Spatial Context As An Aid To Page Layout:  
A System for Planning and Sketching

by Morissa Miller  
B.F.A Rhode Island School of Design  
Providence, Rhode Island  
June 1982

Submitted to the Department of Architecture  
In partial fulfillment of the requirements of the degree of  
Master of Science in Visual Studies at Massachusetts Institute  
of Technology  
June 1984

©Morissa Miller 1984

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

Signature of Author, Morissa Miller, Department of Architecture,  
May 9, 1984

Certified by Muriel Cooper, Associate Professor of Visual Studies  
Thesis Supervisor

Certified by Nicholas Negroponte, Chair, Departmental Committee for  
Graduate Students
Spatial Context As An Aid To Page Layout:
A System for Planning and Sketching

by Morissa Miller

Submitted to the Department of Architecture
on May 9, 1984 in partial fulfillment
of the requirements for the degree of
Master of Science in Visual Studies

Abstract

A graphics system has been developed to
enable a user/designer to create a spatial
context as an aid in page layout. This
system facilitates the planning and
structuring of a complex design work by
providing programs to construct, view and
evaluate serial pages. A user dynamically
defines pages and their constituent elements.
Pages are defined by their dimensions and have
areas delineated by grid structures. The
pages become the structure upon which the
design problem is mapped out. Text and image
areas are defined and modified dynamically.
Series of pages constitute the spatial context
and can be combined, ordered, and arranged.
Using this system a designer has the ability
to view the work as a whole, simultaneously
developing the compositions of distinct pages.
Thus design problems are viewed in their
entirety. Unity is maintained, pacing and
sequencing are analyzed and applied to page
compositions.

Thesis Supervisor: Muriel Cooper
Associate Professor of Visual Studies
Table of Contents

1. Design Process.......................4
   Introduction
   Designer as User
   Methods and Applications
   Structured Systems for
   Page Composition

2. Computers and Design.................21
   Environments
   User Interface

3. Sketching and Planning System.......26
   Context of Work
   Operation of the System
   Conclusion

4. Appendices................................46
   Internal Specifications
   User Manual
   Program Directory

5. Notes........................................63

6. Bibliography...............................64
Introduction

Composing page layouts requires both the design of individual pages and the development of a structure for an entire work. Successfully designed books and other printed materials are easy to read because they have a visual rhythm and a visual consistency. Designers faced with complex problems thus require the ability to make changes on a global level during the design process.

A system has been developed to enable a user/designer to create a spatial context as an aid in page composition. Its primary purpose is to provide the tools that support a holistic approach to design. The design problem is studied in a rough form as the designer records the basic structure of the work [Lawson 1]. The individual parts as illustrated in this project by the pages of a book can be developed simultaneously. This encourages designers to consider unity and pacing in their work. Spatial context describes the area where serial pages are made, arranged and evaluated. Placing pages in a context provides a mechanism for making visual
comparisons. A spatial dimension makes it possible to have many pages accessible at all times. As pages are designed they can be ordered, stacked and rearranged.

A designer uses this system by interactively defining and manipulating the basic components of page composition: pages, grids, image and text representations. As pages are designed they can be arranged in the workspace (figure page 6). Related pages can be grouped together while other pages can be temporarily stacked. Procedures to organize the workspace reflect the different needs of the designer at different stages of the design process.

Though different design applications require different approaches to design, this system is concerned with a working process common to many applications. It is intended to help a designer visually organize material on any page. Using this system a designer can expect to complete thumbnail sketches for an entire book, pamphlet or other 'printed' document. The development of page compositions is the evolution from the abstraction of information to the presentations of
information. This system has been interfaced to a system for page composition where individual pages can be fully articulated. A series of pages can thereby be developed from a rough plan to a fully rendered composition (figures pages 8, 9, and 10). The implementation of this system as a production tool would require that text areas represent text files. Pagination techniques could then be applied to the planning stage. As text areas are modified successive pages could be updated accordingly.
Planning stage of a brochure. The completion of pages one and eight are shown on the following pages.
Page one, above, and page eight, below, are shown here as they appear in Pager. The planned regions are indicated.
The Arts

Art has a net gain and education doubles its small allocation, but the already modest budgets for arts

Page one, above, and page eight, below are shown as finished pages. The original sketches appear at left.
Designer as User

Traditionally a graphic designer worked in a tangible space. Studio, desk and tools were an integral part of the design process. Computer graphic systems have brought many new tools but more fundamentally have brought a new environment for the designer.

Systems for the graphic arts have changed from manual and mechanical production methods to electronic operations. Image processing, color correction, stripping, text setting and pagination constitute part of the production phase of graphic arts. In addition to the production phase, digital processing is currently being applied to the design phase. The software developed here addresses an early stage of the design process - the conceptual and planning phase.

In creating an environment for designers to implement ideas, one must address both the projected scope of the new technology and the established methodologies used by designers. An understanding of their methods, tools and conventions is fundamental for producing a useable tool. The way the system appears to
work must reflect the way the user expects it to work and not the way it performs operations.

"The application of computer techniques to any system forces us to make explicit otherwise implicit procedures, and drives us to examine and investigate our assumptions about the way we make decisions. Thus whatever the ultimate value of computer-aided design systems their ultimate value of development is quite likely to teach us more about designing itself." [Lawson 2]

A graphic designer arranges visual elements as a method of communication. Problems may vary in scope and in application but there are common purposes and processes to the established methodology. The end product, be it poster, book, architectural sign, map or symbol must convey a message that has a specific meaning, unlike a work of art where the meaning can be openly interpreted and the purpose is defined by the artist.

"In design, the problem usually originates not in the designer's mind but with a client or user; someone in need who is unable to solve the problem, or perhaps, even fully to understand it without help. The designer, unlike the artist, is almost always commissioned; the task albeit ill-defined, is brought to him." [Lawson 3]
Methods and Applications

Once having received a stated problem the designer begins to redefine it in an effort to open up a variety of solutions. An early part of the design process involves enlarging the problem. Different approaches to finding a solution are explored and the context is defined in terms of similarities to previous problems and solutions. Later stages of the process involve choosing and editing.

"The designer ... has never resembled Rodin's 'Thinker' who sits in solitary meditation, but has in contrast always externalized his thoughts, not only as an end product in the form of a design, but as an integral part of the process itself in the form of drawings and sketches. The whole purpose of doodles, sketches or models is to act as a kind of additional memory to freeze and store spatial ideas which can then be evaluated and manipulated." [Lawson 4]

Using quick sketches and notes a designer symbolically records ideas. For example: headlines may be notated by rectangular areas while body text by linear markings. The need to generate fast sketches is a strong argument for computer aided design. It becomes easy to produce variations of a design. A designer's energy can then be focused on making visual decisions rather than making
visual records.

Comparing and revising work is basic to any design methodology. Working at a desk a designer has a flexible working process. One or more sketches may be developed at one time. A similar position of papers on a desk may refer to an organizational grouping of similar ideas. A designer may spread out a collection of solutions on a desk or pin them up on a wall in order that the sketches can be viewed simultaneously (figure page 15). By increasing the distance from the work a designer is able to view it in an unfamiliar way. Distance provides a way for a designer to gain objectivity and apply established criteria. Distance also provides a larger field of vision so that many parts may be viewed at once.

The relationship between the designer and the tools affects the finished product. Ultimately a designer with a particular problem may combine two processes to achieve a solution. In the first process the solution is determined by the tools at hand. In the second process a designer begins with the idea for a solution which subsequently
Sketches of magazine pages showing the plan for several issues.
From: *Magazine Design*, by Ruari Mclean
requires identification of the proper tools.

Technological developments have increased the distinction between craftsman and designer. Visual effects can be simulated in the photographic, the printing and now the digital domains. With greater specialization a designer often becomes dependent on other trained personnel for the reproduction of their work. Thus the ability to design becomes dependent on the ability to clearly describe ideas to others. Conventions have been developed to minimize ambiguity in production specification. Similarly, computer systems require users to describe their ideas through the selection of menu buttons. Designers make menu selections in an effort to communicate their ideas. These selections instruct the computer to produce the work. Advances in technology have made the designer a coordinator of various activities.
Structured Systems for Page Composition

Graphic design must communicate information effectively. Clarity is established by presenting information in a hierarchical structure. A book for example has a fixed structure and a sequence imposed by the designer. This sequence implies a spatial organization. Information is referenced by its location in the volume and on the page.

Mathematical and grid based systems have been traditionally employed by designers to produce visually consistent work. In the book Grid Systems, Josef Mueller-Brockmann describes the fixed rules which designers use to structure designs. Typography in particular mandates adherence to a set of rules. The rules govern proportional relationships. For example: the Golden Section provides one method for the division of a page into organizational units. Regularly ordered patterns have produced the finest examples of typography [Mueller-Brockmann 5]. It is important to maintain a balance between consistency and variation. Design requires a structured system to provide flexibility and freedom.
"No systems of ratios, however ingenious can relieve the typographer of deciding how one value should be related to another. He must first recognize the individual value before he can work so that he can judge how much a given ratio can bear." [Ruder 6]

The division of the page into an ordered space is yet another example of a spatial context in which the designer works. Canons and grids are created by the designer as organizational tools. The division of the space (pages of a book) into unit areas creates the guidelines for the positioning of the elements. Subareas have different functions. A consistent location may correspond to the text, titles or page numbers. Combinations of different modules gives the variation necessary for maintaining visual interest. Every element in a book is determined by this invented structure.

Discrete elements can be unified by physically forcing them into predetermined unit modules. It is imperative when developing a grid structure that the design application be considered.

"The typographic grid is a proportional regulator of composition...it is a formal programme to accommodate x unknown items. The difficulty is: to find the balance, the maximum of conformity to a rule with the maximum of freedom. Or: the maximum of constants with the greatest possible
variability." [Gerstner 7]

Structured systems for design have not been limited to book design. Exhibitions, packages, magazines, and newspapers have all employed grid systems in developing their form. Three dimensional graphics require the development of appropriate grid structures. Module units in each dimension serve to organize both individual elements and their spatial relationships (figure page 20). Identity programs use design systems as a marketing tool. Standardization of a corporate symbol is evident in its stationery, office forms, packages, literature, uniforms, and architecture. A consistent image insures public recognition of the firm.
Grid applied to a three dimensional space.
From: Grid Systems, by Josef Mueller-Brockmann
Environments

Attempts to create user friendly environments usually begin by simulating familiar experiences and places. However, the kinds of spaces in which a graphic designer works are varied and cannot be literally translated into a computer space. Display devices inherently lack spatial cues. The boundaries of the screen are finite and define a two-dimensional surface. Computer based information does not have discernible physical properties and yet it can be enormous in quantity and accessed in many different ways. Often informational groupings (pages, chapters, and charts) cannot be displayed at one time. An accepted solution is to divide the data in time, successively displaying screenfuls of information. Unfortunately screen divisions do not always correspond to informational divisions and the user does not have a feeling for their location in the work. Similarly, informational groupings that are recorded on paper may fall on discrete pages. However, the ability to hold and view several pieces of paper simultaneously compensates for this problem.
Spatial references are used to reference and organize large amounts of information. Work in spatial data management at the Architectural Machine Group, MIT [Donnelson 8] and at the Computer Corporation of America [Friedell 9] uses spatial illusions to organize and retrieve specific data. Proximity, similarity, color and proportion are visual cues which elicit intuitive responses when combined with spatial connotations. The appearance of objects close to one another signifies a relationship between them. Depth disparity connotes time or quantitative disparity. For example: the alphabetic ordering of a telephone book for example is easy to use because of its inherent spatial organization. A user of a phone book is able to quickly locate an entry even though he did not previously know its location.

In a book for example there are inherent spatial references. One can feel or see the current location relative to the whole. Developing a work space for designers requires a way to view separate pieces of the work simultaneously. The earlier stages of design require developing several pages at one time. However, working on them consecutively is
awkward. This thesis attempts to minimize this problem by making all pages visible and accessible.
User Interface

Designing is a visual process that is greatly aided by tactile experiences. For example in developing a page composition a designer may physically move small pieces of paper around a page. The act of pushing the elements around helps to generate more solutions. Originally a designer may have had one idea. In moving the pieces to a secondary position he becomes aware of the intermediate positions.

Physical interaction with computer systems is not a "real" experience but rather a simulation of an experience. Continued use of computers in design may change what are now simulated design experiences to real design experiences. This will be especially true as designers work in the absence of traditional tools and materials. New systems are often designed with knowledge of the user's previous experiences. Recent work on user interfaces encourages the use of sensory input devices. Pointing to an object is preferred to describing its cartesian coordinates. Ben Schneiderman has developed a model for
enjoyable interactive systems.

"The central ideas seemed to be visibility of the object of interest; rapid, reversible, incremental actions; and the replacement of complex command language syntax by direct manipulation of the object." [Shneiderman 10].

Although a user of this sketching package travels through a menu structure to choose basic operations, specific operations are defined with the pages themselves.
Context of Work

This project was developed at the Visible Language Workshop, M.I.T. The workshop is a laboratory environment where there are several existing systems for designers and artists. Big Picture is a system for storing and accessing multiple images. Images in the system are stored as tiles. One image can be stored as several tiles. Alternatively, each tile can represent an individual image or page. Tiles are stored consecutively and provide a storage space for the development of page compositions. Different zoom factors allow different numbers of tiles to be viewed at one time. A user can roam through the two dimensional space at different levels of magnification. The farthest viewing location displays the total workspace while the closest viewing location displays one page. An important feature of this system is the ability to view page both individually and in the context of other component pages.

Pager, a package of programs to finalize or produce refined compositions, was developed several years ago at the workshop and is
currently being revised and updated. Using this system multiple pages can be worked on at one time. Type size, leading, letter, and word spacing can be dynamically modified. Using one bit for color information greeking is used to simulate the current text relationships. After alterations are made type is "set" in an anti-aliased font. Images are displayed realistically using 24 bits per pixel. They are modified, cropped, and scaled using bit map representations. In order to display such a high level of detail the entire screen must be used for each page. Though grid structures are defined and used to design multiple pages, visual comparisons between pages are not possible. This impedes the production of a visually cohesive work.
Operation of the System

Basic procedures supported by this system fall into two major categories: that of making/defining objects and that of editing/using objects. The former group includes defining page proportions, grid intervals, grid structures and text and image areas. The latter group includes procedures to edit and manipulate existing objects. Editing procedures require first the selection of the operation and then the identification of the object that operations will be performed upon. Procedures are menu driven, each operation being initiated by a soft button. An existing menu-maker [Henigan 11] directs the flow of control and the skeletal framework of the user interface.

Designing requires making visual comparisons and moving from one page to another. This system has to compensate for having a limited area and large amounts of information to display. Pages have been layered in space. Simple shapes are used to represent image and text areas (figure page 29). Solid rectangular areas represent image. Series of rules represent text. Pages are
Pages in their Spatial Context
presumed to be of a small scale (figure page 31). Symbolic representations of text and image are therefore adequate in detail. If one was to reduce a full page of a book to thumbnail size, the represented information would consequently be reduced to areas of texture. Areas are distinguished by their black value.

The designer begins each session by either calling up previously saved work or by creating a new work. All work is automatically recorded. Communication between the system and the user is based entirely on tablet input. The physical movement of the puck corresponds directly to events for example: pages move as the puck is dragged. Choices are indicated through a menu structure. The menu is arranged so that programs with similar functions are located at the same levels. A selection from the menu may be made at any time and the user always has the option to return to the bottom level. A generalized working process is implied by a left to right reading of the menu choices.

Typically the user would proceed by defining a page. The user can choose any location within the workspace for a page and
Actual size of a sheet used for planning magazines
From: *Magazine Design*, by Ruari Mclean
define both its size and proportion. Once a page is defined it can be used as a rubber stamp for subsequent pages. A series of pages of various proportions can be developed. When the best page proportion is determined it can be duplicated for the entire work.

There are two aids that assist the designer in defining page proportions. One option is to define the size of the page based on a set of predetermined standard proportions, 6 X 9, 8.5 X 11, 9 X 12 and European A series. As the user defines the size of the page its proportion will remain locked to the appropriate format. A second option displays the current proportion of the page as the user modifies its size. A designer can therefore accurately meet specific page requirements.

The grid structure can be defined at any point after a page has been made. Users have the option to define several pages first (of varying or similar size) and then apply grids to one or many pages in any order. A page and a corresponding grid can also be defined before defining subsequent pages. There are two procedures for defining grid structures. The first option is a way to make fast, simple
grids. A number ranging from one to twenty is chosen and a grid containing n modules in x and y is placed on the current page. The second option requires adjusting a ruler alongside a designated page (figure page 34). The frequency of the markings on the ruler are determined by the user. As the user moves a slider the ruler is immediately updated. These increments are then used to create the grid. By pointing to locations on the ruler, lines can be either erased or drawn. Symmetrical and asymmetrical grids can be made. Any grid structure, like any page can be used as a rubberstamp. A grid can therefore be developed for a book and reused to design every page.

There are several options for making duplicate pages. All of the options require identification of the page to be copied. To place a single copy the user has to chose a new location for the page. Double page spreads can be made in two ways, by reflection and duplication of the first page. A location for the first page is chosen anywhere in the workspace. The four possible locations for the second page are determined by the sides of the first page. The user simply points to the
Multiple Pages with Grid Based on Ruler Markings
general location in reference to the first page. The second page is drawn with its grid oriented accordingly. A matrix of multiple copies can also be made. This is useful when the page and grid structure have been developed (figures page 36 and 37). If the book requires the repetition of certain information, such as a heading on every page it is automatically copied onto every page.

Composing individual pages requires the identification of a specific page. Once a page has been chosen text areas or image areas can be placed in the predetermined grid modules. A location within the page is chosen and becomes attached to the nearest grid intersection. A box is then stretched to indicate the size of the area. As it stretches it jumps to the grid intersections. Any number of modules can be enclosed in an area. This allows a designer to quickly map out areas on a page that adhere to the greater grid structure. Text and image areas can be erased by pressing a second button on the puck. The workspace is redisplayed after an area is erased.

Several procedures operate on the page as
Matrix of Pages
Partially Composed Series of Page Layouts
an individual unit. These procedures include moving, deleting, choosing, and identifying pages. Such operations can be executed on pages at any stage. Choosing a page provides the mechanism for changing the status of the current page. The user points to pages which become highlighted and unhighlighted accordingly. When the user stops pointing the last page to be highlighted remains the current page. This procedure is embedded in many operations. It is also available to the user as a way to move through the workspace and visually identify existing pages. To delete a page the user moves through the workspace identifying pages. When the page to be deleted is identified it is eliminated from the record and the display is updated. The move operation is similar to the delete operation. It first requires identification of the page. By moving the puck the user can move the page around the workspace. When its new location is chosen the display is updated. In addition to visual identification, pages can be identified by an associated numerical value. The numeric value represents the page's position in the sequence of total pages. A list of existing page values is displayed.
Pointing to the page value causes the corresponding page to be highlighted. For example: a user can inquire, "Show me page five."

Pages are located by their position in the workspace. The relationship between pages can be modified by moving individual pages. Mechanisms also exist to modify the general characteristics of the workspace. Pages can be displayed in a two dimensional matrix format. The numerical value associated with each page determines its position in the matrix. A second possibility is to display the order of the pages in the matrix form on a secondary display. Both the arrangement of the workspace is maintained and the order of the pages is illustrated. The third method for organizing the workspace involves stacking pages in a linear sequence. Pages only partially overlap so that they can be distinguished from one another (figure page 40). By stacking pages several goals are accomplished. Pages occupy a smaller area of the screen. This provides more room for a work area. Pages that are not currently being worked on may be temporarily placed aside and
Stacked Pages
yet are still easily accessible (figure page 42). Stacking pages also shows the page sequence. A page can be easily located by its location in the stack. The numeric value associated with each page determines its position in the workspace regardless of the arrangement of the workspace. This value can be changed by the user who points to pages and consecutively assigns the page order (figure page 43). A user can easily move between the various configurations of the workspace. Additionally, pages can be rearranged by moving individual pages.

When the planning of pages is finished, page compositions can be refined using Pager, a page composition system. The user points to a page and the associated data is packaged and sent to the required data file. When a designer uses Pager, a list of existing pages is displayed. Pages sent from the sketch system are identified by their file name and their numeric value in the sequence. When the page is loaded, the grid and existing regions defined in the sketch system will be indicated. Specific text files and images can then be resolved.
Pages Moved Out of the Stack
Ordered Pages that Determine the Sequential Order
Conclusion

When I began this project it was my belief that the spatial environment in which a designer works is an integral part of the decision making process. Designing is the process of evaluating visual relationships. Computer systems do not inherently provide a spatial context conducive to such decision making. I have established a way to create a spatial context by creating spatial illusions. Stacked and ordered book pages produce points of reference.

Once located, the compositions for individual pages can be developed and compared with other roughly planned pages in the work. The system as developed to this point addresses only the first stage of the design process, that of producing a rough plan for the work. This stage requires visualization of as much of the work as possible. However, completion of the page compositions require a more detailed visualization of specific text and images.

Many areas in the development of page composition systems remain to be explored.
Designers traditionally use personal notation systems when developing sketches. Such notations affect a final design. Color and texture are also important cues that affect a design. These options need to be incorporated as additional tools for designers. Individual pages which have been designed in detail using other systems such as Pager cannot yet be reinserted into the spatial context and compared with other completed pages. In addition to viewing the entire work in the early stages of the design process it would be advantageous to have such viewing operations throughout the design process. The development of these capabilities will be important steps in providing a useable system for working designers.
Appendix 1
Internal Specifications

At the beginning of every session the user has the option to open a new file or reopen an existing one. In each file there is a copy of the data structure which contains all the information for a work (up to twenty distinct pages). Any number of files containing different work sessions are theoretically possible and are only limited by available disk storage.

A minor structure defining the elements for each page as well as general information about the session compose the data structure (figure page 47). The color of the desk top, the current working page and the number of total pages are examples of global data that are not specific to any page.

An initialization procedure assigns default values for various parameters. There is always one page that is defined to be the current page. This designation is assigned in two instances: one creating a new page and two choosing a page. New pages automatically become the current page. This for example
Data Structure

declare
linkage$page_ptr

pointer external,
fix init(20),

/* see each_page below */

1 page
2 dcolor
2 c_page
2 order_page
2 ordered_space
2 each_page [20],

3 define
3 stacked
3 orginx
3 orginy
3 next

3 proportions,
4 xdim
4 ydim

3 statistics,
4 stat_on_off
3 n_interval
3 v_interval
3 no_areas
3 page_contents[10],

4 text_image
4 ax1
4 ay1
4 ax2
4 ay2

4 dummy4
3 g_lines[300],

4 h
4 v

/ fix,

/ based(linkage$page_ptr),
/ bit(32), /* desk top (background) color */
/ fix, /* current page operatin on */
/ fix, /* for reordering the sequence */
/ fix, /* \0 = free 1 = ordered */
/ /* number of pages */
/ bit(1), /* 1 = on \0 = off */
/ bit(1), /* 1 = stacked \0 = unstacked */
/ fix, /* lower location of page */
/ fix, /* lower location of page */
/ fix, /* points to next page in sequence */
/ /* x dimension of page - width */
/ /* y dimension of page - height */
/ /* interval for ruler */
/ /* interval for ruler */
/ /* no. of filled page contents */
/ /* not sure yet how it will work */
/ /* lower x coord of area */
/ /* lower y coord of area */
/ /* upper x coord of area */
/ /* upper y coord of area */
/ /* to align words on full word */
/ /* grid lines - if a line then = 1 */
/ /* horizontal lines */
/ /* vertical lines */
allows the user to develop a grid for a new page without specifically identifying it. Identifying a page makes it the current page and is necessary in order to perform operations on it.

A page is defined by its lower left and upper right coordinates. The origin of each page is located at the lower right and is stored as screen coordinates in the structure. Additionally the x and y dimensions of the page are stored. Using this combination of information screen locations (through tablet input) can be used to indentify pages. Each page is checked to see if that point lies within its bounds. If the relationship is found to be true then the index to that page is set to be a match. Each additional match that is found becomes the current match thereby locating the closest page. If a page is moved only its origin coordinates needs to be updated as all other page information is stored as relative to the origin.

Grid structures for each page are stored as a minor structure. An array with a high bound of three hundred contains two elementary items for horizontal and verticle values. The location of a grid line is used as the index
into the array. The value one is used to
denote a drawn line and a zero an erased line
for example:

\[
\text{grid_lines}[14].v = 1 \\
\text{grid_lines}[15].v = 1
\]

The number of areas on a page is recorded
and its value is used as an index to the total
number of areas on a page. Coordinates for
each area are relative to the page origin.
The values 1 and 0 are respectively used to
distinguish text and image areas. Areas are
bounded by grid lines. The smallest possible
area is equal to two adjacent grid lines in
both horizontal and vertical dimensions.

Tablet input is used to locate the nearest
module intersection. A box can then be
dynamically dragged to mark the opposite bound
of the area. Grid gravity causes the expanding
box to jump to the nearest grid location. Any
number of modules can be enclosed in an area.

Each page is denoted 'active' by the define
variable. As each page is allocated define is
set to 1. When a page is deleted its status
becomes inactive and define is reset to 0.
Additionally, all data in the deleted page
must be reinitialized. This is necessary so
that the slot can be used for the next page. Allocation for a new page is made by checking the define for all of the pages. The first available (0 value) slot is designated thus garbage collection is done as new pages are allocated.

The interface between this system and the page composition system, Pager, primarily consists of translating existing data into a different format. The user identifies pages which are translated and sent to Pager. Pager requires a data file for each page. The name of the work file and the index of the page in the sketch system are used as the name for the new file. This eliminates additional user input. Though the two systems were developed separately the data structures are similar (figure page 51) enough that some data can be assigned directly. For example the number of regions on a page where there is either text or image and the background color are read from the sketching system and assigned to the structure in Pager. Because the page composition system is concerned with a later phase of design there are variables in Pager that do not have an equivalent counterpart. These variables are assigned a default value
declare

spp

ptr, /* pointer to structure */

1 saved_page
2 name
2 bgcolor
2 unit
2 n[0:1]
2 d[0:1,0:60]
2 nregion
2 r[1],
 3 x1
 3 y1
 3 x2
 3 y2
 3 ox1
 3 oy1
 3 ox2
 3 oy2
 3 line_set
 3 label_name
 3 text_color
 3 background_on
 3 background_color
 3 picture_name
 3 px1
 3 py1
 3 px2
 3 py2

based (spp), /* name of the story */
char(32)varying, /* background color */
bit(32), /* The mesh unit */
fix, /* no. of grid lines x,y */
fix, /* the grid lines */
fix, /* number of regions */
fix, /* the regions */
fix, /* lower x coord of region */
fix, /* lower y coord of region */
fix, /* upper x coord of region */
fix, /* upper y coord of region */
fix, /* original lower x */
fix, /* original lower y */
fix, /* original upper x */
fix, /* original upper y */
char(1), /* f-llush or J-justify left */
char(32)vary, /* which text to use */
bit(32), /* color of text */
bit(1), /* when writing text */
bit(32), /* color of region */
char(32)vary, /* pic on disk to draw */
fix, /* cropping pic. lower x */
fix, /* cropping pic. lower y */
fix, /* cropping pic. upper x */
fix; /* cropping pic. upper y */

Pager's Data Structure
which can later be changed by the user. The remaining data, grid lines, page proportion and text and image areas are read, scaled and then written to Pager.
The package is installed on the system so it can be run from any directory. Type "sketchplan". A new process will be created and the menu invoked. Enter the name of the file. This is the record of the work for a session and can either be an existing or new file. The bottom level of the menu always remains constant. Buttons at this level can be chosen at any time. The following shows available menu choices and describes the function of each option.
PAGE contains the programs to define new pages. NEW makes pages of any proportion. STATISTICS is used in conjunction with new and displays the current proportion. It is a toggle switch which is controlled by two buttons, ON and OFF. OFF is the default mode. STAND(ARD) is used to produce pages of specific proportions. The orientation of the format, HORIZONTAL or VERTICAL, must be chosen as must the specific proportion, 6 X 9, 8.5 X 11, 9 X 12, A SERIES.
The three programs in GRID, SIMPLE, RULER, and LINES provide the means for defining grid structures. In SIMPLE one chooses a number and a grid of that number of modules is produced. RULER displays two rulers alongside the current page. LINES draws and erases lines based on the ruler.

LAYOUT includes the programs that place areas on pages as well as place individual pages. SINGLE copies and places a single page. REFLEC(T) and DUPLIC(ATE) make double page spreads based on a copy of an existing page. ALL produces multiple copies of a designated page. TEXT and IMAGE allow the user to compose and erase areas on the page.
In SERIES are operations performed on pages. CHOSE highlights and unhighlights pages and resets the current page status. UPDATE redipslays the entire screen. DELETE removes a page from the display and the data structure. MOVE refers to a page and is done by pointing to a page and moving the puck to a new location. ORDER contains an additional level of programs, SEE, RE, and STACK. These programs are involved with the order and formation of the pages in the workspace. SEE allows the user to display the pages in their sequential order on the upper monitor. The user can also reassign the page order before it is displayed. RE is similar in usage to SEE but it will displays the page sequence in the matrix formation in the workspace. STACK also is used in the same manner but pages are stacked on one side of the workspace. Only pages specifically numbered in this procedure will be stacked. TO COMP is the interface between Pager and this system by sending the information for designated pages.

<table>
<thead>
<tr>
<th>SEE</th>
<th>RE</th>
<th>STACK</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CHOSE</th>
<th>UPDATE</th>
<th>DELETE</th>
<th>MOVE</th>
<th>ORDER</th>
<th>TO COMP</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PAGE</th>
<th>GRID</th>
<th>LAYOUT</th>
<th>SERIES</th>
<th>UTIL</th>
<th>QUIT</th>
</tr>
</thead>
</table>
UTIL is comprised of utility operations, COLOR, OOFF, and ZAP. COLOR displays a palette for choosing the background color. OOFF erases the overlay planes and ZAP erases the entire screen.

QUIT exits the program.

<table>
<thead>
<tr>
<th>COLOR</th>
<th>OOFF</th>
<th>ZAP</th>
</tr>
</thead>
</table>

| PAGE | GRID | LAYOUT | SERIES | UTIL | QUIT |
Appendix 3
Program Directory

Program Name: ALL_PAGES.pll
Description: Makes many copies of a page including its contents. Calls procedures to choose the current page, redraw the box, redraw the grid and redraw the page contents.

Program Name: BOX2.pll
Description: Draws the page when no specific proportion is required. Uses tablet input for origin and drags box to desired size. Calls procedure to color the page and display page statistics.

Program Name: CHOOSE_MOD.pll
Description: Uses tablet input to locate and outline grid modules on the current page.

Program Name: CHOOSE_SET.pll
Description: Uses tablet input to choose and set the current page. Calls procedures to identify the page, highlight the page, and unhighlight the page.

Program Name: CLOSE.pll
Description: Terminates the pointer to the segment.

Program Name: COLOR.pll
Description: Determines the color of the pages based on the background color.

Program Name: DELETE_AREA.pll
Description: Deletes an area of either text or image from a page. Uses tablet input to locate the area. Calls a procedure to update the screen.
Program Name: DELETE_PAGE.pll

Description: Deletes the current page from the data structure by setting define to "0". Reinitializes all variables associated with the page. Updates the screen. Call procedures to chooses the page and update the screen.

Program Name: DESK_COLOR.pll

Description: Chooses a color from a palette and draws it as the background color. Calls procedure to display palette and get color value.

Program Name: DRAW_GRID.pll

Description: Draws grid lines on the current page. Uses tablet input to determine the closest ruler interval. Calls procedure to find the mod value for line gravity.

Program Name: DRAW_IMAGE.pll

Description: Draws and assigns the image area in the chosen page and module. Calls procedures to choose the page, to choose the module, and to erase areas on the page.

Program Name: DRAW_PAGES.pll

Description: Draws copies of existing pages either as single or double page spreads. Uses tablet input to indicate the location of the pages. Double page spreads either reflect or duplicate the reference grid. Calls procedure to color the page.

Program Name: DRAW_SEQ.pll

Description: Draws the pages in sequential order based on each pages' next. It can display the order on the upper monitor but will not order the workspace. Another possibility is to redraw the workspace in a
linear arrangement. Calls procedures to redraw the box, redraw the grid, redraw the page contents, update the screen, color the page, and order the pages.

Program Name: DRAW_TEXT.pll
Description: Draws and assigns the text area in the chosen page and module. Calls procedures to choose page, to choose the module, and to erase areas on the page.

Program Name: FORMAT.pll
Description: Draws the page when specific proportions are required. Uses tablet input for origin and drags box to desired size while maintaining the chosen proportion. Calls procedure to set page color.

Program Name: HIGHLIGHT_PAGE.pll
Description: Highlights the page designated current page. Calls procedures to identify the page, redraw the box and redraw the grid.

Program Name: IDENTIFY_PAGE.pll
Description: Based on tablet input to identify the chosen page.

Program Name: INIT2.pll
Description: Initializes data structure elements.

Program Name: MOD.pll
Description: Mod function

Program Name: MOVE_PAGE.pll
Description: Uses tablet input to move a chosen page. Calls procedures to chose the page, highlight the page and update the screen.
Program Name: NAME_PAGE.pll

Description: Displays the page numbers and allows pages to be chosen based on the number. Highlights the chosen pages. Calls procedures to highlight and unhighlight the pages.

Program Name: OPEN_SEG2.pll

Description: Initiates and appends the segment for the data structure. New files call procedures to initialize values and update the screen.

Program Name: PALET.pll

Description: Puts up a palette.

Program Name: PASS_PAGE.pll

Description: Chooses a page and sends it to Pager. Scales the page data and initiates and writes it to a file.

Program Name: PSTATS.pll

Description: Used in conjunction with BOX2. Calculates and displays the current proportion of the page as it is being dragged into position.

Program Name: REDRAW_BOX.pll

Description: Given page coordinates and a plane, redraws the page outline.

Program Name: REDRAW_CONTENTS.pll

Description: Given page origin and a plane assigns and redraws the page contents, either text or image areas. Calls procedure to draw the image area.

Program Name: REDRAW_GRID.pll
Description: Given page coordinates and a plane, redraws the page grid.

Program Name: RESEQUENCE.pll

Description: Orders pages by choosing pages and sequentially assigning the order. Un-named pages are assigned a default order value. Calls procedures to choose pages and update the screen.

Program Name: SIMPLE_GRIDS.pll

Description: Draws a grid of n modules in x and y dimensions on the current page.

Program Name: STACKED.pll

Description: Stacks chosen pages on side of workspace and redraws unstacked pages in location in workspace. Calls procedures to redraw the box, redraw the grid, redraw the page contents, update the screen, reorder the pages, and assign the page color.

Program Name: UNHIGHLIGHT.pll

Description: Unhighlights the page designated current page. Calls procedures to redraw the box, grid, and contents.

Program Name: UPDATE_SCREEN.pll

Description: Clears the screen and redraws the current state of all the pages.
Notes


2. Lawson, p.190.

3. Lawson, p.64.

4. Lawson, p.94.


Bibliography


Karsh, Arlene and Murray Philip. 
"Merging Text and Graphics in Printing" 

Lakin, Fred H.  "A Structure from 
Manipulation for Text-Graphic Objects" 
ACM, 1980, Stanford Artificial 
Intelligence Laboratory.

Lawson, Byran.  How Designers Think. 
London: The Architectural Press Ltd., 
1980.

Lewis, John.  Typography: Design and 
Practice.  New York: Taplanger Publishing 

Marcus, Aaron.  "Computer Aided Design: An 
Exploration" Penrose Annual, No. 66, 

Marcus, Aaron.  "Computer-Assisted Chart 
Making from the Graphic Designer's 
Perspective" Technical Report ACM, 
April 1980, Lawrence Berkeley Laboratory.


Morse, Alan.  "Some Principles For the 
Effective Display of Data" ACM, 1979, 
University of Massachusetts, Amherst.

Muller-Brockmann, Josef.  Grid Systems, 
trans.  D.Q.Stephenson.  Teufen, 

Negroponte, Nicholas.  "Books Without Pages" 
IEEE, 1979, Architecture Machine Group, 
M.I.T.

Pike, Rob.  "Graphics in Overlapping Bitmap 
Layers" ACM Transactions on Graphics, 
Vol.2 No.2, April 1983.

Reed, Jack.  "Intergrating CAD/CAM and 
Publishing" Computer Graphics World, 
March 1983.

Rice, Stanley.  Book Design: Systematic 
Aspects.  New York: V.R.R. Bowker Company, 
1978.


