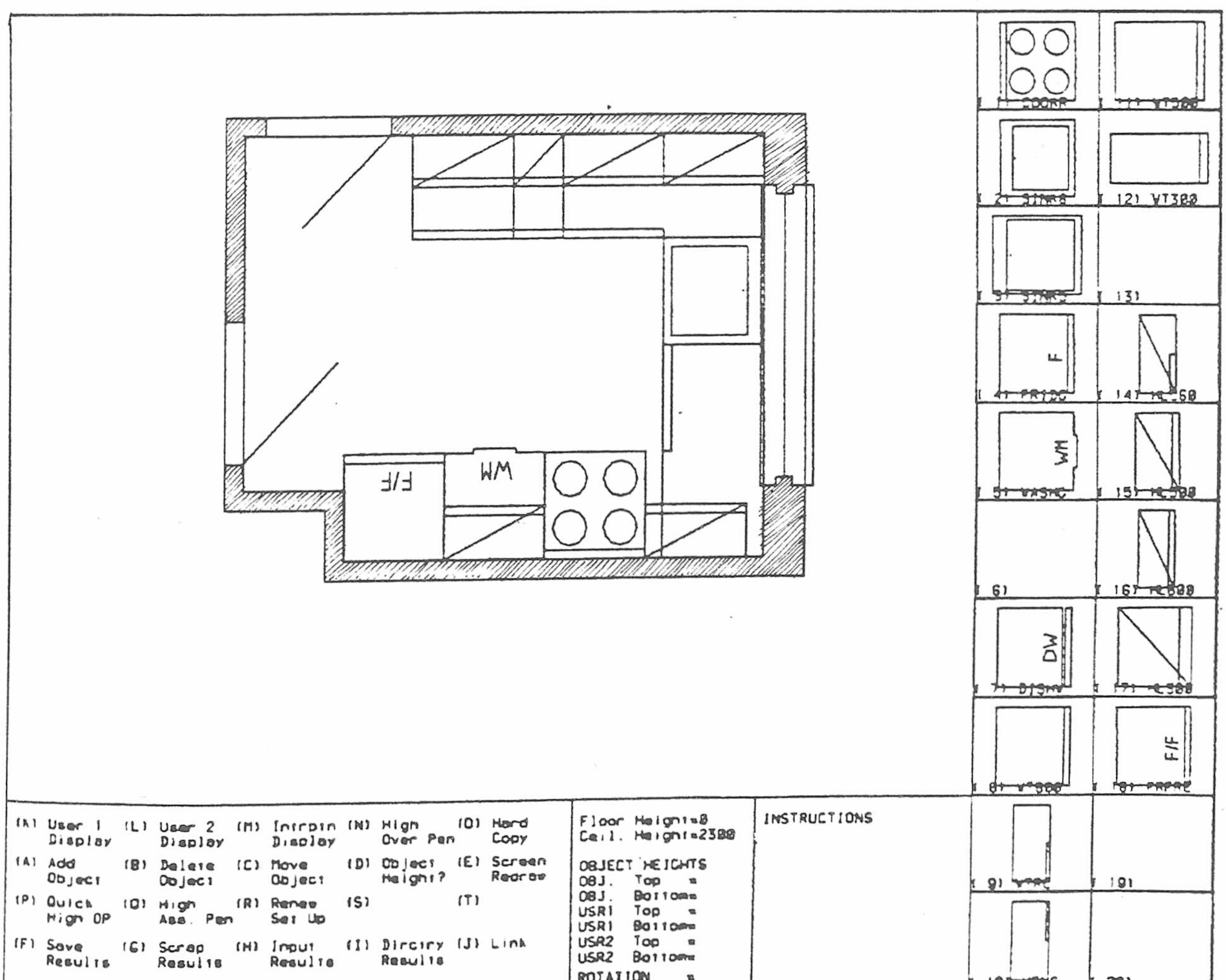
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =

INSTRUCTIONS

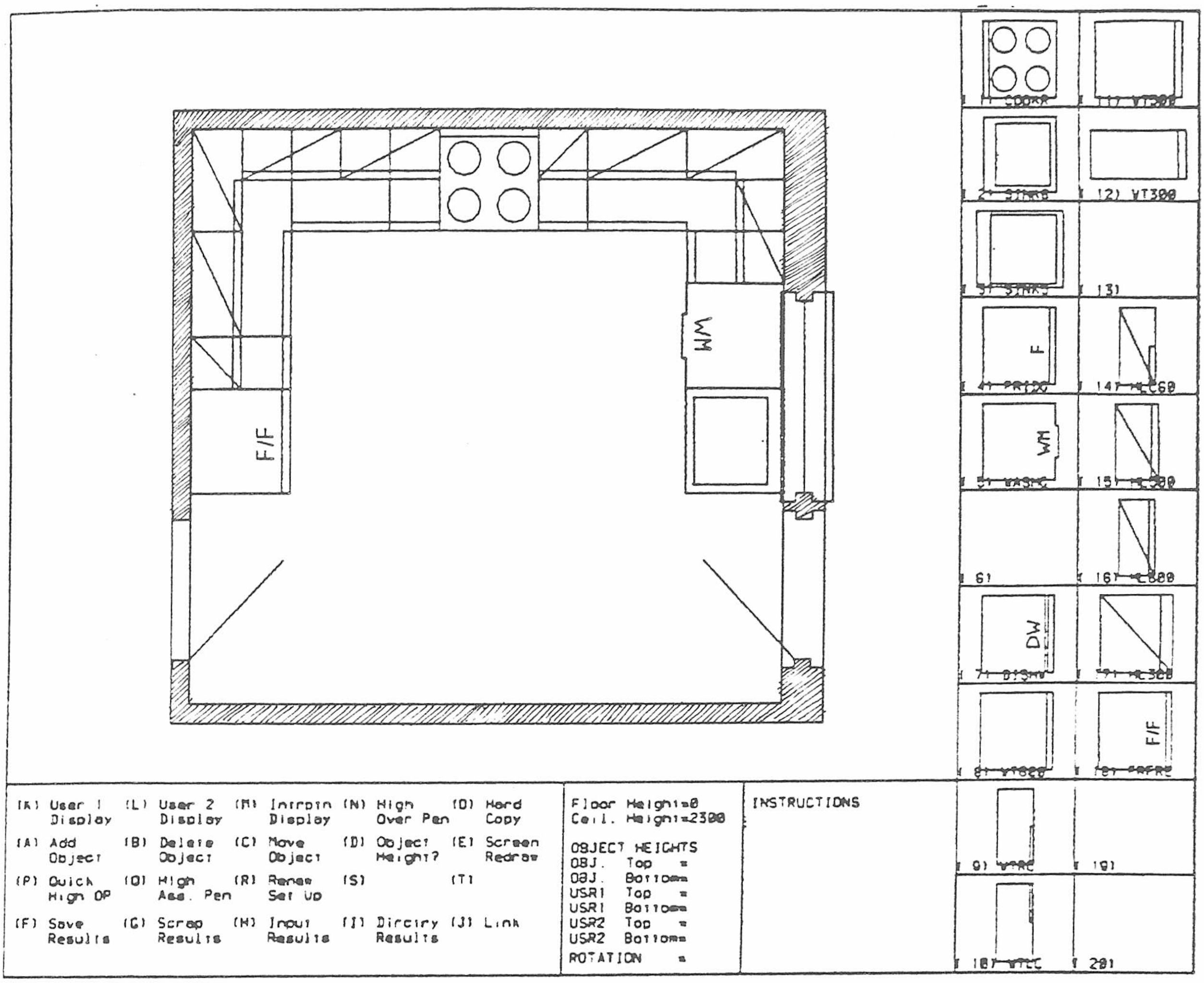
2a


F/F
WM
COOKR
VT300
SINKB
VT300
SINKS
FRIDG
F
WM
WASHG
DW
DISHW
F/F
FRFRZ
(K) User 1 Display
(L) User 2 Display
(M) Intrpln Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

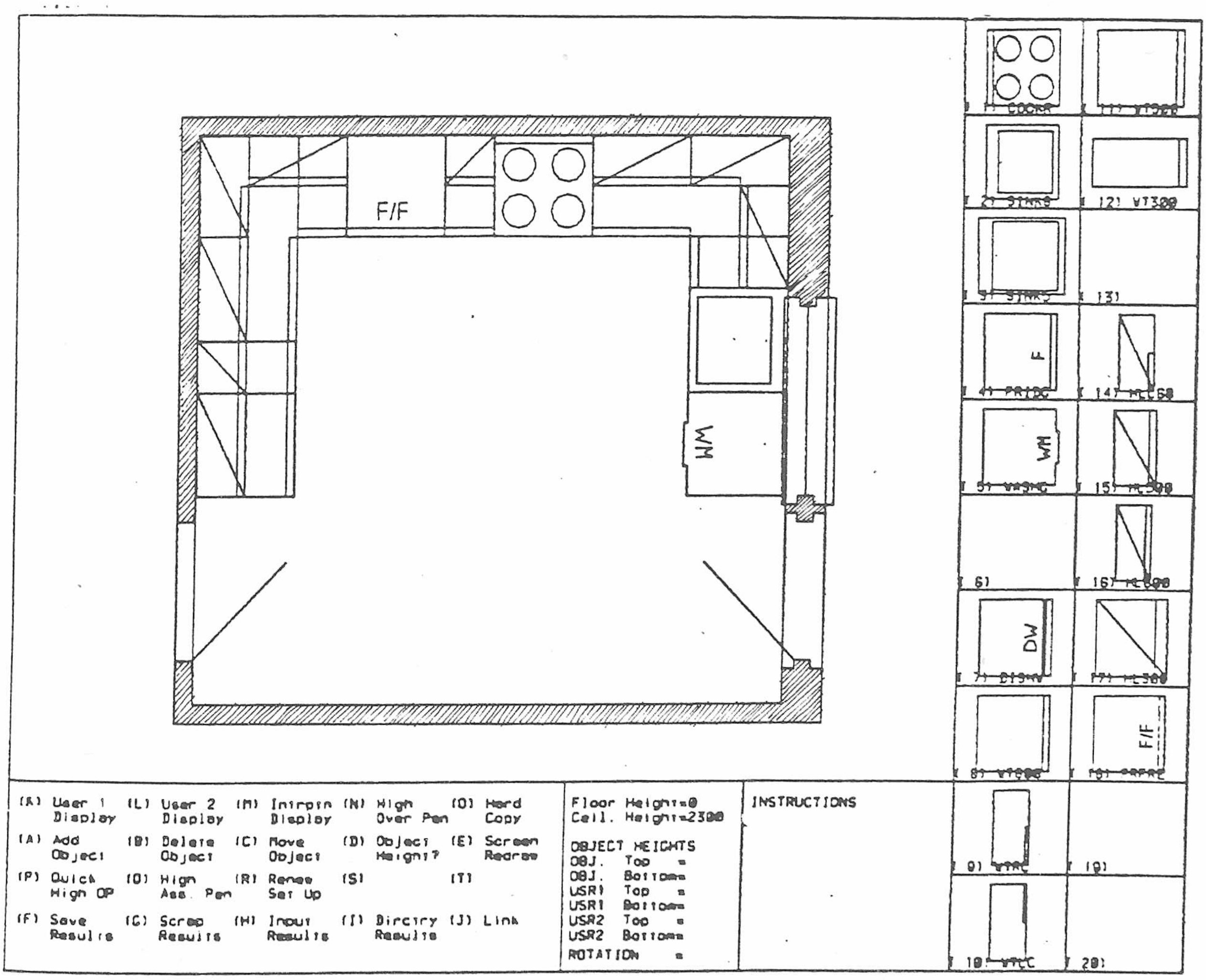
2b


F/F
WM
COOKR
VT300
SINKB
VT300
SINKS
FRIDG
F
WM
WASHG
DW
DISHW
F/F
FRFRZ
(K) User 1 Display
(L) User 2 Display
(M) Intrpln Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

3a

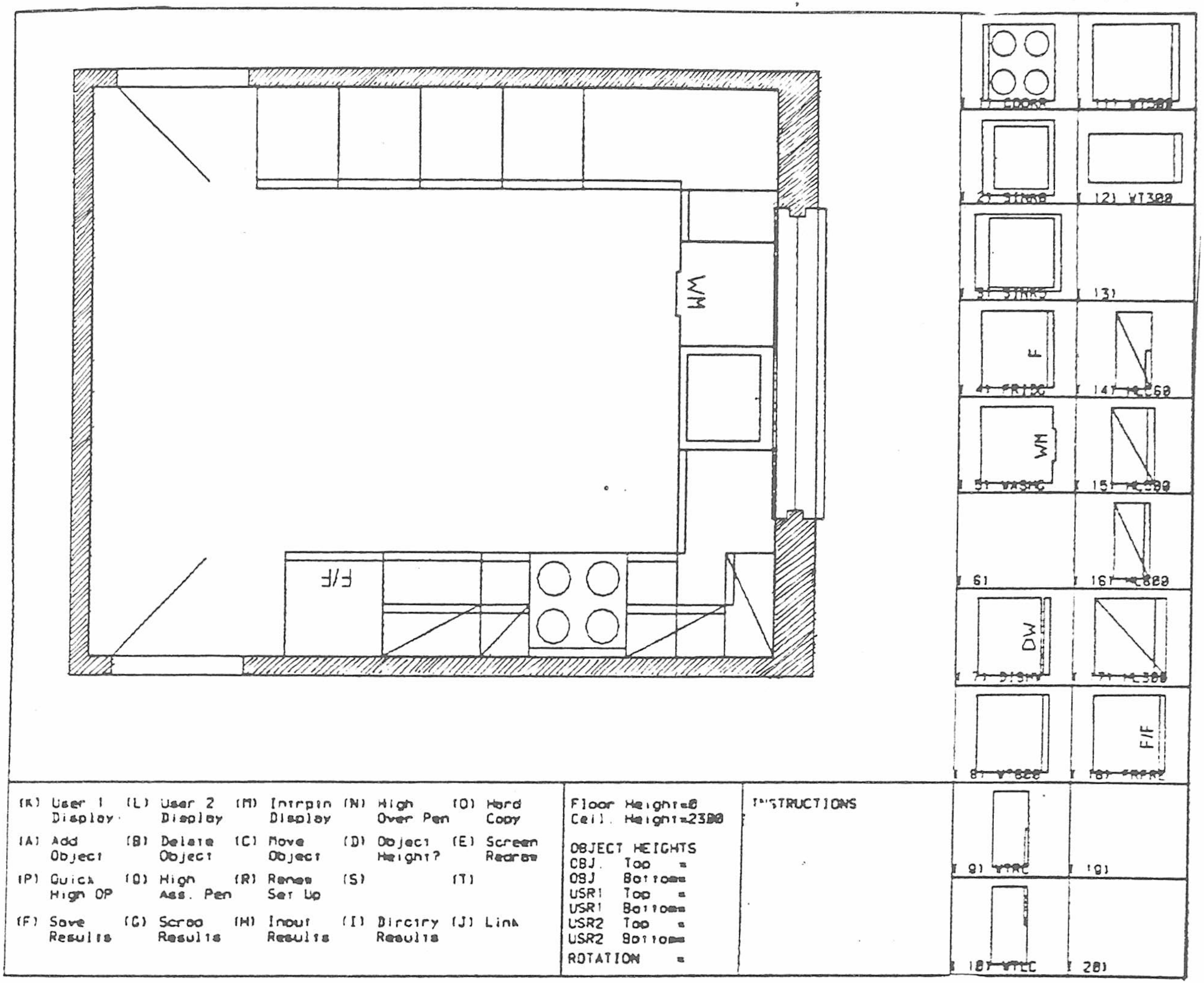

F/F
WM
F
DW
(K) User 1 Display
(L) User 2 Display
(M) Intrpin Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
12) VT300
13)
6)
19)
20)

3b


F/F
WM
F
DW
(K) User 1 Display
(L) User 2 Display
(M) Intrpin Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
12) VT300
13)
6)
19)
20)

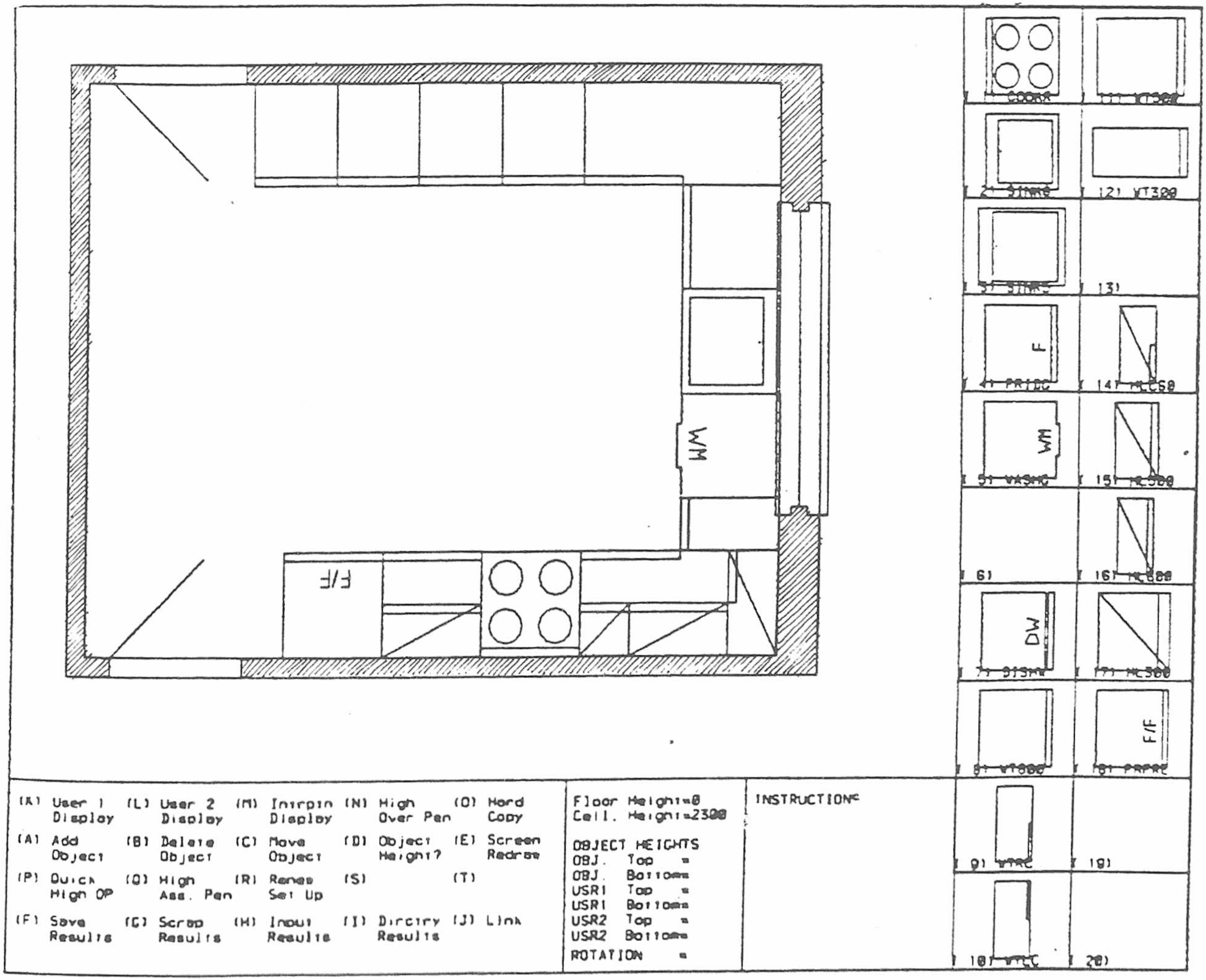
A.G.30

4a

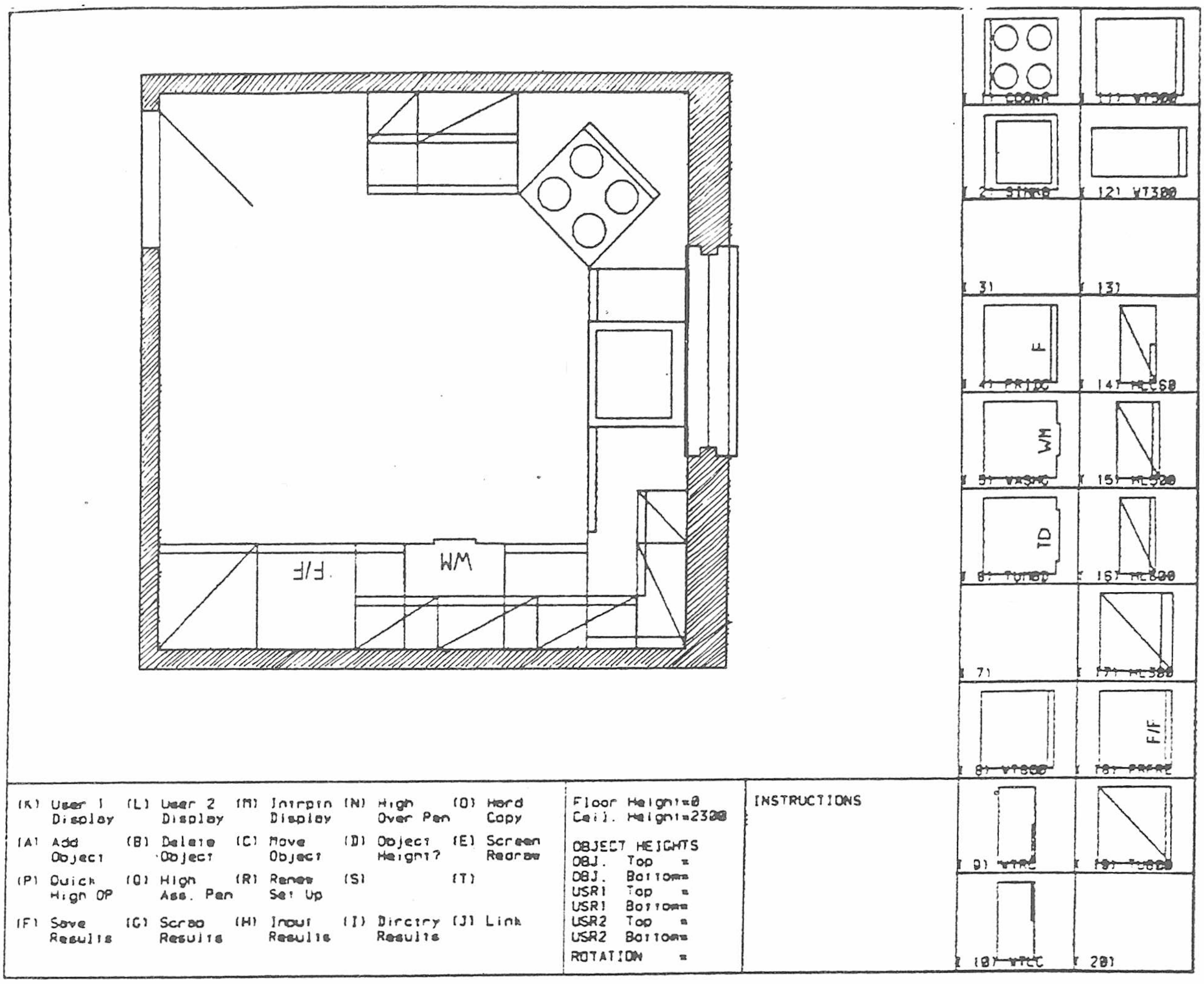


WM
F/F
DW
F
(12) WT300
(13)
(6)
(9)
(19)
(20)
(K) User 1 Display
(L) User 2 Display
(M) Intrpin Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renes Set Up
(S)
(T)
(F) Save Results
(G) Scrod Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

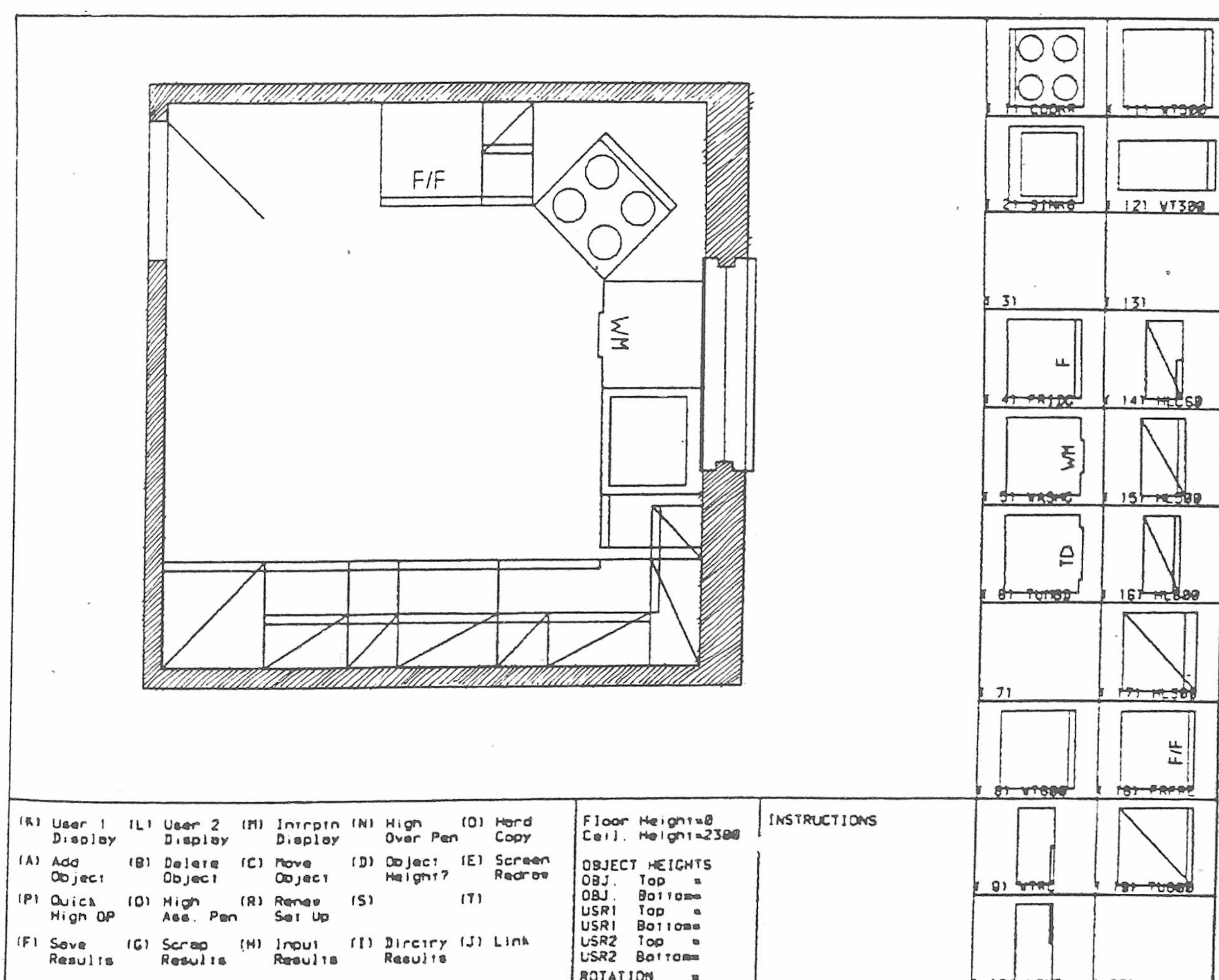
4b


F/F
WM
F
DW
12) WT300
13)
6)
19)
20)
(K) User 1 Display
(L) User 2 Display
(M) Intrpin Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

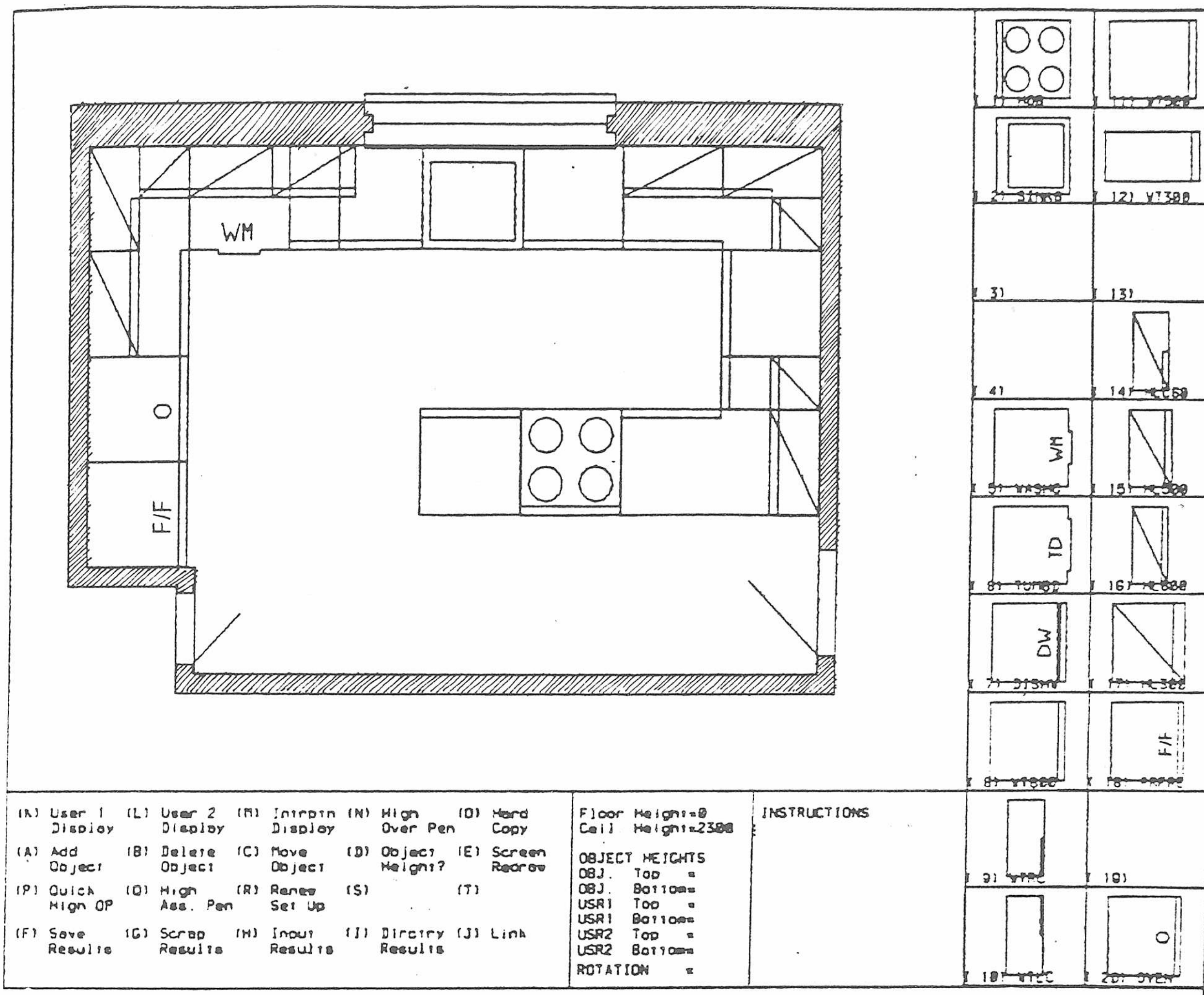
5a


F/F
WM
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
1) COOKR
11) VT300
2) SINK
12) VT300
3)
13)
4) FRIDG
14) HLC50
5) WASHG
15) HL500
6) TUMBD
16) HL600
7)
17) HL300
8) VT600
18) FRFRZ
9) VTRC
19) TUB60
10) VTCC
20)
F
WM
TD
F/F

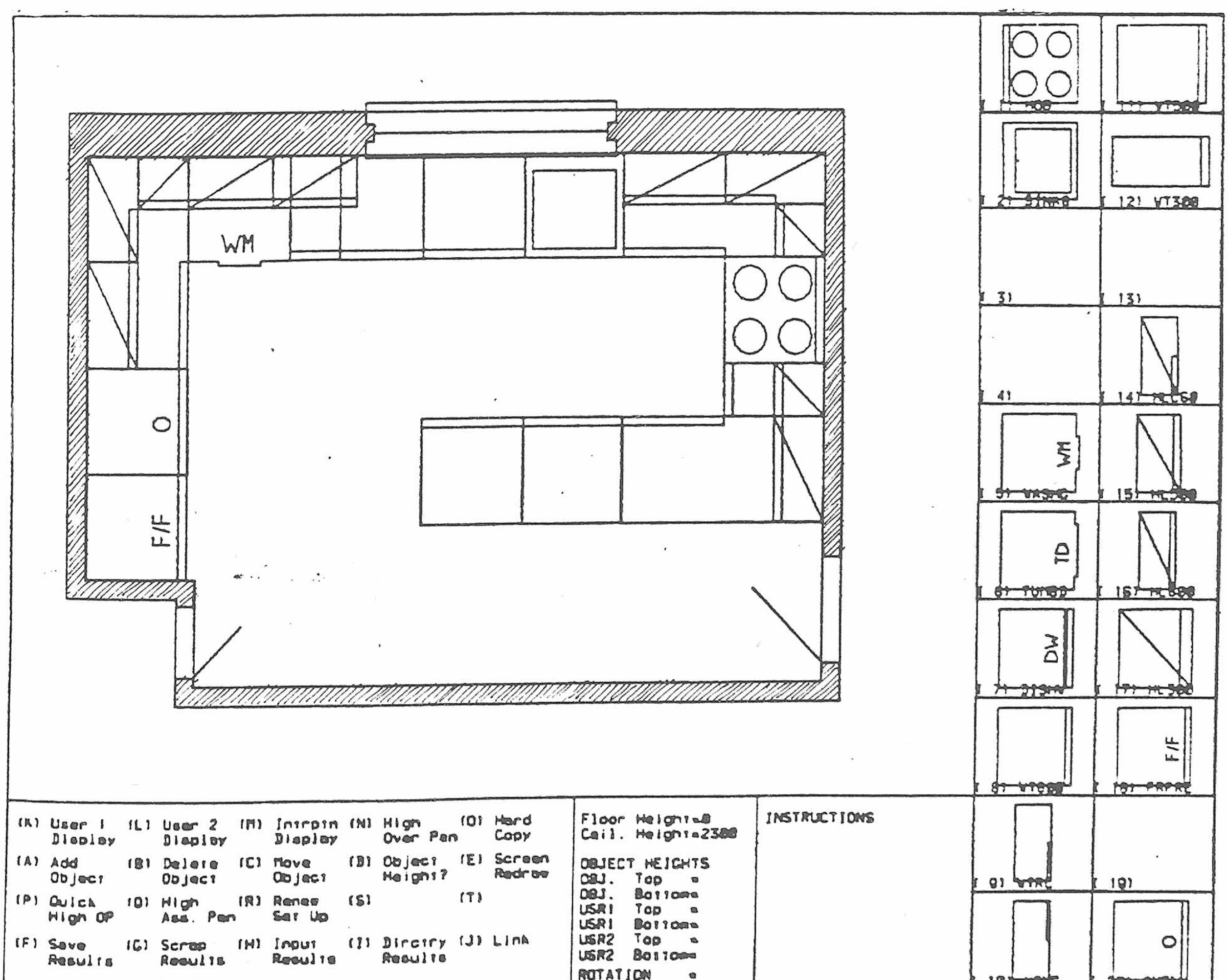
5b


F/F
WM
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
1) COOKR
11) VT300
2) SINK
12) VT300
3)
13)
4) FRIDG
14) HLC50
5) WASHG
15) HL500
6) TUMBD
16) HL600
7)
17) HL300
8) VT600
18) FRFRZ
9) VTRC
19) TUB60
10) VTCC
20)
F
WM
TD
F/F

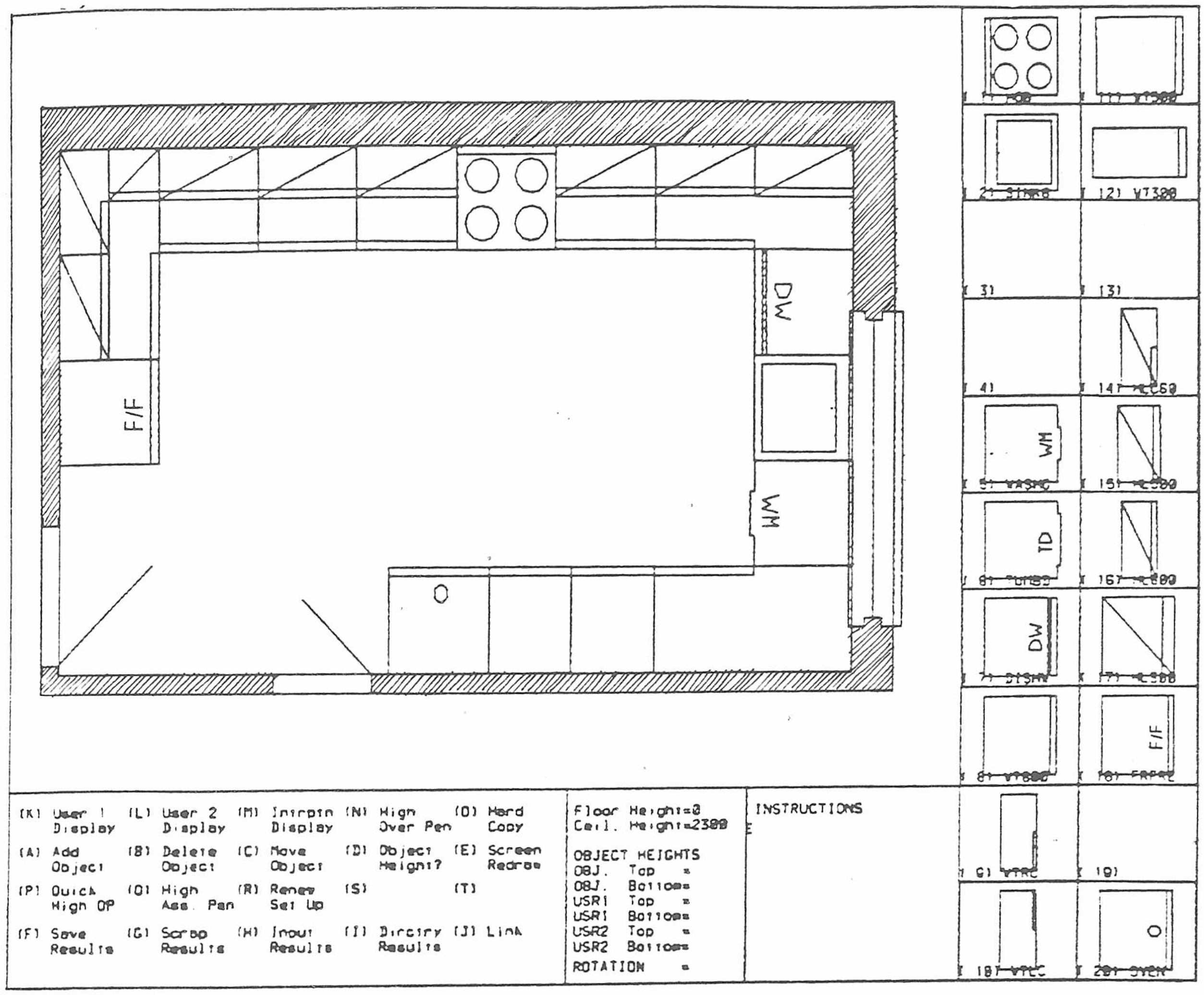
6a


WM
O
F/F
1) HOB
11) VT300
2) SINK
12) VT300
3)
13)
4)
14)
5)
15)
8)
16)
7)
17)
8)
18)
9)
19)
10)
20) OVEN
TD
DW
(K) User 1 Display
(L) User 2 Display
(M) Intrpln Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

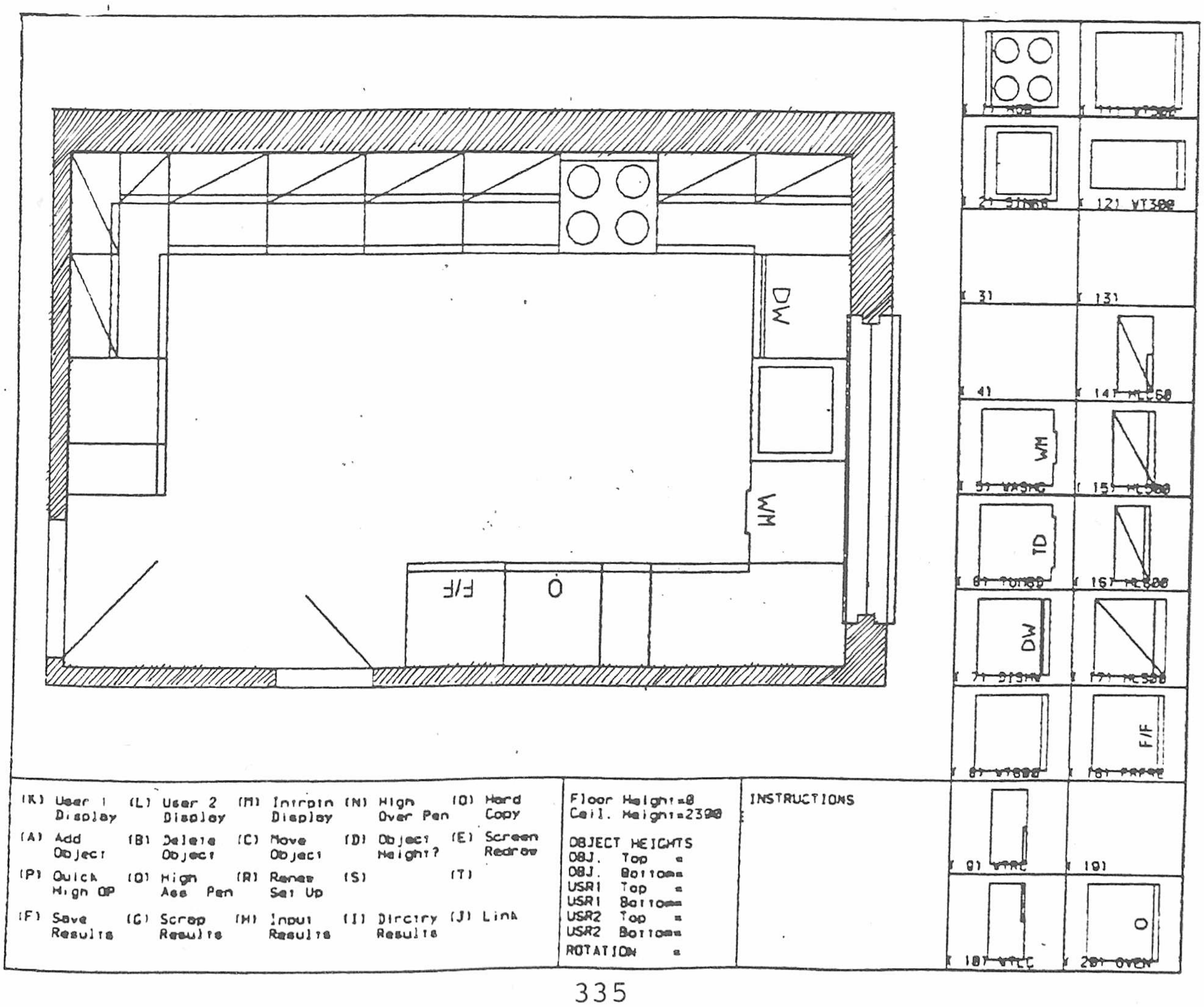
6b


WM
O
F/F
12) VT300
3)
13)
4)
TD
DW
(K) User 1 Display
(L) User 2 Display
(M) Intrpln Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

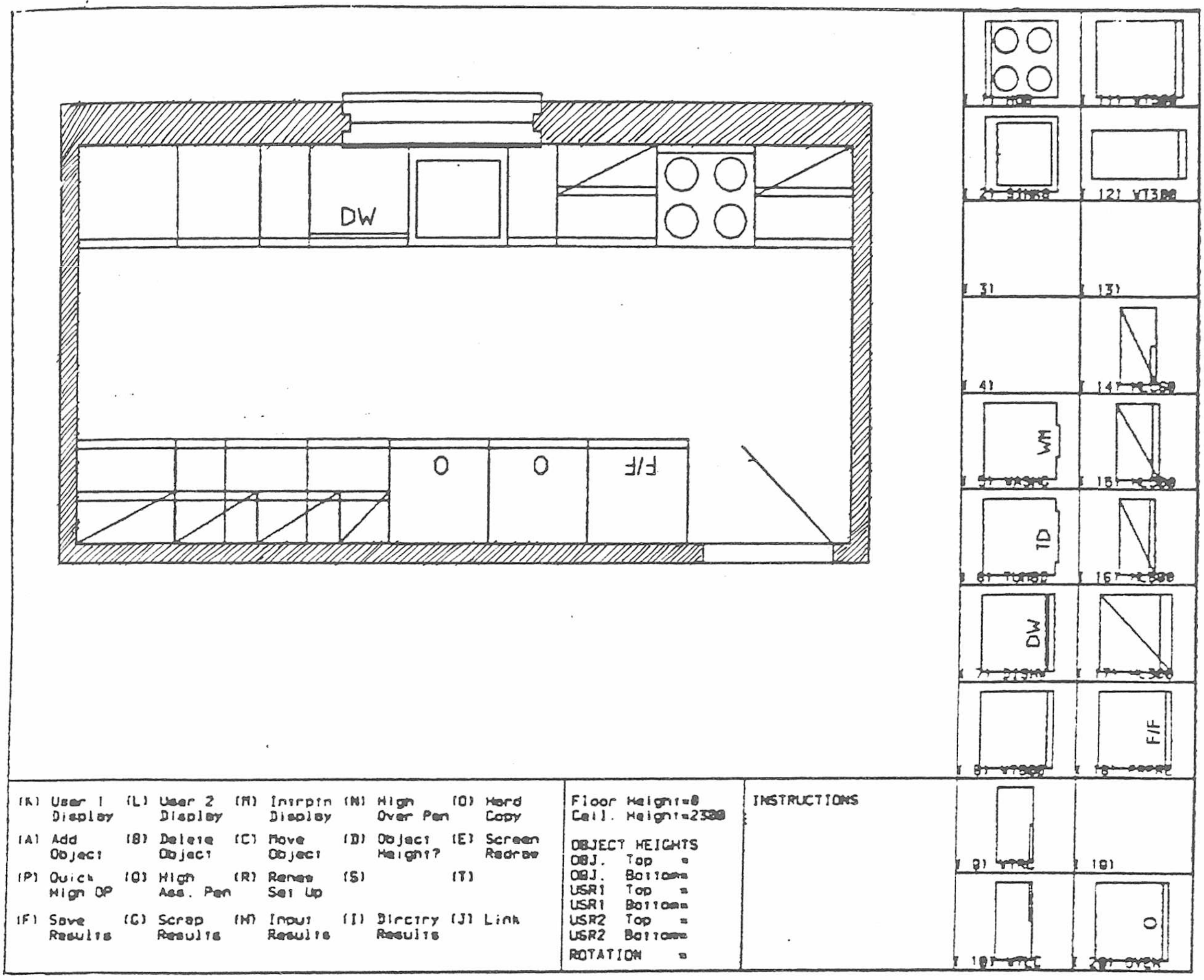
7a


F/F
DW
WM
O
(K) User 1 Display (L) User 2 Display (M) Intrptn Display (N) High Over Pen (O) Hard Copy
(A) Add Object (B) Delete Object (C) Move Object (D) Object Height? (E) Screen Redraw
(P) Quick High OP (Q) High Ass. Pen (R) Renew Set Up (S) (T)
(F) Save Results (G) Scrap Results (H) Input Results (I) Dirctry Results (J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
WM
TD
DW
F/F
O

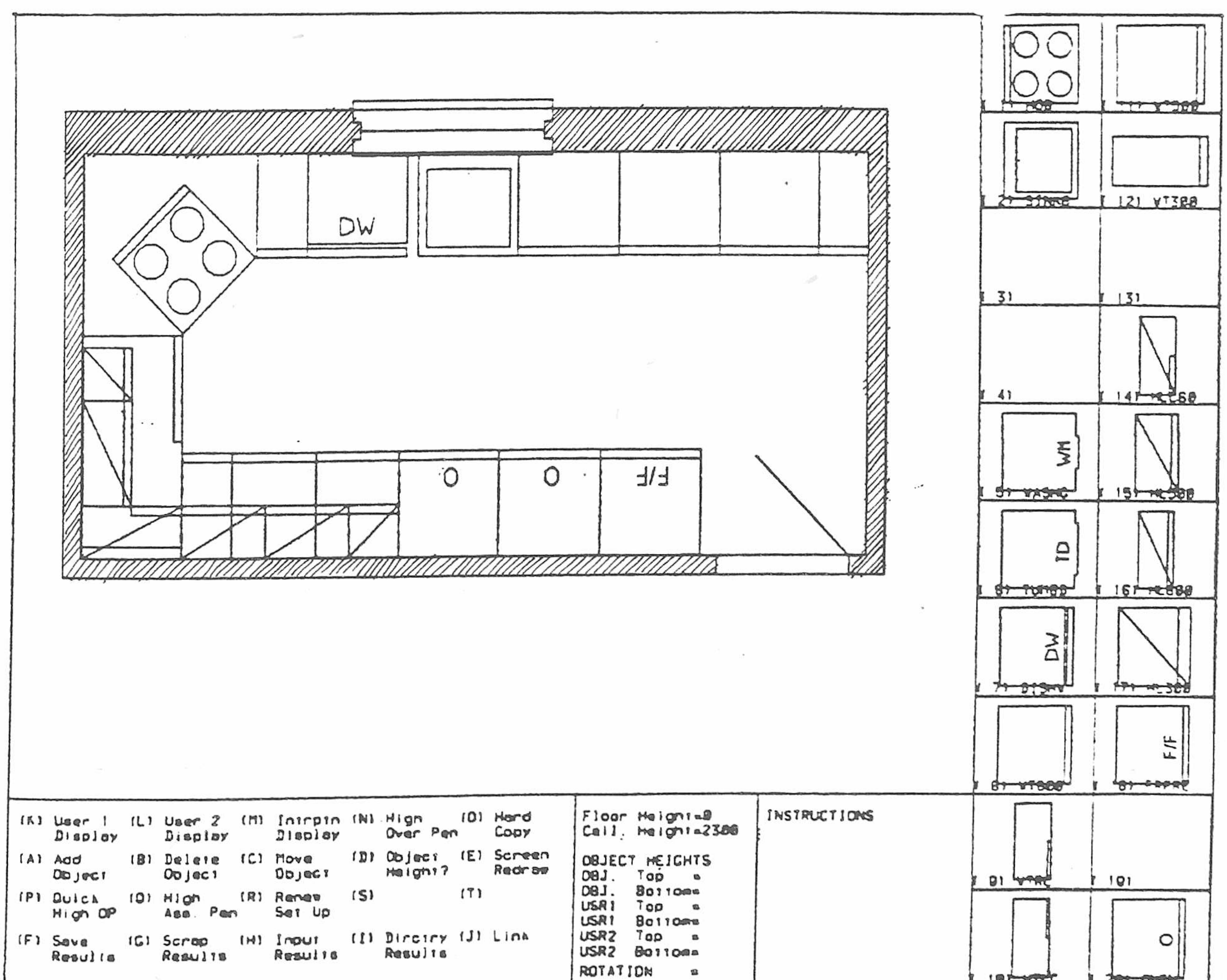
7b


DW
WM
F/F
O
(K) User 1 Display (L) User 2 Display (M) Intrptn Display (N) High Over Pen (O) Hard Copy
(A) Add Object (B) Delete Object (C) Move Object (D) Object Height? (E) Screen Redraw
(P) Quick High OP (Q) High Ass. Pen (R) Renew Set Up (S) (T)
(F) Save Results (G) Scrap Results (H) Input Results (I) Dirctry Results (J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
WM
TD
DW
F/F
O

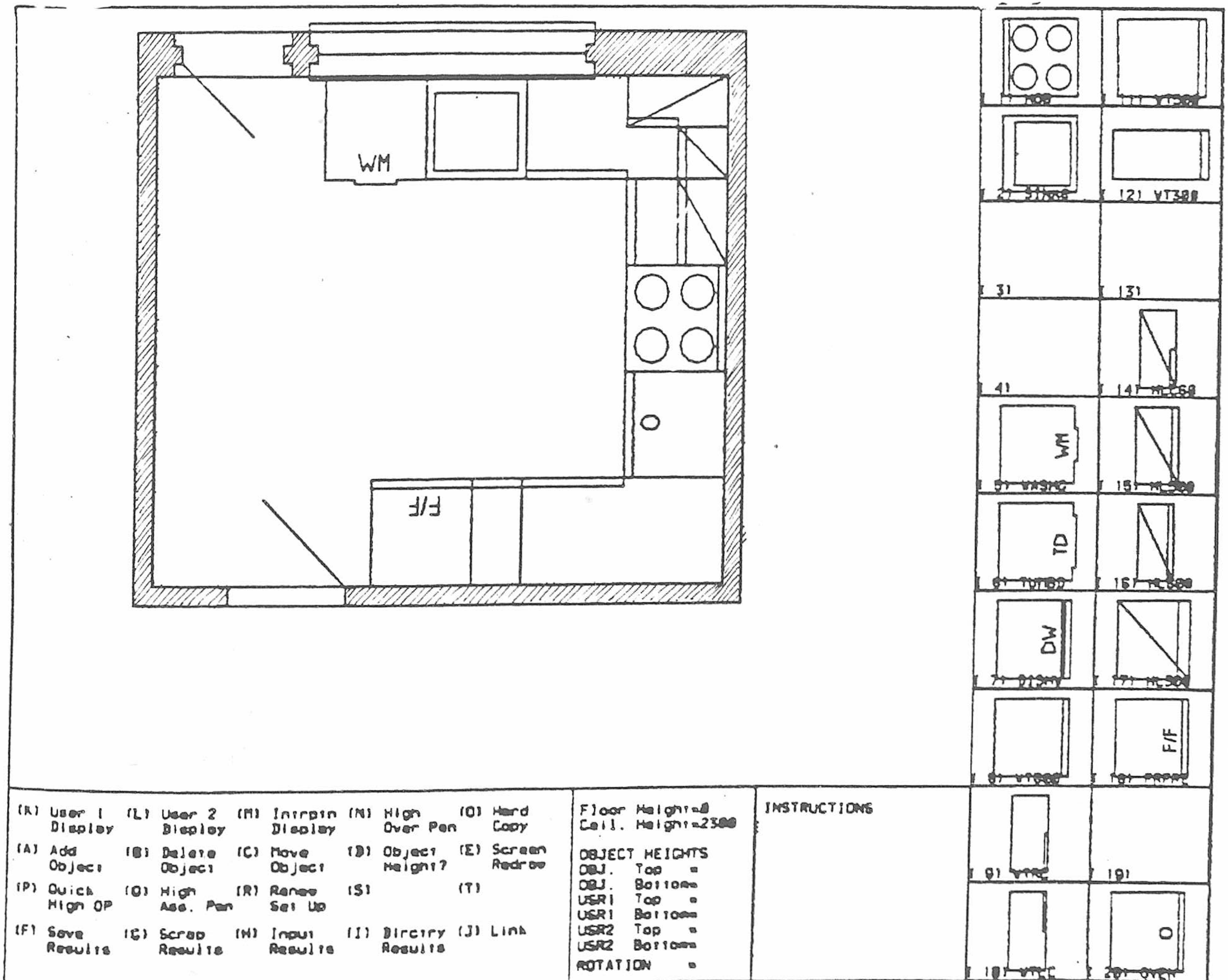
8a


DW
O
O
F/F
WM
TD
DW
F/F
O
12) VT300
3)
13)
4)
9) VTRC
19)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

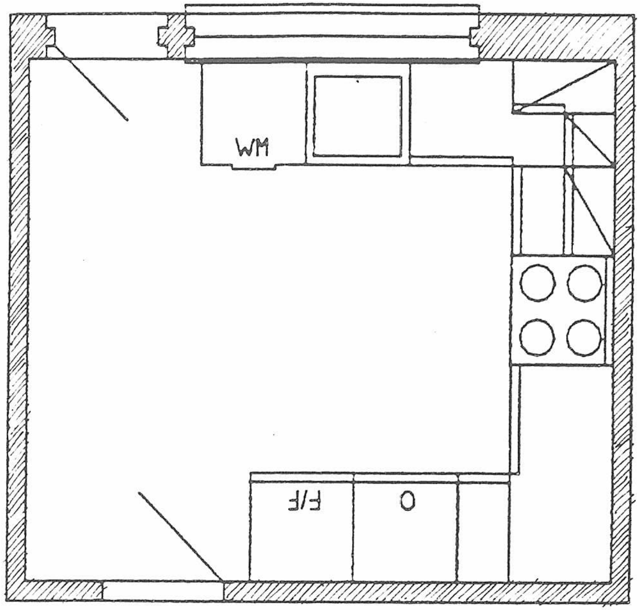
8b

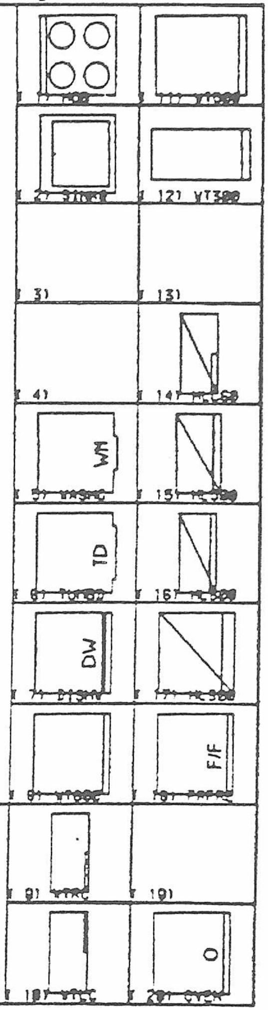

DW
O
O
F/F
WM
TD
DW
F/F
O
12) VT300
3)
13)
4)
9) VTRC
19)
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

9b


WM
F/F
O
(2) VT388
3)
(3)
4)
(4)
WM
TD
DW
F/F
(9)
O
(K) User 1 Display
(L) User 2 Display
(M) Intrpln Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Ass. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS

9a





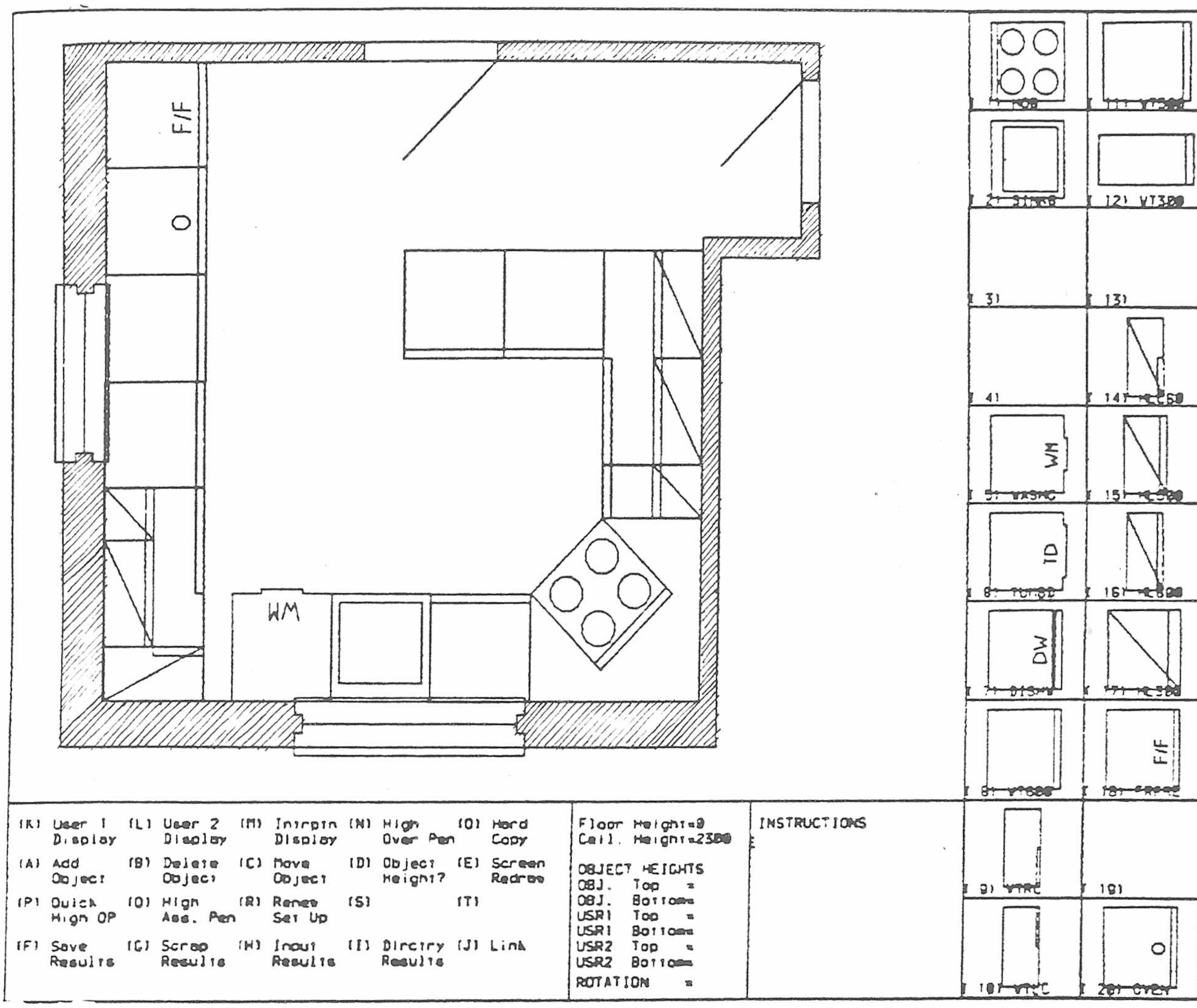
(K) User 1 Display	(L) User 2 Display	(M) Intrpln Display	(N) High Over Pen	(O) Hard Copy
(A) Add Object	(B) Delete Object	(C) Move Object	(D) Object Height?	(E) Screen Redraw
(P) Quick High OP	(Q) High Ass. Pen	(R) Renew Set Up	(S)	(T)
(F) Save Results	(G) Scrap Results	(H) Input Results	(I) Dirctry Results	(J) Link

Floor Height=0
Ceil. Height=2300

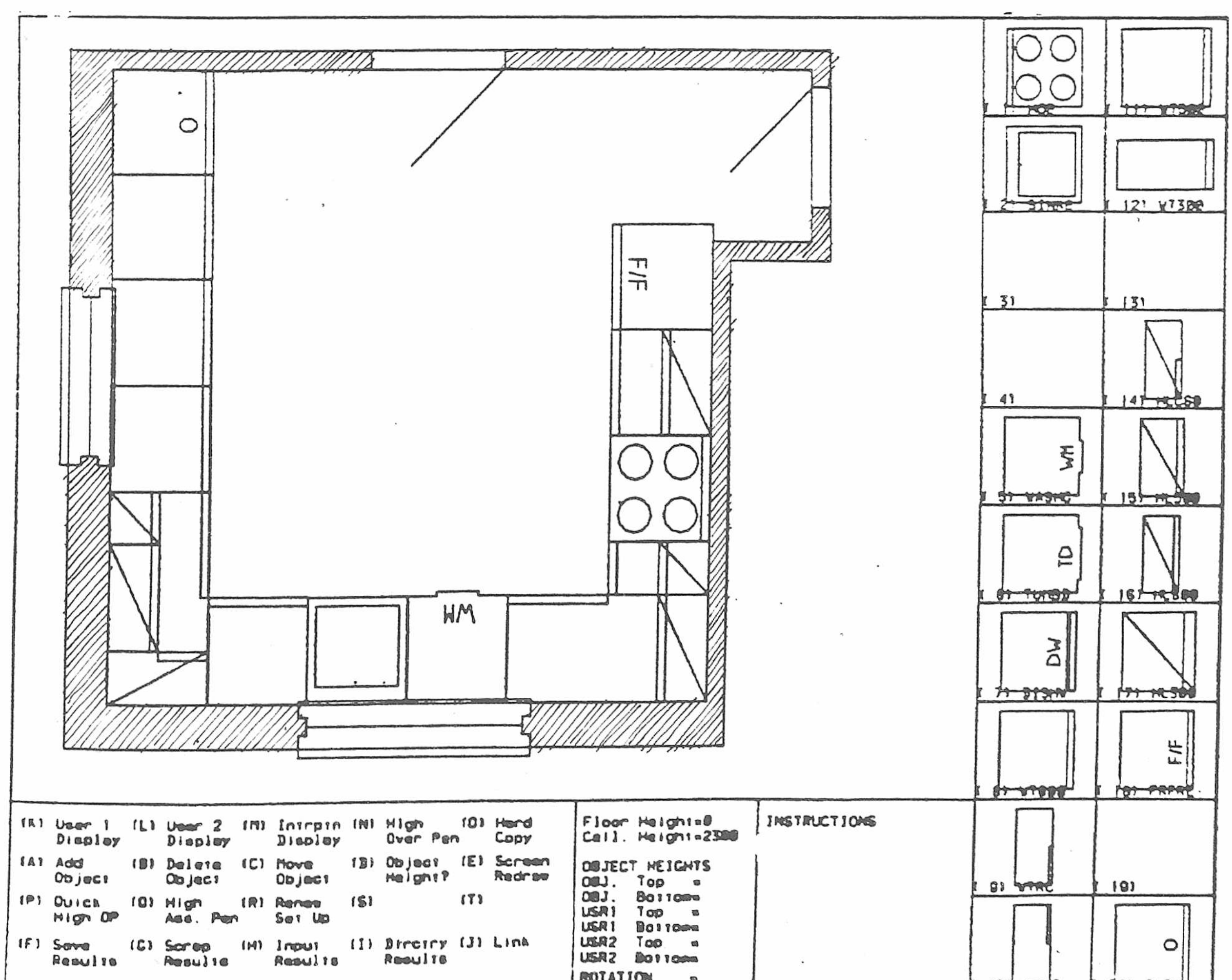
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =

INSTRUCTIONS

10a


F/F
O
WM
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Acc. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
12) VT300
3)
13)
4)
WM
ID
DW
F/F
9) VTRC
19)
O

10b


O
F/F
WM
(K) User 1 Display
(L) User 2 Display
(M) Intrptn Display
(N) High Over Pen
(O) Hard Copy
(A) Add Object
(B) Delete Object
(C) Move Object
(D) Object Height?
(E) Screen Redraw
(P) Quick High OP
(Q) High Acc. Pen
(R) Renew Set Up
(S)
(T)
(F) Save Results
(G) Scrap Results
(H) Input Results
(I) Dirctry Results
(J) Link
Floor Height=0
Ceil. Height=2300
OBJECT HEIGHTS
OBJ. Top =
OBJ. Bottom=
USR1 Top =
USR1 Bottom=
USR2 Top =
USR2 Bottom=
ROTATION =
INSTRUCTIONS
12) VT300
3)
13)
4)
WM
ID
DW
F/F
9) VTRC
19)
O

G.3.2 Results and evaluation

Below are tables of results for each sub-group.

Numeric $ Ranking Results (2nd Quest)	SP penalty	VU penalty	AP penalty	OP penalty	ER penalty.			AP ranking	OP ranking	ER ranking
Kitchen 1a	1·004	1·135	0·101	1·548	0·947					
Kitchen 1b	1·004	1·135	0·051	1·351	0·885			b	b	b
Kitchen 2a	1·016	0·858	0·000	2·527	1·100			a		
Kitchen 2b	1·016	0·858	0·055	2·328	1·064				b	b
Kitchen 3a	1·001	1·240	0·023	1·900	1·041			a	a	a
Kitchen 3b	1·001	1·240	0·033	1·976	1·063					
Kitchen 4a	1·004	1·342	0·000	1·328	0·919			a	a	a
Kitchen 4b	1·004	1·342	0·006	1·464	0·954					
Kitchen 5a	1·000	1·182	0·107	1·757	1·011				a	a
Kitchen 5b	1·000	1·182	0·081	1·837	1·025			b		
Kitchen 6a	1·019	1·120	0·471	2·843	1·363					
Kitchen 6b	1·019	1·120	0·103	2·790	1·258			b	b	b
Kitchen 7a	1·015	1·079	0·203	2·103	1·100					
Kitchen 7b	1·015	1·071	0·091	1·893	1·018			b	b	b
Kitchen 8a	1·038	0·945	0·098	2·419	1·125			a	a	a
Kitchen 8b	1·038	0·984	0·172	3·013	1·307					
Kitchen 9a	1·002	1·251	0·120	2·001	1·095				a	a
Kitchen 9b	1·002	1·251	0·101	2·072	1·107			b		
Kitchen 10a	1·033	1·213	0·551	1·836	1·158					
Kitchen 10b	1·033	1·213	0·376	1·307	0·981			b	b	b

Fig. G.16 Table of computer evaluation of ten pairs of kitchens

Staff	Return1	Return2	Return3	Return4	Return5	Return6	Return7	Return8	Return9	Return10	Return11	Return12		Totals		%	Net %
Kitchen 1a														0		0.0	-100.0
Kitchen 1b	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		12		100.0	
Kitchen 2a	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		12		100.0	+100.0
Kitchen 2b														0		0.0	
Kitchen 3a		✓			✓			✓	✓			✓		5		41.7	-16.6
Kitchen 3b	✓		✓	✓		✓	✓			✓	✓			7		58.3	
Kitchen 4a		✓	✓									✓		3		25.0	-50.0
Kitchen 4b	✓			✓	✓	✓	✓	✓	✓	✓	✓			9		75.0	
Kitchen 5a	✓	✓	✓	✓	✓				✓		✓	✓		8		66.7	+33.4
Kitchen 5b						✓	✓	✓		✓				4		33.3	
Kitchen 6a	✓			X	✓		✓	✓		✓	✓			6	NOTE:- ONE NO CHOICE MADE	54.5	+9.0
Kitchen 6b		✓	✓	X		✓			✓			✓		5		45.5	
Kitchen 7a	✓	✓	✓	✓	✓		✓	✓		✓				8		66.7	+33.4
Kitchen 7b						✓			✓		✓	✓		4		33.3	
Kitchen 8a	✓	✓			✓		✓	✓		✓	✓			7		58.3	+16.6
Kitchen 8b			✓	✓		✓			✓			✓		5		41.7	
Kitchen 9a	✓		✓			✓	✓	✓	✓			✓		7		58.3	+16.6
Kitchen 9b		✓		✓	✓					✓	✓			5		41.7	
Kitchen 10a						✓			✓		✓			3		25.0	-50.0
Kitchen 10b	✓	✓	✓	✓	✓		✓	✓		✓		✓		9		75.0	

Fig. G.17 Table of staff results

Post Part 2	Return1	Return2	Return3	Return4	Return5	Return6	Return7	Return8	Return9		Totals		%	NET %
Kitchen 1a								✓			1		11.1	-77.8
Kitchen 1b	✓	✓	✓	✓	✓	✓	✓		✓		8		88.9	
Kitchen 2a	✓		✓	✓	✓	✓	✓	✓	✓		8		88.9	+77.8
Kitchen 2b		✓									1		11.1	
Kitchen 3a		✓			✓	✓			✓		4		44.4	-11.2
Kitchen 3b	✓		✓	✓			✓	✓			5		55.6	
Kitchen 4a	✓	✓	✓					✓			4		44.4	-11.2
Kitchen 4b				✓	✓	✓	✓		✓		5		55.6	
Kitchen 5a		✓	✓		✓	✓		✓			5		55.6	+11.2
Kitchen 5b	✓			✓			✓		✓		4		44.4	
Kitchen 6a				✓				✓	✓		3		33.3	-33.4
Kitchen 6b	✓	✓	✓		✓	✓	✓				6		66.7	
Kitchen 7a		✓				✓			✓		3		33.3	-33.4
Kitchen 7b	✓		✓	✓	✓		✓	✓			6		66.7	
Kitchen 8a		✓	✓	✓	✓	✓			✓		6		66.7	+33.4
Kitchen 8b	✓						✓	✓			3		33.3	
Kitchen 9a	✓	✓	✓		✓				✓		5		55.6	+11.2
Kitchen 9b				✓		✓	✓	✓			4		44.4	
Kitchen 10a	✓	✓	✓	✓			✓				5		55.6	+11.2
Kitchen 10b					✓	✓		✓	✓		4		44.4	

Fig. G.18 Table of post-part II architecture student results

Fig. G. 19 Table of 3rd year architecture student results

Students Arc 3.	Return 1	Return 2	Return 3	Return 4	Return 5	Return 6	Return 7	Return 8	Return 9		Totals		%	NET %
Kitchen 1a	✓								✓		2		22·2	-55·6
Kitchen 1b		✓	✓	✓	✓	✓	✓	✓			7		77·8	
Kitchen 2a	✓		✓	✓	✓	✓	✓	✓	✓		8		88·9	+77·8
Kitchen 2b		✓									1		11·1	
Kitchen 3a	✓		✓				✓				3		33·3	-33·4
Kitchen 3b		✓		✓	✓	✓		✓	✓		6		66·7	
Kitchen 4a	✓		✓	✓	✓	✓	✓	✓			7		77·8	+55·6
Kitchen 4b		✓							✓		2		22·2	
Kitchen 5a	✓	✓			✓	✓		✓	✓		6		66·7	+33·4
Kitchen 5b			✓	✓			✓				3		33·3	
Kitchen 6a	✓	✓			✓						3		33·3	-33·4
Kitchen 6b			✓	✓		✓	✓	✓	✓		6		66·7	
Kitchen 7a					✓	✓			✓		3		33·3	-33·4
Kitchen 7b	✓	✓	✓	✓			✓	✓			6		66·7	
Kitchen 8a	✓		✓	✓	✓		✓				5		55·6	+11·2
Kitchen 8b		✓				✓		✓	✓		4		44·4	
Kitchen 9a									✓		1		11·1	-77·8
Kitchen 9b	✓	✓	✓	✓	✓	✓	✓	✓			8		88·9	
Kitchen 10a			✓								1		11·1	-77·8
Kitchen 10b	✓	✓		✓	✓	✓	✓	✓	✓		8		88·9	

Arc 2. Students	Return 1	Return 2	Return 3	Return 4	Return 5	Return 6		Totals		%	NET %
Kitchen 1a								0		0·0	-100·0
Kitchen 1b	✓	✓	✓	✓	✓	✓		6		100·0	
Kitchen 2a	✓	✓	✓	✓	✓	✓		6		100·0	+100·0
Kitchen 2b								0		0·0	
Kitchen 3a	✓	✓	✓	✓	✓	✓		6		100·0	+100·0
Kitchen 3b								0		0·0	
Kitchen 4a	✓	✓	✓	✓	✓	✓		6		100·0	+100·0
Kitchen 4b								0		0·0	
Kitchen 5a	✓	✓	✓	✓	✓			5		83·3	+66·6
Kitchen 5b						✓		1		16·7	
Kitchen 6a	✓	✓	✓	✓				4		66·7	+33·4
Kitchen 6b					✓	✓		2		33·3	
Kitchen 7a		✓	✓	✓	✓			4		66·7	+33·4
Kitchen 7b	✓					✓		2		33·3	
Kitchen 8a		✓		✓				2		33·3	-33·4
Kitchen 8b	✓		✓		✓	✓		4		66·7	
Kitchen 9a								0		0·0	-100·0
Kitchen 9b	✓	✓	✓	✓	✓	✓		6		100·0	
Kitchen 10a	✓		✓		✓	✓		4		66·7	+33·4
Kitchen 10b		✓		✓				2		33·3	

Fig. G. 20 Table of 2nd year architecture student results

Fig. G. 21 Table of 3rd year home economics student results

Students HE III	Return 1	Return 2	Return 3	Return 4	Return 5	Return 6	Return 7		Totals		%	NET %
Kitchen 1a									0		0.0	-100.0
Kitchen 1b	✓	✓	✓	✓	✓	✓	✓		7		100.0	
Kitchen 2a	✓	✓	✓	✓	✓	✓	✓		7		100.0	+100.0
Kitchen 2b									0		0.0	
Kitchen 3a									0		0.0	-100.0
Kitchen 3b	✓	✓	✓	✓	✓	✓	✓		7		100.0	
Kitchen 4a			✓	✓	✓	✓	✓		5		71.4	+42.8
Kitchen 4b	✓	✓							2		28.6	
Kitchen 5a						✓	✓		2		28.6	-42.8
Kitchen 5b	✓	✓	✓	✓	✓				5		71.4	
Kitchen 6a	✓			✓	✓		✓		4		57.1	+14.2
Kitchen 6b		✓	✓			✓			3		42.9	
Kitchen 7a					✓	✓	✓		3		42.9	-14.2
Kitchen 7b	✓	✓	✓	✓					4		57.1	
Kitchen 8a	✓	✓	✓	✓		✓			5		71.4	+42.8
Kitchen 8b					✓		✓		2		28.6	
Kitchen 9a									0		0.0	-100.0
Kitchen 9b	✓	✓	✓	✓	✓	✓	✓		7		100.0	
Kitchen 10a			✓						1		14.3	-71.4
Kitchen 10b	✓	✓		✓	✓	✓	✓		6		85.7	

Students HE II	Return 1	Return 2	Return 3	Return 4	Return 5	Return 6	Return 7	Return 8	Return 9	Return 10	Return 11	Return 12		Totals		%	NET %
Kitchen 1a				✓		✓								2		16.7	-66.6
Kitchen 1b	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓		10		83.3	
Kitchen 2a		✓	✓	✓	✓			✓	✓	✓	✓	✓		9		75.0	+50.0
Kitchen 2b	✓					✓	✓							3		25.0	
Kitchen 3a		✓			✓				✓	✓	✓			5		41.7	-16.6
Kitchen 3b	✓		✓	✓		✓	✓	✓				✓		7		58.3	
Kitchen 4a	✓				✓		✓	✓	✓	✓	✓			7		58.3	+16.6
Kitchen 4b		✓	✓	✓		✓						✓		5		41.7	
Kitchen 5a		✓	✓	✓				✓						4		33.3	-33.4
Kitchen 5b	✓				✓	✓	✓		✓	✓	✓	✓		8		66.7	
Kitchen 6a					✓		✓			✓	✓	✓		5		41.7	-16.6
Kitchen 6b	✓	✓	✓	✓		✓		✓	✓					7		58.3	
Kitchen 7a		✓		✓					✓			✓		4		33.3	-33.4
Kitchen 7b	✓		✓		✓	✓	✓	✓		✓	✓			8		66.7	
Kitchen 8a	✓		✓			✓	✓		✓	✓	✓	✓		8		66.7	+33.4
Kitchen 8b		✓		✓	✓			✓						4		33.3	
Kitchen 9a			✓	✓			✓							3		25.0	-50.0
Kitchen 9b	✓	✓			✓	✓		✓	✓	✓	✓	✓		9		75.0	
Kitchen 10a		✓						✓	✓					3		25.0	-50.0
Kitchen 10b	✓		✓	✓	✓	✓	✓			✓	✓	✓		9		75.0	

Fig. G. 22 Table of 2nd year home economics student results

Misc.	Return1	Return2	Return3	Return4	Return5		Totals		%	NET %
Kitchen 1a					✓		1		20.0	-60.0
Kitchen 1b	✓	✓	✓	✓			4		80.0	
Kitchen 2a	✓	✓	✓	✓	✓		5		100.0	+100.0
Kitchen 2b							0		0.0	
Kitchen 3a					✓		1		20.0	-60.0
Kitchen 3b	✓	✓	✓	✓			4		80.0	
Kitchen 4a	✓	✓		✓			3		60.0	+20.0
Kitchen 4b			✓		✓		2		40.0	
Kitchen 5a	✓	✓	✓				3		60.0	+20.0
Kitchen 5b				✓	✓		2		40.0	
Kitchen 6a	✓		✓	✓			3		60.0	+20.0
Kitchen 6b		✓			✓		2		40.0	
Kitchen 7a					✓		1		20.0	-60.0
Kitchen 7b	✓	✓	✓	✓			4		80.0	
Kitchen 8a	✓	✓		✓			3		60.0	+20.0
Kitchen 8b			✓		✓		2		40.0	
Kitchen 9a			✓	✓			2		40.0	-20.0
Kitchen 9b	✓	✓			✓		3		60.0	
Kitchen 10a	✓						1		20.0	-60.0
Kitchen 10b		✓	✓	✓	✓		4		80.0	

Fig. G.23 Table of miscellaneous respondant results

G.4.0 EXPERIMENT 3

This section contains addenda to the third experiment.

G.4.1 Questionnaire design

Overleaf are two examples of the open questionnaire used in this experiment.

OPEN QUESTIONNAIRE

Name:

A.G.42

1. When evaluating how good a domestic kitchen layout is, what is the first thing you consider?

   1. relationship between units
   2. no. of elements
   3. overall room shape

The first thing to be considered when evaluating a domestic kitchen is that certain fitments can be found in the room;- sink, cooker, fridge (washing machine?), together with worktop area and adequate storage.

Key relationships to be considered are fridge/sink/cooker, being the main cooking sequence, and washing machine/sink, since many people do their washing in the kitchen.

2. What would be your ideal room shape for a domestic kitchen?

   1. Square
   2. Rectangular
   3. What size?

Respondant thought room shape didn't matter given good layout; however went on to say that an ideal layout of units would be U-shaped (implies a squareish room).

As regards size it should not be too cramped but at the same time it should avoid the need for excessive walking around the kitchen.

Also thought that a small table (not a breakfast bar) should be in the kitchen for occassional meals – particularly for children.

3. What is the most common mistake in the planning of a domestic kitchen?

   1. wasted space
   2. badly sited units
   3. thro' circulation

Rephrased question to:- What are the mistakes kitchen designers make in the design of kitchens?

(a) No worktop close to kitchen sink.
(b) Doors open in the wrong place, or open the wrong way.
(c) Lack of planning for ~~an~~ cooker extract hood.
(d) Window difficult to open because sink usually infront of it
~ perhaps longer window handles would help.
~ window should be near sink so as can see out when using sink.
(e) Lack of storage space.
(f) Bad lighting on worktop - prefer natural light if possible.
(g) Sink & cooker should be joined by continuous worktop, fridge should be close by but need not be joined.

4. Which appliances bear the most critical relationship to each other?

   1. cooker/sink/fridge
   2. sink/window
   3. negative relationships

(min 300mm)

Key relationship is Fridge-(WT?)-SINK - WT - Cooker - WT?-TABLE

Other key relationships might be plumbed in washing machine beside drier and probably beside sink. Cutlery & dishes storage should be close to the table.

High level units are for infrequent usage - perhaps for bleach & dangerous chemicals (so as to be out of reach of small children. Also perhaps baking ingredients.
No high level units above sink.

Negative relationships would be fridge - cooker - should not be directly adjacent.
Cookers should not be in a draught, because it affects sponges when baking.
Food & cleaning or washing materials should be kept apart.

OPEN QUESTIONNAIRE

Name: A.G.44

1. When evaluating how good a domestic kitchen layout is, what is the first thing you consider?

   1. relationship between units
   2. no. of elements
   3. overall room shape

The main thing to be considered in the evaluation of a kitchen is the relationship between cooker, sink and foodstore (fridge), since these are the three main elements of the working kitchen. The relationship between sink and cooker is particularly important.

The second thing to be considered is the organisation of the kitchen, particularly as regards the amount of worktop length between sink and cooker.

The third thing considered is the amount of walking that use of the kitchen would require. It should be minimal.

2. What would be your ideal room shape for a domestic kitchen?

   1. Square
   2. Rectangular
   3. What size?

The preferred shape for a dining kitchen is a long rectangle, or possibly an L-shape, since this allows the seperation of the dining function.

In a pure working kitchen, a fairly square room would be best.

With any room shape islands make good use of space since with a perimeter layout much floor area is wasted.

3. What is the most common mistake in the planning of a domestic kitchen?

   1. wasted space

   2. badly sited units

   3. thro' circulation

The most common mistake in planning domestic kitchens, particularly in old-houses, is that the kitchen is too small.

Alternatively the room shape may be very poor, a very narrow rectangular kitchen forces a completely linear layout, with consequent increased walking

Also doors may be badly sited, clashing with themselves or with fitments when opened.

Space is often underutilised at corners.

Inadequacy of storage space provision.

4. Which appliances bear the most critical relationship to each other?

   1. cooker/sink/fridge

   2. sink/window

   3. negative relationships

The most critical relationship is cooker/sink. Other important, though less so, relationships are cooker/freezer, washing machine sink, and dishwasher/sink.

The sink should be beside a window, because it's nice to enjoy the view and a lot of time is spent at the sink.

Certain negative relationships were noted.

Fridge hard to cooker.
cooker hard to sink.
cooker hard to wall.

APPENDIX H

CALCULATIONS RELEVANT TO CHECK OF NUMERICAL ACCURACY OF COMPUTER MODEL

CONTENTS

H.1.0 INTRODUCTION

This appendix presents the manual check done on the computer results for a single kitchen layout shown diagramatically in fig.H.1 below.

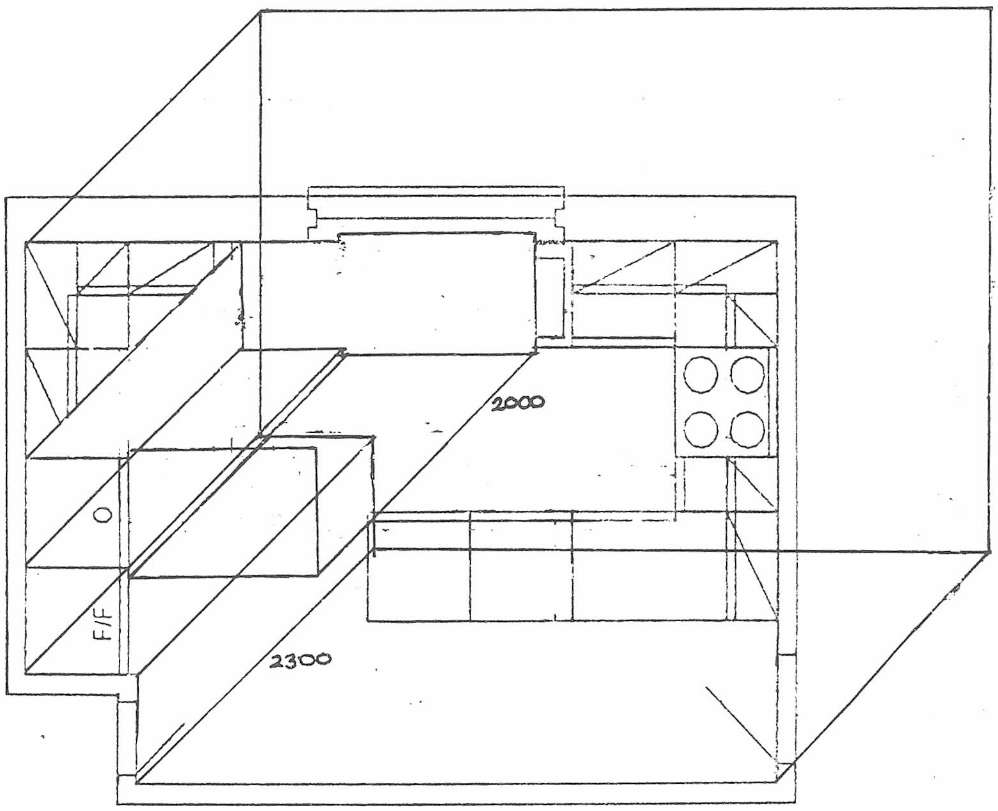


Fig.H.1 Volume of oven housing - Room shape

H.2.0 METHODOLOGY

The mathematical basis of the efficiency factors has already been discussed in earlier chapters. In the following pages, each of the four efficiency values will be assessed individually on a particular sample geometry, the calculations being done manually.

H.3.0 SPACE EFFICIENCY PENALTY

This penalty is given by:-

$$\frac{P.H+2A}{4\sqrt{V/c}.c+2V/c}$$

where:-

- P = perimeter length of room.
- V = volume of room
- c = a constant (set to optimal height for room - in this case 2300mm)
- H = height of room.
- A = area of room

NOTE. throughout this appendix all dimensions are in millimeters.

H.3.1 Substitutions

V=((3000x4400)-(600x650))x2300
=1.281E+7 x 2.3E+3
=2.9463E+10

P=3000+4400+600+2400+650+3750
=1.48E+4

c=2.3E+3

H=2.3E+3

A=((3000x4400)-(600x650))
=1.281E+7

Substituting in;-

$$SP=\frac{(1.48E+4 \times 2.3E+3)+(2 \times 1.281E+7)}{4\sqrt{(2.9463E+10/2.3E+3)}\times 2.3E+3+2(2.9463E+10/2.3E+3)}$$

=5.966E+7/(3.293E+7 + 2.562E+7)

=1.0189968

This matches the computer value of 1.01899687018

H.4.0 VOLUME UTILISATION

This penalty is given by:-

$$\frac{TV}{\sum_{1}^{n}(REVmax)+\sum_{1}^{m}(FEUmax)}$$

where:-

TV = total volume of room
REUmax = maximum user area of a single room element
FEUmax = maximum user area of a single furniture element
n = no of room elements
m = no of furniture elements

Already identified in this example are:-

TV = 2.9463E+10 (from previous section)
n = 3 (two doors and one window)
m = 16

H.4.1 Element volumes

Below the volumes of each of the furniture elements and room elements are identified.

1. FR/FRE
   Volume=(600x600x2000)+(950x1000x2000)
   =2.62E+9

2. OVEN
   Volume=(600x600x2000)+(1100x700x2000)
   =2.26E+9

3. WTRC
   Volume=(600x1200x1350)+(600x1000x1350)
   =1.782E+9

4. WASHG
   Volume=(600x600x1350)+(850x1000x1350)
   =1.6335E+9

5. WT300
Volume=1350x1600x300
=6.48E+8

6. WT500
Volume=500x1600x1350
=1.08E+9

7. WT600
Volume=600x1600x1350
=1.296E+9

8. SINK6
Volume=(600x1200x1350)+(850x400x1350)
=1.431E+9

9. HOB
Volume=(600x600x1350)+(700x600x1350)
=1.053E+9

10. WTLC
Volume=(600x1200x1350)+(600x1000x1350)
=1.782E+9

11. HL600
Volume=1300x600x650
=5.07E+8

12. HLC60
Volume=(300x600x650)+(300x1000x650)
=3.12E+8

13. HL300
Volume=500x1300x650
=2.535E+8

14. HL500
Volume=650x500x1300
=4.225E+8

15. WINDOW
Volume=1500x1050x200
=3.15E+8

16. DOOR (small)
Volume=((400x600)-(100x50)+(100x300))x2100
=5.565E+8

17. DOOR (large)
Volume=((600x800)-(100x50)+(100x400))x2100
=1.0815E+9

H.4.2 Calculation

Substituting in:-

$\sum_{1}^{n}$(REUmax)=3.15E+8 + 5.565E+8 + 1.0815E+9
=1.9530E+9

$\sum_{1}^{m}$(FEUmax)=2.62E+9 + 2.26E+9 + 1.782E+9
+1.6335E+9 + 0.648E+9 + 1.08E+9
+1.296E+9 + 1.431E+9 + 1.782E+9
+1.053E+9 + 0.648E+9 + 1.782E+9
+1.296E+9 + 1.296E+9 + 0.507E+9
+0.312E+9 + 0.2535E+9 + 0.4225E+9
+0.4225E+9 + 0.507E+9 + 0.507E+9
+0.312E+9 + 0.2535E+9 + 0.2535E+9
=2.4358E+10

$$VU=\frac{2.9463E+10}{2.6311E+10}$$

=1.1197978

This value matches that of the computer which is 1.1197978032

H.5.0 OVERLAP PENALTY

This penalty is given by:-

$$\frac{\sum_{1}^{n}\left(OVr.Pr+\sum_{1}^{n-1}(OVns.Pns)\right)+\sum_{1}^{m}\left(OVr.Pr+\sum_{1}^{n}(OVns.Pns)+\sum_{1}^{m-1}(OVms.Pms)\right)}{\sum_{1}^{n}VR+\sum_{1}^{m}VF}$$

where:-

VR = volume of an individual room element
VF = volume of an individual furniture element
m = no of furniture elements
n = no of room elements
OV.P = overlap value for a discrete pair of conflicting elements multiplied by the weighting factor for those elements
r = subscript indicating room edge being used as one of the conflicting elements
ns = subscript indicating that a room element is one of the conflicting elements
ms = subscript indicating that a furniture element is one of the conflicting elements

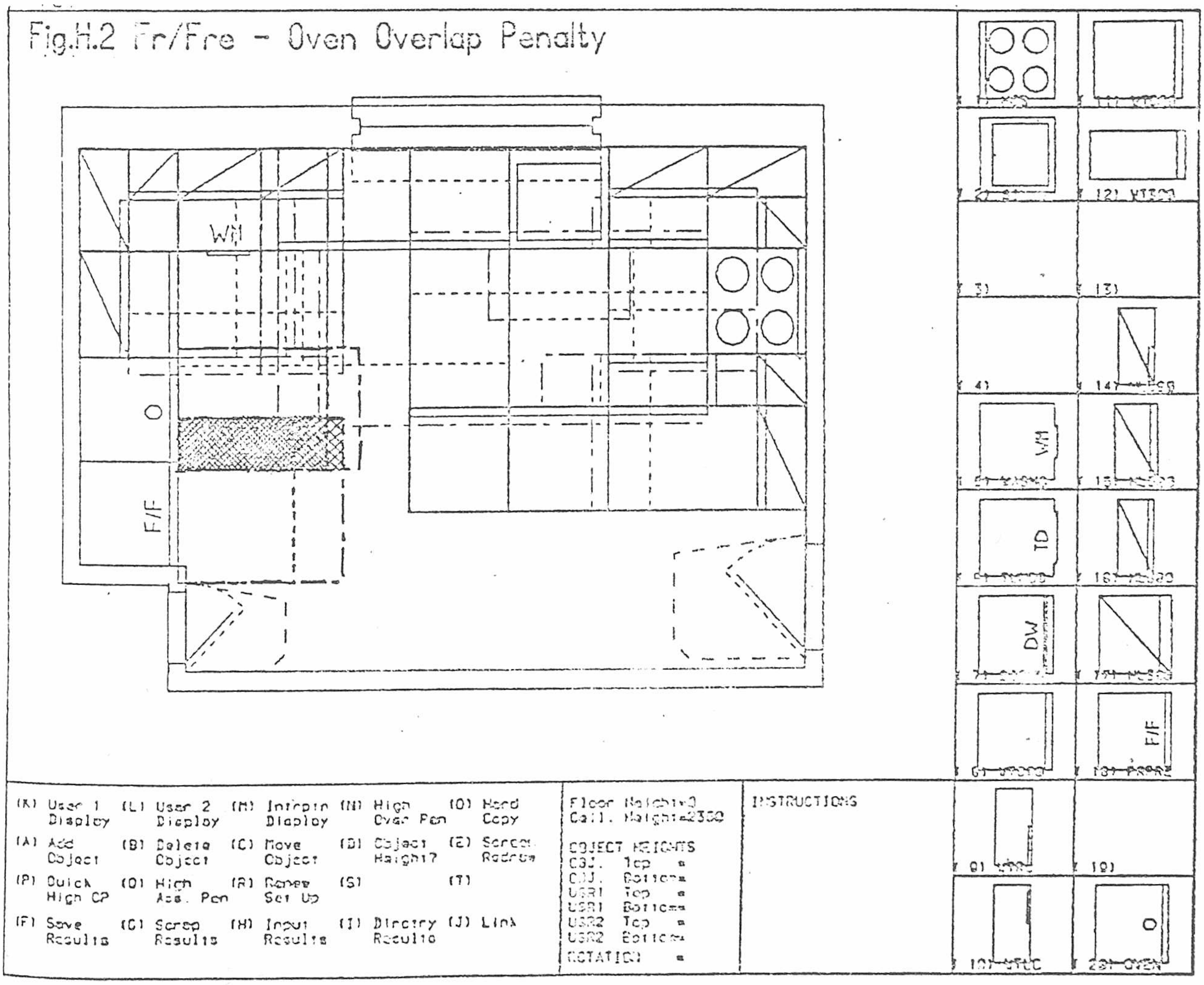


Fig.H.2 Fr/Fre - Oven Overlap Penalty

H.5.1 Intermediate Steps

1. Room elements

    1. No room element overlaps the room edge.

    2. No room element overlaps any other room element.

2. Furniture element n=1 (HLC60)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Cannot overlap any previous furniture element.

3. Furniture element n=2 (HL300)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Overlaps with furniture element n=1

        1. 2-2 overlap=(300x300x650)x5=2.925E+8
        2-3 overlap=(300x300x650)x3=1.755E+8
        3-2 overlap=(300x300x650)x3=1.755E+8
        3-3 overlap=(300x300x650)x2=1.17E+8
        Weighted overlap penalty = 7.605E+8

4. Furniture element n=3 (HL600)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Overlaps with furniture element n=2

        1. 2-2 overlap=(300x350x650)x5=3.4125E+8
        2-3 overlap=(300x350x650)x3=2.0475E+8
        3-2 overlap=(300x600x650)x3=3.51E+8
        3-3 overlap=(300x600x650)x2=2.34E+8
        Weighted overlap penalty=1.131E+9

5. Furniture elements n=4, n=5 (HLC60, HL600)

    1. Do not overlap room edge.

    2. Do not overlap any room element.

    3. Do not overlap any furniture element.

6. Furniture element n=6 (HL300)

    1. Does not overlap room edge.

    2. Does not overlap any furniture element.

    3. Overlaps with furniture elements n=4, n=5

        1. 2-2 overlap=(300x300x650)x5=2.9375E+8
           2-3 overlap=(300x300x650)x3=1.746E+8
           3-2 overlap=(300x300x650)x3=1.746E+8
           3-3 overlap=(300x300x650)x2=1.17E+8
           Weighted overlap penalty=7.605E+8

        2. 2-2 overlap=(300x350x650)x5=3.4125E+8
           2-3 overlap=(300x350x650)x3=2.0475E+8
           3-2 overlap=(300x600x650)x3=3.51E+8
           3-3 overlap=(300x600x650)x2=2.34E+8
           Weighted overlap penalty=1.131E+9

    Total weighted overlap penalty=1.8915E+9

7. Furniture element n=7 (HL300)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Overlaps with furniture elements n=4, n=5

        1. 2-3 overlap=(300x100x650)x3=5.85E+7
           3-3 overlap=(300x100x650)x2=3.9E+7
           Weighted overlap penalty=9.75E+7

        2. 2-3 overlap=(350x100x650)x3=6.825E+7
           3-3 overlap=(600x100x650)x2=7.8E+7
           Weighted overlap penalty=1.4625E+8

    Total weighted overlap penalty=2.4375E+8

8. Furniture element n=8 (HL600)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Does not overlap any furniture element.

9. Furniture element n=9 (HL500)

    1. Does not overlap room edge.

    2. Does not overlap with any room element.

    3. Overlaps with furniture elements n=1, n=3

        1. 2-2 overlap=(350x300x650)x5=3.4125E+8
           2-3 overlap=(500x300x650)x3=2.925E+8
           3-2 overlap=(300x350x650)x3=2.0475E+8
           3-3 overlap=(300x500x650)x2=1.95E+8
           Weighted overlap penalty=1.0335E+9

        2. 2-2 overlap=(350x350x650)x5=3.98125E+8
           2-3 overlap=(350x500x650)x3=3.4125E+8
           3-2 overlap=(600x350x650)x3=4.095E+8
           3-3 overlap=(500x600x650)x2=3.9E+8
           Weighted overlap penalty=1.538875E+9

        Total weighted overlap penalty=2.572375E+9

10. Furniture element n=10 (HL500)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Overlaps with furniture elements n=1, n=3

        1. 2-3 overlap=(200x300x650)x3=1.17E+8
           3-3 overlap=(200x300x650)x2=0.78E+8
           Weighted overlap penalty=1.95E+8

        2. 2-3 overlap=(350x200x650)x3=1.365E+8
           3-3 overlap=(200x600x650)x2=1.56E+8
           Weighted overlap penalty=2.925E+8

        Total weighted overlap penalty=4.875E+8

11. Furniture elements n=11, n=12 (WTRC, WTRC)

    1. Do not overlap room edge.

    2. Do not overlap any room element.

    3. Do not overlap any furniture element.

12. Furniture element n=13 (WTLC)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Overlaps furniture element n=12

        1. 1-3 overlap=(600x100x900)x4=2.16E+8
           3-1 overlap=(600x100x900)x4=2.16E+8
           2-2 overlap=(600x400x1350)x5=16.2E+8
           2-3 overlap=(750x600x1350)x3=18.225E+8
           3-2 overlap=(750x600x1350)x3=18.225E+8
           3-3 overlap=(1100x600x1350)x2=17.82E+8
           Weighted overlap penalty=7.479E+9

13. Furniture element n=14 (WT300)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Overlaps furniture elements n=13, n=12

        1. 2-2 overlap=(300x600x1350)x5=1.215E+9
           2-3 overlap=(300x600x1350)x3=0.729E+9
           3-2 overlap=(300x600x1350)x3=0.729E+9
           3-3 overlap=(300x600x1350)x2=0.486E+9
           Weighted overlap penalty=3.159E+9

        2. 2-2 overlap=(50x600x1350)x5=0.2025E+9
           2-3 overlap=(300x600x1350)x3=0.729E+9
           3-2 overlap=(50x600x1350)x3=0.1215E+9
           3-3 overlap=(300x600x1350)x2=0.486E+9
           Weighted overlap penalty=1.539E+9

    Total weighted overlap penalty=4.698E+9

14. Furniture element n=15 (HOB)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Overlaps furniture elements n=12, n=13 n=14

        1. 1-2 overlap=(50x450x900)x6=1.215E+8
           1-3 overlap=(50x600x900)x4=1.08E+8
           2-2 overlap=(700x450x1350)x5=21.2625E+8
           2-3 overlap=(700x450x1350)x3=12.7575E+8
           3-2 overlap=(700x600x1350)x3=17.01E+8
           3-3 overlap=(700x600x1350)x2=11.34E+8
           Weighted overlap penalty=64.665E+8

        2. 2-2 overlap=(400x450x1350)x5=12.15E+8
           2-3 overlap=(450x700x1350)x3=12.7575E+8
           3-2 overlap=(400x600x1350)x3=9.72E+8
           3-3 overlap=(700x600x1350)x2=11.34E+8
           Weighted overlap penalty=45.9675E+8

        3. 2-2 overlap=(450x50x1350)x5=1.51875E+8
           2-3 overlap=(450x50x1350)x3=0.91125E+8
           3-2 overlap=(600x50x1350)x3=1.215E+8
           3-3 overlap=(600x50x1350)x2=0.81E+8
           Weighted overlap penalty=4.455E+8

        Total weighted overlap penalty=1.150875E+10

15. Furniture element n=16 (WT600)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Overlaps furniture element n=14

        1. 2-2 overlap=(300x50x1350)x5=1.0125E+8
           2-3 overlap=(300x400x1350)x3=4.86E+8
           3-2 overlap=(300x50x1350)x3=0.6075E+8
           3-3 overlap=(300x400x1350)x2=3.24E+8
           Weighted overlap penalty=9.72E+8

16. Furniture element n=17 (WT600)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Does not overlap any furniture element.

17. Furniture element n=18 (OVEN)

    1. Does not overlap room edge.

    2. Does not overlap any room element.

    3. Overlaps furniture elements n=11, n=3, n=2, n=9, n=10

        1. 2-2 overlap=(50x650x1350)x5=2.19375E+8
           2-3 overlap=(50x900x1350)x3=1.8225E+8
           3-2 overlap=(50x650x1350)x3=1.31625E+8
           3-3 overlap=(50x1000x1350)x2=1.35E+8
           Weighted overlap penalty=6.6725E+8

        2. 2-2 overlap=(50x350x650)x5=0.56875E+8
           2-3 overlap=(50x700x650)x3=0.6825E+8
           3-2 overlap=(50x350x650)x3=0.34125E+8
           3-3 overlap=(50x700x650)x2=0.455E+8
           Weighted overlap penalty=2.0475E+8

        3. 1-3 overlap=(100x300x650)x4=0.78E+8
           2-3 overlap=(100x300x650)x3=0.585E+8
           3-3 overlap=(100x300x650)x2=0.39E+8
           Weighted overlap penalty=1.755E+8

        4. 2-3 overlap=(150x500x650)x3=1.4625E+8
           3-3 overlap=(150x500x650)x2=0.975E+8
           Weighted overlap penalty=2.4375E+8

        5. 2-3 overlap=(150x400x650)x3=1.17E+8
           3-3 overlap=(150x500x650)x2=0.975E+8
           Weighted overlap penalty=2.145E+8

    Total weighted overlap penalty=1.50675E+9

18. Furniture element n=19 (FR-FRE)

    1. Overlaps room edge.

        1. RE-2 overlap=(100x50x2000)x6=0.6E+8
           RE-3 overlap=(100x50x2000)x4=0.4E+8
           Weighted overlap penalty=1.0E+8

    2. Does not overlap any room element.

    3. Overlaps furniture element n=18

        1. 2-2 overlap=(700x300x2000)x5=21.0E+8
           2-3 overlap=(700x300x2000)x3=12.6E+8
           3-2 overlap=(900x300x2000)x3=16.2E+8
           3-3 overlap=(1000x300x2000)x2=12.0E+8
           Weighted overlap penalty=61.8E+8

    Total weighted overlap penalty=71.8E+8

19. Furniture element n=20 (WASHG)

    1. Does not overlap room edge.

    2. Does not overlap room element.

    3. Overlaps furniture elements n=11, n=18, n=19

        1. 2-2 overlap=(650x600x1350)x5=26.325E+8
           2-3 overlap=(650x600x1350)x3=15.795E+8
           3-2 overlap=(750x600x1350)x3=18.225E+8
           3-3 overlap=(850x600x1350)x2=13.77E+8
           Weighted overlap penalty=74.115E+8

        2. 2-2 overlap=(100x750x1350)x5=5.0625E+8
           2-3 overlap=(100x750x1350)x3=3.0375E+8
           3-2 overlap=(450x850x1350)x3=15.49125E+8
           3-3 overlap=(450x850x1350)x2=10.3275E+8
           Weighted overlap penalty=33.91875E+8

        3. 2-3 overlap=(50x700x1350)x3=1.4175E+8
           3-3 overlap=(50x850x1350)x2=1.1475E+8
           Weighted overlap penalty=2.565E+8

    Total weighted overlap penalty=1.1059875E+10

20. Furniture element n=21 (SINK6)

    1. Does not overlap room edge.

    2. Overlaps room element window.

        1. W-2 overlap=(550x200x300)x3=0.99E+8
           W-3 overlap=(550x200x300)x2=0.66E+8
           Weighted overlap penalty=1.65E+8

    3. Overlaps furniture elements n=12, n=13, n=14, n=15, n=16, n=17

        1. 2-2 overlap=(125x400x1350)x5=3.375E+8
           2-3 overlap=(125x400x1350)x3=2.025E+8
           3-2 overlap=(125x400x1350)x3=2.025E+8
           3-3 overlap=(125x400x1350)x2=1.35E+8
           Weighted overlap penalty=8.775E+8

        2. 2-2 overlap=(125x150x1350)x5=1.265625E+8
           2-3 overlap=(125x400x1350)x3=2.025E+8
           3-2 overlap=(125x150x1350)x3=0.759375E+8
           3-3 overlap=(125x400x1350)x2=1.35E+8
           Weighted overlap penalty=5.4E+8

3. 3-2 overlap=(50x300x1350)x3=6.075E+7
3-3 overlap=(400x300x1350)x2=32.4E+7
Weighted overlap penalty=3.8475E+8

4. 2-3 overlap=(125x400x1350)x3=2.025E+8
3-3 overlap=(125x400x1350)x2=1.35E+8
Weighted overlap penalty=3.375E+8

5. 1-3 overlap=(100x600x900)x4=2.16E+8
3-1 overlap=(100x600x900)x4=2.16E+8
2-2 overlap=(600x150x1350)x5=6.075E+8
2-3 overlap=(600x500x1350)x3=12.15E+8
3-2 overlap=(600x750x1350)x3=18.225E+8
3-3 overlap=(600x1100x1350)x2=17.82E+8
Weighted overlap penalty=58.59E+8

6. 2-2 overlap=(125x150x1350)x5=1.265625E+8
2-3 overlap=(125x400x1350)x3=2.025E+8
3-2 overlap=(125x150x1350)x3=0.759375E+8
3-3 overlap=(125x400x1350)x2=1.35E+8
Weighted overlap penalty=5.4E+8

Total weighted overlap penalty=8.70375E+9

21. Furniture element n=22 (WT600)

1. Does not overlap room edge.

2. Overlaps room element window.

1. W-2 overlap=(600x200x300)x3=1.08E+8
W-3 overlap=(600x200x300)x2=0.72E+8
Weighted overlap penalty=1.8E+8

3. Overlaps furniture elements n=21, n=17

1. 2-2 overlap=(125x400x1350)x5=3.375E+8
2-3 overlap=(125x400x1350)x3=2.025E+8
3-2 overlap=(125x400x1350)x3=2.025E+8
3-3 overlap=(125x400x1350)x2=1.35E+8
Weighted overlap penalty=8.775E+8

2. 1-3 overlap=(600x100x900)x4=2.16E+8
3-1 overlap=(600x100x900)x4=2.16E+8
2-2 overlap=(600x400x1350)x5=16.2E+8
2-3 overlap=(750x600x1350)x3=18.225E+8
3-2 overlap=(750x600x1350)x3=18.225E+8
3-3 overlap=(1100x600x1350)x2=17.82E+8
Weighted overlap penalty=74.79E+8

Total weighted overlap penalty=8.5365E+9

22. Furniture element n=23 (WT500)

1. Does not overlap room edge.

2. Overlaps room element window.

   1. W-2 overlap=(350x200x300)x3=0.63E+8
      W-3 overlap=(350x200x300)x2=0.42E+8
      Weighted overlap penalty=1.05E+8

3. Overlaps furniture elements n=18, n=19, n=11

   1. 2-3 overlap=(200x100x1350)x3=0.81E+8
      3-3 overlap=(200x450x1350)x2=2.43E+8
      Weighted overlap penalty=3.24E+8

   2. 3-3 overlap=(100x50x1350)x2=0.135E+8

   3. 2-3 overlap=(100x600x1350)x3=2.43E+8
      3-3 overlap=(100x600x1350)x2=1.62E+8
      Weighted overlap penalty=4.05E+8

   Total weighted overlap penalty=8.475E+8

23. Furniture element n=24 (WT300)

1. Does not overlap room edge.

2. Does not overlap any room element.

3. Overlaps furniture elements n=11, n=18, n=19, n=20

   1. 2-2 overlap=(50x600x1350)x5=2.025E+8
      2-3 overlap=(300x600x1350)x3=7.29E+8
      3-2 overlap=(50x600x1350)x3=1.215E+8
      3-3 overlap=(300x600x1350)x2=4.86E+8
      Weighted overlap penalty=15.39E+8

   2. 2-2 overlap=(300x100x1350)x5=2.025E+8
      2-3 overlap=(300x100x1350)x3=1.215E+8
      3-2 overlap=(300x450x1350)x3=5.4675E+8
      3-3 overlap=(300x450x1350)x2=3.645E+8
      Weighted overlap penalty=12.3525E+8

   3. 3-2 overlap=(50x100x1350)x3=0.2025E+8
      3-3 overlap=(50x300x1350)x2=0.405E+8
      Weighted overlap penalty=0.6075E+8

4. 2-2 overlap=(150x650x1350)x5=6.58125E+8
   2-3 overlap=(250x650x1350)x3=6.58125E+8
   3-2 overlap=(150x650x1350)x3=3.94875E+8
   3-3 overlap=(250x1000x1350)x2=6.75E+8
   Weighted overlap penalty=23.86125E+8

Total weighted overlap penalty=5.221125E+8

H.5.2 Substitutions

$\sum_{1}^{n}((OVr.Pr)+\sum_{1}^{n-1}(OVns.Pns))=0$

$\sum_{1}^{m}((OVr.Pr)+\sum_{1}^{n}(OVns.Pns)+\sum_{1}^{m-1}(OVms.Pms))=$

7.605E+8 + 11.31E+8 + 18.915E+8 +
2.4375E+8 + 25.72375E+8 +4.875E+8 +
74.79E+8 + 46.98E+8 + 115.0875E+8 +
9.72E+8 + 15.0675E+8 + 71.8E+8 +
110.59875E+8 + 87.0375E+8 + 85.365E+8 +
8.475E+8 + 52.21125E+8

=7.4799875E+10

$\sum_{1}^{n}VR+\sum_{1}^{m}VF$=2.6311E+10 (from Section H.4.2)

Thus SP=7.4799875E+10/2.6311E+10
=2.8429126

This matches the computer value of 2.8429126

H.6.0 ASSOCIATION PENALTY

This penalty is given by:-

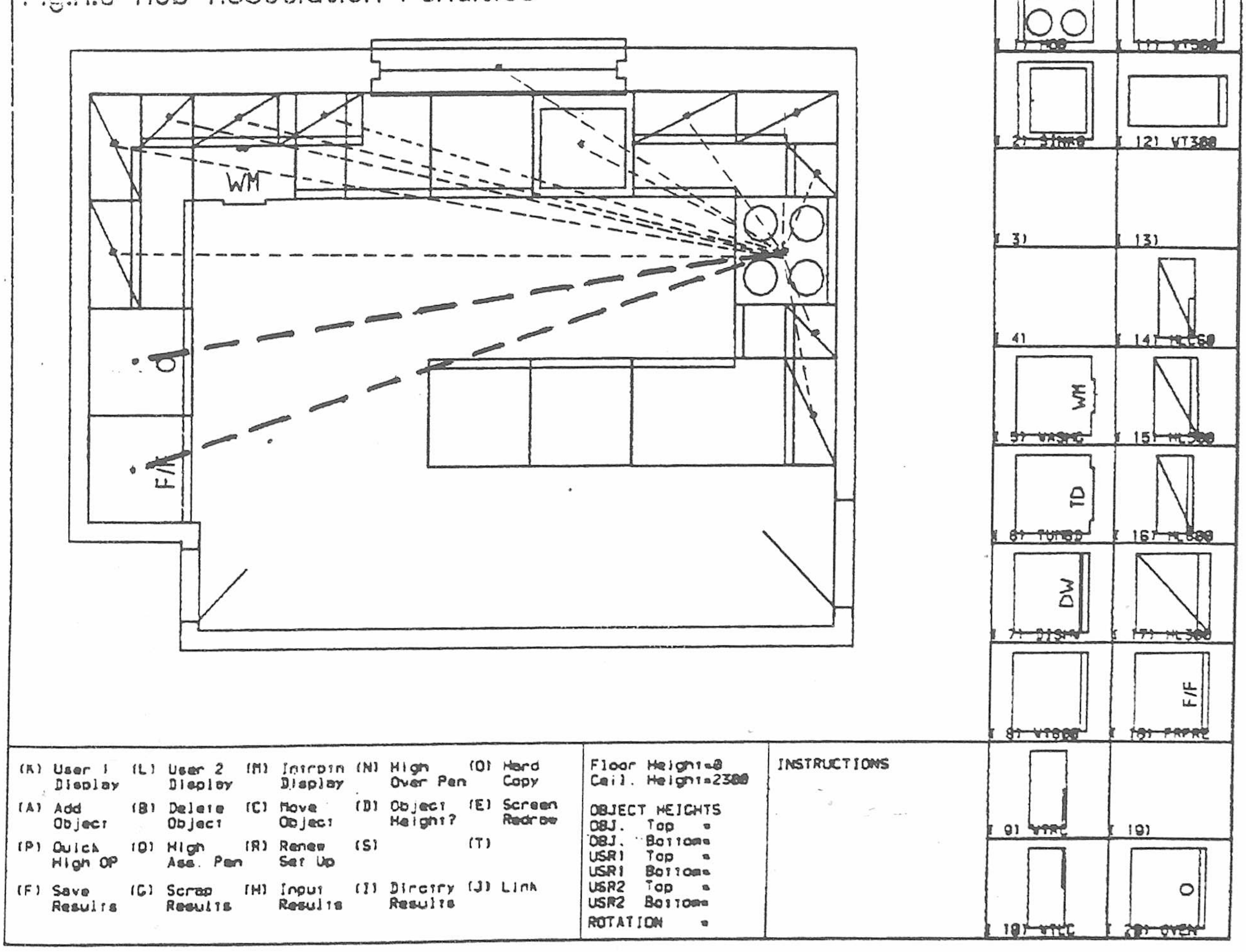
$$\frac{\sum_{i=1}^{n}(\Delta XLap.W)}{\sum_{i=1}^{m}(\Delta Lap/2)}$$

where:-

n = no of failed association pairs
m = no of possible association pairs
$\Delta XLap.W$ = weighted excess length of failed association pair
ΔLap = length permitted between minimum and maximum distance apart for a possible association pair

This is explained with the aid of fig.H.3 where thin dashed lines represent possible association pairs and thicker dashed lines indicate failed association pairs.

Fig.H.3 Hob Association Penalties



H.6.1 Preliminary steps - possible association pairs

HLC60/HOB	=(4500-1365)/2	=1567.5
HLC60/SINK6	=(4500-1365)/2	=1567.5
HL300/HOB	=(4500-1305)/2	=1597.5
HL300/SINK6	=(4500-1305)/2	=1597.5
HL600/HOB	=(4500-1365)/2	=1567.5
HL600/SINK6	=(4500-1365)/2	=1567.5
HLC60/HOB	=(4500-1365)/2	=1567.5
HLC60/SINK6	=(4500-1365)/2	=1567.5
HL600/HOB	=(4500-1365)/2	=1567.5
HL600/SINK6	=(4500-1365)/2	=1567.5
HL300/HOB	=(4500-1305)/2	=1597.5
HL300/SINK6	=(4500-1305)/2	=1597.5
HL300/HOB	=(4500-1305)/2	=1597.5
HL300/SINK6	=(4500-1305)/2	=1597.5
HLC60/HOB	=(4500-1365)/2	=1567.5
HLC60/SINK6	=(4500-1365)/2	=1567.5
HL500/HOB	=(4500-1350)/2	=1575
HL500/SINK6	=(4500-1350)/2	=1575
HL500/HOB	=(4500-1350)/2	=1575
HL500/SINK6	=(4500-1350)/2	=1575
HOB/WINDOW	=(4500-1800)/2	=1350
HOB/DOOR	=(4500-1250)/2	=1625
HOB/DOOR	=(4500-1250)/2	=1625
OVEN/HOB	=(2290-1320)/2	= 485
OVEN/SINK6	=(2880-1320)/2	= 780
OVEN/FR-FRE	=(2820-1200)/2	= 810
FR-FRE/HOB	=(2880-1320)/2	= 780
FR-FRE/SINK6	=(2880-1320)/2	= 780
WASHG/SINK6	=(1520-10)/2	= 755
SINK6/HOB	=(2220-1200)/2	= 510
SINK6/WINDOW	=(2160-10)/2	=1075

Total of possible association pairs=42135

H.6.2 Preliminary steps - penalty pairs

1.	HOB/OVEN	distance apart	=3886.1935
		failure distance	=1596.1935
		penalty factor	=1596.1935x5
			=7980.9675
2.	HOB/FR-FRE	distance apart	=4022.7478
		failure distance	=1142.7478
		penalty factor	=1142.7478x5
			=5713.739

3.	SINK6/FR-FRE	distance apart	=3209.7507
		failure distance	=329.7507
		penalty factor	=329.7507x5
			=1648.7535
4.	SINK6/OVEN	distance apart	=2915.9046
		failure distance	=35.9046
		penalty factor	=35.9046x5
			=179.523
5.	OVEN/FR-FRE	distance apart	=600
		failure distance	=-600
		penalty factor	=600x4
			=2400
6.	SINK6/WASHG	distance apart	=2000.1562
		failure distance	=480.1562
		penalty factor	=480.1562x4
			=1920.6248

The total penalty factor for all failed association pairs is 19843.606

H.6.3 Substitution

$\sum_{i=1}^{n}(\Delta XLap.W)=19843.606$
$\sum_{i=1}^{n}(\Delta Lap/2)=42135$

Thus AP=19843.606/42135
=0.470953

This matches the appropriate computer value.

H.7.0 EFFICIENCY RATIO

The efficiency ratio is given by:-

(SP+VU+OP+AP)/4

where:-
SP = space efficiency penalty
VU = volume utilisation penalty
OP = overlap penalty
AP = association penalty

Substituting in;-

ER = (1.0139968+1.1197978+2.8429126+0.470953)/4
= 1.363165

This matches the computer value.

H.8.0 CONCLUSIONS

The computer model is numerically accurate, within the limits of arithmetic accuracy imposed by the electronic calculator with which it was checked.

APPENDIX I

NOTE ON EXTERNAL SUPPLEMENT TO THESIS

CONTENTS

I.1.0 INTRODUCTION

This appendix indicates additional information pertaining to the research project, but not included in the body of the thesis.

I.1.1 Work forming part of the research project

This sub-section lists reports completed as part of the research project. These reports, generally, give background information to the progress of the research, rather than additional information. The reports listed below are all available for inspection through Scott Sutherland School of Architecture.

1. Report on Mathematical Sciences (B.Sc) module in Computer Studies at RGIT.

2. Report on attendance at relevant conferences.

3. Report on miscellaneous seminars, symposia etc.

4. Progress, annual, and CNAA transfer to Ph.D. reports.

5. Posters presented at ASSA Symposium, 1st. May 1985

6. Paper presented at ASSA CAAD Seminar, Edinburgh 1985

I.1.2 Proposed additional work.

This sub-section lists reports completed,or to be completed which do not form part of the submission for this Ph.D research project, but may be of some interest to researchers following up this research work.

1. Full user manual for KAPABLE.

2. Programmers manual for KAPABLE, includes current listing,and description of data structures.

It is intended that the above reports will form part of the work submitted for The Diploma in Advanced Architectural Studies which the author intends completing in 1986.