ESL- TH-832
July, 1978

AN ANALYTICAL APPROACH TO COMPUTERIZED NEWS LAYOUT FOR NEWSPAPERS

by

JOHN DANIEL DETREVILLE

Electronic Systems Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

FOREWORD

This report is an unedited version of the doctoral thesis of John D. DeTreville. His research represents further results of continuing investigations by staff members of the M.I.T. Electronic Systems Laboratory of techniques for computer-assisted layout of newspaper pages, including layout of both display advertising and news items.

In an earlier doctoral thesis, Dr. Hsin-Kuo Kan presented an approach to automated news layout which utilizes a library of _a priori_ layouts called "templates". He developed a special graphic language that permits template descriptions to be independent of news-item dimensions, thereby minimizing the size of the computer-stored template library. A comprehensive report describing computer-assisted display ads and news layout based on Dr. Kan's techniques has been published.*

In his research, Dr. DeTreville sought a layout method that would eliminate need for a computer-stored template library of layouts. His technique consists of an _ad hoc_ generation of a series of candidate (legal) layout templates for each page of news. A search of legal templates for a page is then made to determine which among them will yield a layout for the specific news items assigned to the page. The final layout is chosen from the several possibilities. Since, in general, the number of legal templates is far too large to be tested _in toto_, a search strategy is necessary in order to find acceptable layouts quickly through partial testing. In his thesis, Dr. DeTreville examines the merits of several search strategies and draws conclusions about their relative effectiveness. In contrast to Kan's approach, which is built upon a manual formation of templates and a graphic-language description of them, DeTreville's layout algorithms rely on an analytical technique in which templates are generated from solutions to simultaneous inequalities.

As a result of our research in computer-assisted newspaper layout, we recognize that automating the newspaper-layout process poses a major challenge. Clearly, fresh new approaches that go beyond simply automating traditional manual procedures are needed in order to meet, simultaneously, requirements for cost effectiveness, speed, variety, and aesthetically pleasing layouts. We offer, as two possibilities, the approaches of Kan and DeTreville and hope these will serve as stimuli for others. More research and evaluation are needed before a complete operational computerized layout system can be postulated with confidence.

J. Francis Reintjes
Professor of Electrical Engineering

Donald R. Knudson
Staff Member
Electronic Systems Laboratory

---

* J. Francis Reintjes, Donald R. Knudson, and Hsin-Kuo Kan, "Computer-Assisted Layout of Newspapers", ESL-R-778, Electronic Systems Laboratory, M.I.T., December, 1977.

AN ANALYTICAL APPROACH TO COMPUTERIZED NEWS LAYOUT FOR NEWSPAPERS

by

JOHN DANIEL DeTREVILLE

Submitted to the Department of Electrical Engineering and Computer Science on May 5, 1978 in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

ABSTRACT

Although many operations in newspaper production are being rapidly automated, the operation of laying out news items on the individual pages remains labor-intensive and time-consuming. This thesis presents an approach to the design of algorithms for performing this operation automatically.

The news layout problem, which is to produce a page layout given a numerical definition of the news items, is divided into two parts. The first part produces a *template*, which is an abstraction of a layout. The second part, producing a layout from the template, can be accomplished by generating a set of analytical *layout equations* from the template. Their solution, if any, corresponds to a layout based on the template.

The problem of generating a suitable series of templates from the original problem is made easier by the development of a *template grammar*, which defines the set of legal templates. From the generative form of the grammar, all legal templates can be generated incrementally, although they are far too numerous for all of them to be tested individually for correspondence to a layout. Instead, an analogy with game-playing programs leads to the concept of a *layout tree*, which is to be searched for templates corresponding to layouts. Intermediate nodes in the tree correspond to partial templates.

Various search strategies are examined, and it is concluded that the most effective of these is a breadth-first search modified to consider a very limited number of nodes at each ply. These nodes are selected on a semi-random basis, with the selection biased toward partial templates whose corresponding partial layout equations indicate that they are relatively likely to be successful.

Testing of this approach indicates that it can produce layouts of quality comparable to layouts found on inside pages of current newspapers, but requiring less human time and effort.

Acknowledgements

This thesis work would never have been attempted without the encouragement and support of Dr. Thomas F. Jones, who did everything possible to help me get back to work on my doctorate after a lengthy absence from school.

Prof. J. Francis Reintjes originally suggested the area of news layout as one for possible research, and offered to supervise my thesis work long before it became obvious exactly what that work would be. He has been a great help as my thesis supervisor, particularly in regard to helping me present my ideas and results far better than I could have without his assistance.

My readers, Prof. Joseph Weizenbaum and Prof. Donald Troxel, provided many helpful comments and suggestions during the course of this research. The diversity of interests of my committee members was a great help in producing a well-rounded thesis.

Don Knudson served as a day-to-day contact for me to describe my thesis work to, and helped me to work out many of my ideas. He and Mike Tersoff assisted in proofreading early drafts of this thesis. Gina Dallabetta helped in the preparation of experimental test cases.

Prof. Michael Dertouzos kindly arranged for me to receive financial support during my thesis work. Prof. William Martin and Ellen Lewis helped provide the computer facilities used in this research, and generously allowed the use of them over the entire period of the research.

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Office of Naval Research under Contract Number N00014-75-C-0661.

Table of Contents

Abstract ii
Acknowledgements iii
Table of Contents iv
List of Figures vii
List of Tables x

Chapter 1. Introduction 11
1.1 Background 11
1.2 Statement of Problem 16
1.3 Overview of Thesis 19

Chapter 2. Templates 25
2.1 Definition 25
2.2 Template Specification 25
2.3 Layout Equations 30
2.4 More Layout Equations 34
2.5 Example 45
2.6 Summary 47

Chapter 3. Template Grammar 48
3.1 Definitions 48
3.2 Descriptive Grammar 49
3.3 Generative Grammar 52
3.4 Summary 56

Chapter 4. Layout Trees 57
4.1 Definition 57
4.2 Partial Layout Equations 59
4.3 Shape of the Layout Tree 61
4.4 Basic Search Strategy 62
4.5 Summary 63

Chapter 5. Direction of Layout 64
5.1 Canonical Orderings 64
5.2 Bottom-Up Layout 65
5.3 Top-Down Layout 79
5.4 A Comparison 86
5.5 Summary 91

Chapter 6. Layout Tree Searching 92
6.1 Exhaustive Search 92
6.2 Basis for Random Searching 94
6.3 Breadth-1 Search 96
6.4 Breadth-N Search 97
6.5 Breadth-(N_1,N_2) Search 102
6.6 Other Approaches 107
6.7 Summary 107

Chapter 7. Template Esthetics 109
7.1 Basic Procedure 109
7.2 Item Shapes 111
7.3 Importance Ranking 113
7.4 Headline Bumping 113
7.5 Pictures 114
7.6 Summary 116

Chapter 8. Asynchronous Layout 117
8.1 Introduction 117
8.2 Leading Factors 117
8.3 Additional Layout Equations 119
8.4 A Problem 120
8.5 Definition 121
8.6 Evaluation 124
8.7 Summary 127

Chapter 9. Examples of System Operation 128
9.1 The Experimental Implementation 128
9.2 Example Layouts 129
9.3 Summary 138

Chapter 10. Implementation Considerations 143
10.1 Memory Size 143
10.2 Instruction Set 147
10.3 Speed of Execution 148
10.4 Summary 149

Chapter 11. Open Problems 150
11.1 Biases in Move Selection 150
11.2 Tuning 151

11.3 Internal Story Formats 152
11.4 Unusual Item Shapes 153
11.5 Variable-Width Columns 154
11.6 Item Clustering 155
11.7 Outside-In Layout 156
11.8 Incremental Story Assignment 156
11.9 Planning 158
11.10 Criticism 158
11.11 Human Interface 158
11.12 Summary 160

Chapter 12. Conclusions 161

Appendix I. Human Layout Methods 162
I.1 Background 162
I.2 The Experiments 162
I.3 Conclusions 176

Appendix II. Theoretical Considerations 178
II.1 Computational Complexity 178
II.2 Number of Templates 180
II.3 Template Satisfiability 184

Appendix III. Comparison with Longtin's Work 190
III.1 Background 190
III.2 Analysis 191

Appendix IV. Comparison with Kan's Work 193
IV.1 Background 193
IV.2 An Experiment 193
IV.3 Analysis 195

References 197

List of Figures

1-1. Example Page Dummy 12
1-2. Examples of Single-Pyramid and Double-Pyramid Ad-Layout Styles 14
1-3. Example Newshole 17
1-4. Example News Items 20
1-5. Example Layout 21

2-1. Layouts With the Same Abstract Template 26
2-2. Abstract Template for Examples 27
2-3. Example Template 27
2-4. Derivation of Pictorial Form from Matrix Form 27
2-5. Interpretation of Abstract Template Columns 28
2-6. Layout in a Non-Rectangular Newshole 29
2-7. Interpretation of Template Rows 31
2-8. Solution of Layout Equations 33
2-9. Example Non-Rectangular Newshole 35
2-10. Example of Acceptable Headline Placement 36
2-11. Example of Unacceptable Headline Placement 37
2-12. Questionable Layout 38
2-13. Format 1 41
2-14. Format 2 42
2-15. Format 3 43
2-16. Format 4 44
2-17. Final Layout for Example 47

3-1. Illegal Template 48
3-2. Legal Template 49
3-3. Illegal Template Item Shapes 50
3-4. Legal Template Item Shapes 50
3-5. Generation of Example Template 54
3-6. Generation of Example Non-Rectangular Template 55

4-1. Part of the Complete Game Tree for Chess 57

5-1. Bottom-Up Generation of Example Template 66
5-2. Non-Rectangular Newshole 68
5-3. Partial Template 76
5-4. Non-Rectangular Newshole 82

6-1. Average Times for Breadth-N Search for Tech Talk Example 98
6-2. Average Times for Breadth-N Search for Tech Talk Example (Bottom-Up Layout) 99
6-3. Decrease of Amount of Residue in Breadth-10 Search 102
6-4. Decrease of Amount of Residue in Breadth-(10,15) Search 105
6-5. Average Times for Breadth-(N_1,N_2) Search for Tech Talk 106

7-1. Template with Item of Unusual Shape 111
7-2. Possible Exception to the Convexity Rule 112
7-3. Template with Item of Unusual Shape 112
7-4. Example of Headline Bumping 113
7-5. Example of Picture Bumping 114
7-6. Examples of Undesirable Picture Placements 115

9-1. Layout for First Boston Globe Example 131
9-2. Layout for Second Boston Globe Example 132
9-3. Layout for Third Boston Globe Example 133
9-4. Layout for Fourth Boston Globe Example 135
9-5. Layout for Fifth Boston Globe Example 136
9-6. Layout for Sixth Boston Globe Example 137
9-7. First Tech Talk Example 139
9-8. Second Tech Talk Example 140
9-9. Third Tech Talk Example 141
9-10. Fourth Tech Talk Example 142

10-1. Format of Possibility List 145

11-1. Example of Template with Item of Unusual Shape 154
11-2. Example of Template with Item of Unusual Shape 154
11-3. Variable-Width Columns 155

I-1. News Items for First Experiment 163
I-2. Partial Layout for First Experiment 164
I-3. Partial Layout for First Experiment 165
I-4. Partial Layout for First Experiment 166
I-5. Final Layout for First Experiment 166
I-6. News Items for Second Experiment 168
I-7. Partial Layout for Second Experiment 169
I-8. Partial Layout for Second Experiment 170
I-9. Partial Layout for Second Experiment 170

I-10. Partial Layout for Second Experiment 171
I-11. Partial Layout for Second Experiment 171
I-12. Partial Layout for Second Experiment 172
I-13. Final Layout for Second Experiment 172
I-14. Partial Layout for Third Experiment 173
I-15. Partial Layout for Third Experiment 174
I-16. Partial Layout for Third Experiment 175
I-17. Final Layout for Third Experiment 176

II-1. All Possible One-Column Abstract Templates for One Item 181
II-2. All Possible One-Column Abstract Templates for Two Items 181
II-3. All Possible Two-Column Abstract Templates for Two Items 182
II-4. All Possible Three-Column Abstract Templates for Two Items 182
II-5. All Possible One-Column Abstract Templates for Three Items 182
II-6. All Possible Two-Column Abstract Templates for Three Items 182
II-7. All Possible Three-Column Abstract Templates for Three Items 183
II-8. All Possible Four-Column Abstract Templates for Three Items 183
II-9. All Possible Five-Column Abstract Templates for Three Items 184
II-10. Universally Applicable Template 184
II-11. Less Applicable Template 184
II-12. Item Sizes Satisfying Template 186
II-13. More Applicable Template 186
II-14. Item Sizes Satisfying Template 187
II-15. Related Template 187
II-16. Item Sizes Satisfying Template with Leading Restriction 189

III-1. Example of Successful Layout with Longtin's Algorithm 190
III-2. Example of Unsuccessful Layout with Longtin's Algorithm 190

List of Tables

1-I. Numerical Definition of Example Newshole 17
1-II. Numerical Definition of News Items in Figure 1-4 19

2-I. Example Layout Problem 32
2-II. News Item Sizes for Example 45
2-III. Layout Equations for Example 46

4-I. Partial Layout Equations 59

5-I. Example Item Areas 84
5-II. Solution to Layout Equations 85
5-III. Tech Talk Layout Problem 87
5-IV. Bottom-Up Layout Tree Statistics for Tech Talk Example 88
5-V. Top-Down Layout Tree Statistics for Tech Talk Example 88

8-I. Asynchronous Layout Equations for Example 122
8-II. Asynchronous Layout Equations for Example 123

II-I. Number of Templates 181

Chapter 1 - Introduction

1.1 Background

The use of computers in newspaper offices has increased rapidly within the last decade. Computers are now used to capture copy typed in at terminals locally and by wire services, to recall and edit the copy later, to make up pages of the paper interactively, to drive photocompositors, and to perform many other functions. Extensive use of computers helps to reduce the lead time required for newspaper production, thereby allowing the inclusion of later-breaking news, as well as reducing costs and improving the quality of production.

One of the newspaper production operations that has been partially computerized is that of *page layout.* Each page of the newspaper contains some combination of stories and pictures and ads; the problem is to find a layout which specifies where they are to be placed on the page. When this task is performed by hand, a layout person uses pencil and paper to produce a page dummy such as the one shown in Figure 1-1.

An approach which has been used in computerizing page layout is to replace the piece of paper with a display screen and the pencil with a light pen. This is the approach currently used in commercially available systems, as well as in the once-proposed NSDG system [NSDG 73].[1]

A computer implementation of some facet of newspaper production can be judged successful to the extent that it allows the computer to perform a task formerly requiring human effort, and allows it to do so either faster or better than the person. Using such a criterion, we can judge the "scratch-pad"

---

1. References inside brackets are listed in the References section at the end of the report.

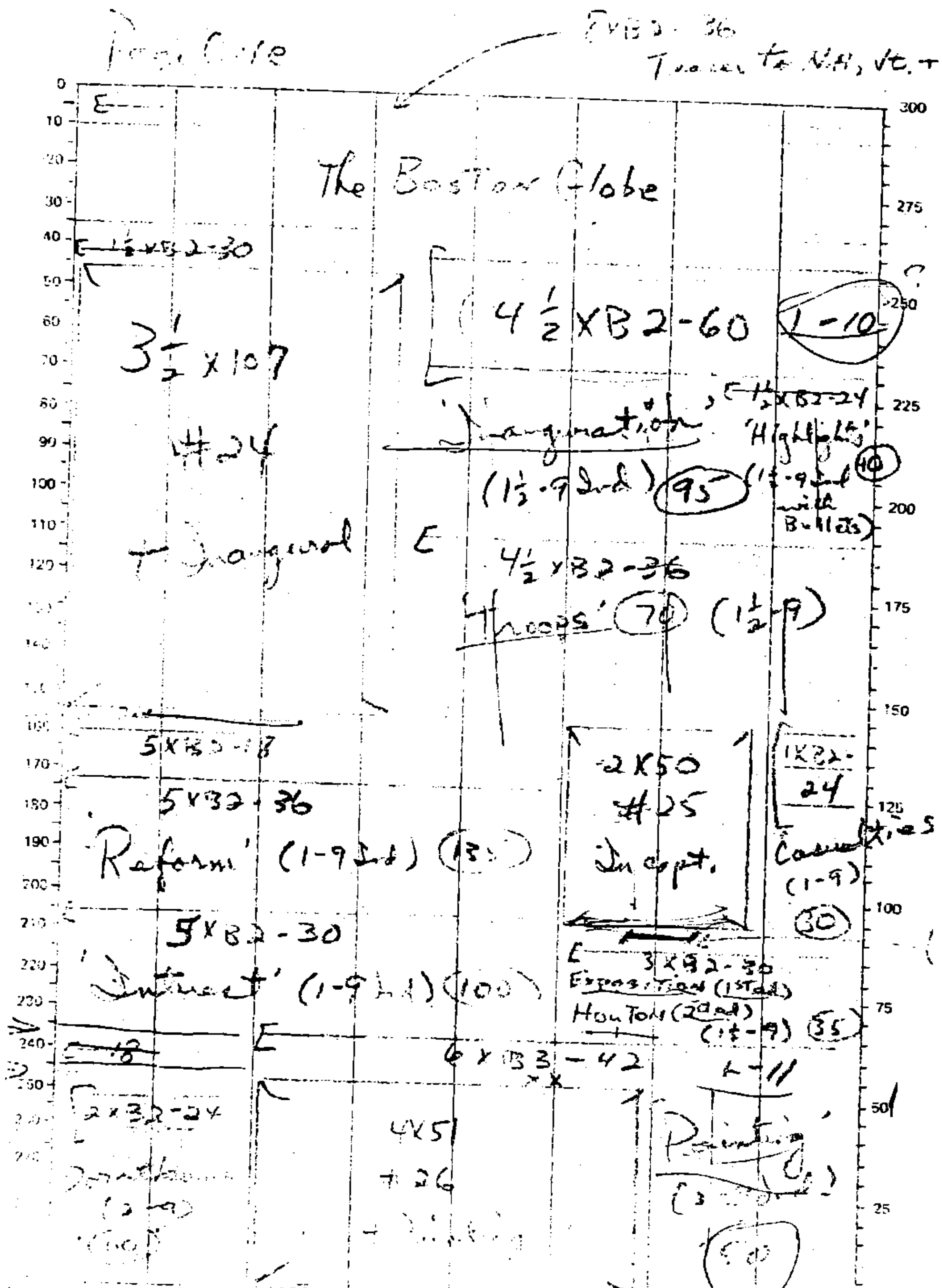


Figure 1-1. Example Page Dummy.

implementation of news layout to be inherently unattractive on both counts, since the layout person will do essentially the same work as in the manual case, and will take essentially the same amount of time doing it, but now requires a computer to drive the display.

Another approach would be for the computer to produce page layouts automatically. In such a system, the layout person would instruct the system to lay out each of the pages of the paper, in turn. It might still be necessary or desirable for the layout person to lay out some pages by hand, such as the front page, but it should be possible for the less-important and less-demanding inside pages to be laid out mechanically. Such a procedure would have the advantage of greatly reducing the amount of human effort required for page layout by having the computer be able to do most of the work.

Layout persons normally divide the problem of page layout into two separate problems: ad layout and news layout. This approach has been found practical because of the different nature of the layout problems for ads and for news. Since the set of ads to be included in the newspaper is typically known long before the set of stories and pictures, the ads are laid out first; this involves decisions on the size of the edition (the greater the number of ads, then typically the greater the number of pages), the size of each of the departments, and the assignment of ads to the pages coupled with the placement of the ads on each individual page.

A computer program for ad layout was designed by Kan [Kan 73]. His program took as inputs the set of ads to appear in the edition, along with a policy file defining the rules to be used in laying out the ads. This program made a decision on the size of the edition and on the assignment of ads to pages, and produced as its output an ad layout for each page. Tests of this approach in an operating newspaper environment [Elkin 74] showed that it was able to produce layouts of comparable quality to those produced by the newspaper staff, and in significantly less time and with far fewer manual steps required.[2] The ad layouts produced were of the *single-pyramid* or *double-pyramid* style, identified as the style most used by newspapers and the style most acceptable to them. Examples of this style are shown in Figure 1-2. All of the ad layouts in the figure are single-pyramid except the last, which is double-pyramid.

After ad layout has been completed, each page not totally taken up by ads

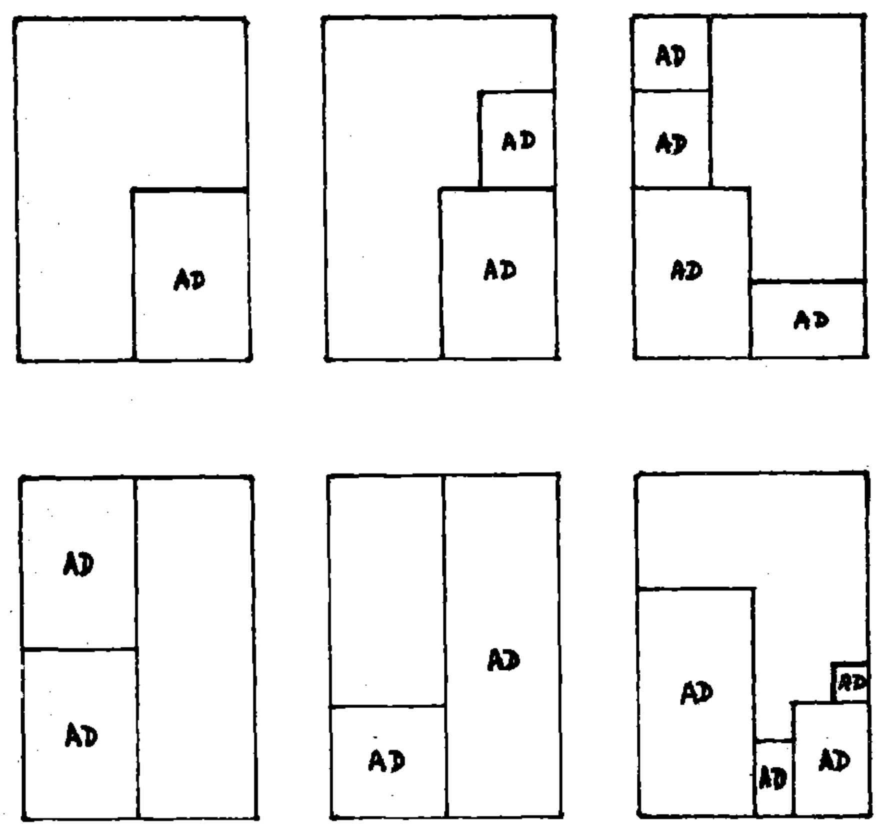


Figure 1-2. Examples of Single-Pyramid and Double-Pyramid Ad-Layout Styles.

2. Although fewer manual steps were required, it was still possible to override any decision made by the layout program, on an interactive basis. This ability was judged a necessity for any such program to be acceptable for use at actual newspaper offices. Ideally, the necessity for such interactions should be kept as small as possible.

has some empty space; this space is called the *newshole*. For each of the departments of the paper (e.g., News, Editorial, Arts, Sports), the sum of the sizes of the newsholes represents the *budget* for the department. Generally, there will be more items available for each of the departments than the budget will allow. The next step in a computerized layout system has been identified [Longtin 72, Reintjes 77] as that of *story assignment*, in which each page is assigned some set of items[3] out of the items available for that department. The sum of the areas of the items assigned to a given page will not exceed the size of the newshole on that page, and will usually be significantly less, because of the discrete sizes of the items. It is reasonable to expect the sum of the areas of the items to represent approximately 95% of the space available in the newshole. The presence of a margin between the actual amount of space needed for the items and the amount of space available makes it easier to produce layouts, since it does not require the items in the layout to fit together exactly, but only within the specified margin. The difference will be made up in the final layout either through the use of *white space* between items, or through the use of *leading* to increase the inter-line spacing in text. (The term "leading" derives from the hot-type method of inserting lead strips between lines of text.)

After story assignment has been performed, each page can be laid out separately. This process involves positioning each item assigned to the page within the newshole in a manner which is in accordance with established newspaper policy for acceptable layouts. Unfortunately, there seems to exist no explicit rules in use by newspaper layout persons which either describe the set of acceptable layouts, or which explain how to achieve such layouts. Indeed, when a layout man at one newspaper was asked what rules he used to

3. In this report, the term *item* is used to denote not only stories, possibly containing dependent pictures, but also independent pictures. A more complete definition of an item is given in the next section.

lay out pages, he replied that he did not know, but would appreciate being told if someone could find out. Layout persons seem to be guided by experience and instruction in such matters.

1.2 Statement of Problem

This thesis is concerned with the problem of *news layout,* which is the problem of placing news items on the pages of the News departments of the paper. In most newspapers, there are certain differences in style among the various departments, but the News departments are prototypical. A procedure for laying out news items, once achieved, can be adapted to other departments.

Each page to be laid out is considered as a separate layout problem. The inputs to the layout procedure are a definition of the newshole, and a definition of the news items assigned to the page. Generally, the number of news items on a page is relatively small, as can be seen by an examination of a typical newspaper. It is reasonable to consider approximately a dozen news items as being a practical upper limit on the number of news items assigned to a page; the number will usually be far less.

The newshole is defined by the number of columns in the newshole, and the height of each. From the nature of pyramidal ad layout, we know that each news column starts at the top of the newshole, and extends downward for some height, terminating either at an ad or at the bottom of the page. For example, the definition of the newshole shown in Figure 1-3 is shown in Table 1-I. In this table, heights are given in *points,* the height standard used in this report. One point is approximately 1/72 inch. We assume that all of the columns have equal width; the problem of variable-width columns is discussed in Section 11.3.

The definitions of the news items are also expressed numerically. We

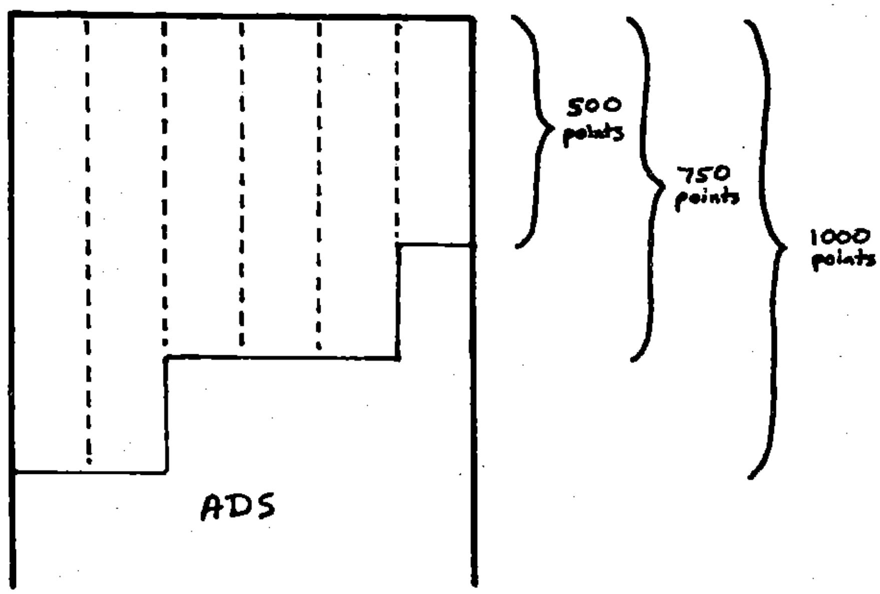


Figure 1-3. Example Newshole.

Table 1-I. Numerical Definition of Example Newshole.

column	height
1	1000
2	1000
3	500
4	750
5	750
6	750

All heights are measured in points.

define a *news item* as either a *story,* which may or may not have some number of associated *dependent pictures,* or as an *independent picture,* which is not associated with any story on the page. We can identify the following components of a news item.

Stories have a *text* component, the definition of which specifies the area of the text for the story. We assume either that the text has already been divided into lines of width equal to the width of a column, so that the number of lines required is definitely known, or else that this quantity has been estimated from the raw length of the story. We can measure the area in *column-points* (i.e., the number of points tall the text would be if it were laid out in one column). For some stories, part of the text may be cut from the final layout if necessary; in this case, it is necessary to know both the amount of text that is available, and the absolute minimum amount that can be run.

Stories also have a *headline* component, whose definition specifies the size of the headline for the story.[4] In some cases, the layout person may wish to specify an exact headline to be used in the layout; in this case, the width of the headline, in columns, and its height, in points, are given. In other cases, a variety of headlines might be possible, and the height and width of each are given. As a limiting case, it is possible to mechanically estimate a probable headline area for a story given its text size, as described by Bracken [Bracken 75]; from this estimated area, a set of headline widths and heights can be generated, and the layout person can write a headline after the space allocated to the headline in the layout is known. In any case, the input to the layout program will be in the form of a set of possible headline widths and heights to be considered.

Still another item component is the *picture*. Pictures may be independent, or may occur as a part of a story. Pictures are defined by their widths, measured in columns, and their heights, measured in points. Pictures may often have a large range of possible heights, since they can be cropped; in such a case, the range of heights is given.

As an example, Table 1-II gives a definition of the news items in the layout shown in Figure 1-4.

---

4. Independent pictures may also have headlines, in rare cases.

Table 1-II. Numerical Definition of News Items in Figure 1-4.

name	text	headline	picture
PCP	1568	1x360,2x180,3x120,4x120,5x120,6x60	-
DiCara	760	1x192,2x96,3x96,4x48	4x343
Chase	532	1x252,2x126,3x84,4x84,5x84,6x42	-
Droney	352	1x144,2x72,3x72,4x36	1x221
School-Board	228	1x48,2x24	-
Chelsea	485	1x144,2x72,3x72,4x36	2x343
Volterra	333	1x72,2x36	-
Correction	114	1x18	-

The variety of headline sizes for each of the stories was generated from the headline width and height used in the layout shown in Figure 1-4, to simulate more closely an actual layout problem.
The height listed for each picture includes the height of the cutline beneath the picture.

The output of the program is a layout for the given news items. This layout will be in the form of a page dummy which can be used to drive the page-makeup procedure. Figure 1-5 shows the page dummy for the page in Figure 1-4. Pictures are denoted by diagonal lines. Headlines are separated from the rest of the story by dotted lines, as are dependent pictures. This graphical notation is the standard used in this report.

In page makeup, following page layout, exact headlines are written and inserted, the text leaded and divided among the columns assigned to the stories, white space inserted, lines separating items drawn, and so forth.

1.3 Overview of Thesis

The basis of the approach taken in this thesis is to divide the news layout problem into two parts. The first part produces not a layout, but an abstraction of a layout called a *template*. The second part transforms a

PCP or angel dust, it's around and deadly

2 suspects shot by police during Dorchester chase

DiCara wins with surprising ease

School board to pick leader today

Droney to seek 5th term as Middlesex County DA

Father of 7 charged in fatal Chelsea fire

CORRECTION

Dukakis names Volterra

Figure 1-4. Example News Items.

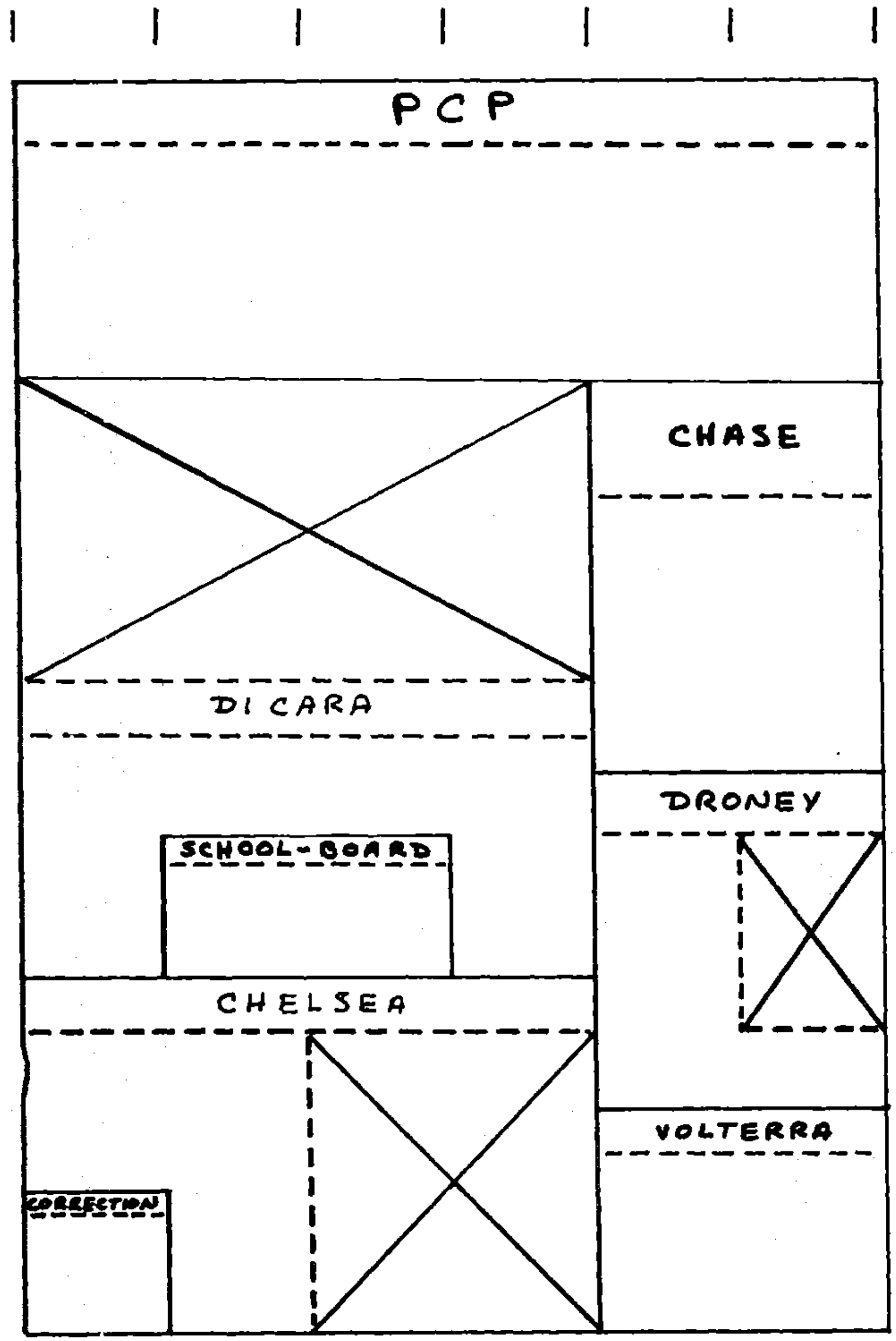


Figure 1-5. Example Layout.

template into a layout, by means of the solution of a set of simultaneous linear inequalities, called *layout equations*, generated from the template (Chapter 2). If the equations based on a given template have a solution, then the template has a corresponding layout, which can be used as the solution to the original news layout problem. Otherwise, another template must be tried.

The problem of how a stream of templates is supplied in this procedure is made easier by the specification of a *template grammar* (Chapter 3). Templates specify the set of geometric interrelationships between the items in the template; some sets of interrelationships can be specified as being grammatical and some ungrammatical, according to a *descriptive grammar* based on the types of layouts commonly used in newspapers. A corresponding *generative grammar* can be used to generate every legal (i.e., grammatical) template.

The operation of the generative grammar for a given layout problem can be modelled by a conceptual *layout tree* (Chapter 4). The terminal nodes of the tree correspond to templates, while the non-terminal nodes correspond to *partial templates*. Just as the validity of a template for a given layout problem can be determined through attempting the solution of a set of layout equations, the validity of a partial template can be partially determined through attempting the solution of a set of *partial layout equations*. The problem of finding a valid template can then be viewed as the problem of searching through the layout tree until a node corresponding to such a template is encountered.

The effectiveness of this search depends on the exact nature of the layout tree. Certain modifications to the generative grammar can greatly increase the probability of a successful search. In particular, requiring that templates be built either upward from the bottom of the page *(bottom-up)* or downward from the top of the page *(top-down)* allows a significant strengthening of the conditions stated in the partial layout equations, allowing earlier detection of invalid paths through the tree; it also greatly reduces the size of the tree (Chapter 5). Moreover, top-down layout gives certain advantages over bottom-up layout, because of the nature of the template grammar.

In any formulation of the generative grammar, the sheer size of the layout tree necessitates the use of highly efficient search strategies to enable the location of a valid template in a reasonable amount of time (Chapter 6). A simple exhaustive search can be ruled out, because of the nature of the layout tree. Instead, a *limited-breadth search* can be effectively used. Such a search can best proceed down the tree on a *semi-random* basis, biasing its selection of moves in a manner related to certain numerical criteria, derived from the partial layout equations; these criteria are statistically related to the eventual success of the paths.

Such a bias improves the chances of the search strategy reaching some valid terminal node, corresponding to a template which can be used to generate a legal layout. The additional incorporation of *esthetic rules* (Chapter 7) can assist in steering the search toward good-looking templates, which produce good-looking layouts.

In general, the presence of restrictions on acceptable layouts will make layouts harder to find. This is the case with setting an upper limit on allowable leading factors for stories (Chapter 8). Such a restriction can be represented by additional layout equations, but its effective incorporation necessitates a significant change in the basic nature of the layout equations, increasing the probability of their successful solution by weakening certain other restrictions in compensation.

The problem being solved is above all a practical problem. As such, the ultimate test of any approach is how well it works on practical cases. An experimental implementation of the layout algorithms has been programmed;

examples of its operation on news layout problems taken from actual newspapers are shown (Chapter 9).

Certain problems encountered in the design of implementations of the algorithms are discussed, along with suggested solutions (Chapter 10). Finally, a number of cpen research problems are listed (Chapter 11), and conclusions are presented (Chapter 12).

Appendix I presents a series of experiments performed to determine what rules, if any, are used by humans in performing news layout. Appendix II presents a number of theoretical considerations relating to the problem of news layout. Appendices III and IV compare past work on the computerization of news layout to the approach used in this thesis.

Chapter 2 - Templates

2.1 Definition

In his doctoral thesis [Kan 76], Kan introduced the concept of a *template*. As defined by Kan, a template contains purely geometrical information about a layout but specifies no numerical dimensions. A single template, therefore, can correspond to an (essentially) infinite number of layouts, each of them with different numerical dimensions, but all similar in the abstract shapes and locations of the items. Thus, a single template represents an abstraction of a large class of layouts. For example, all the layouts shown in Figure 2-1 can be represented by the same abstract template, shown in Figure 2-2. The template contains a number of *template items*, one for each news item in the eventual layout.

The problem of finding a suitable layout for a given page can be subdivided into the problem of first selecting a suitable template, and then the problem of finding a layout corresponding to that template. This chapter deals with the latter problem; template selection is covered in later chapters.

2.2 Template Specification

Before the problem of transforming a template into a layout can be attacked, it is necessary to specify exactly what information is held in a template. By definition, a template holds the geometrical information that will be used to produce the layout. But, how can this information be expressed?

In this thesis, templates are represented in a *matrix form*. Consider the template given pictorially in Figure 2-3. The matrix form for this template, for an arbitrary labelling of the template items, is:

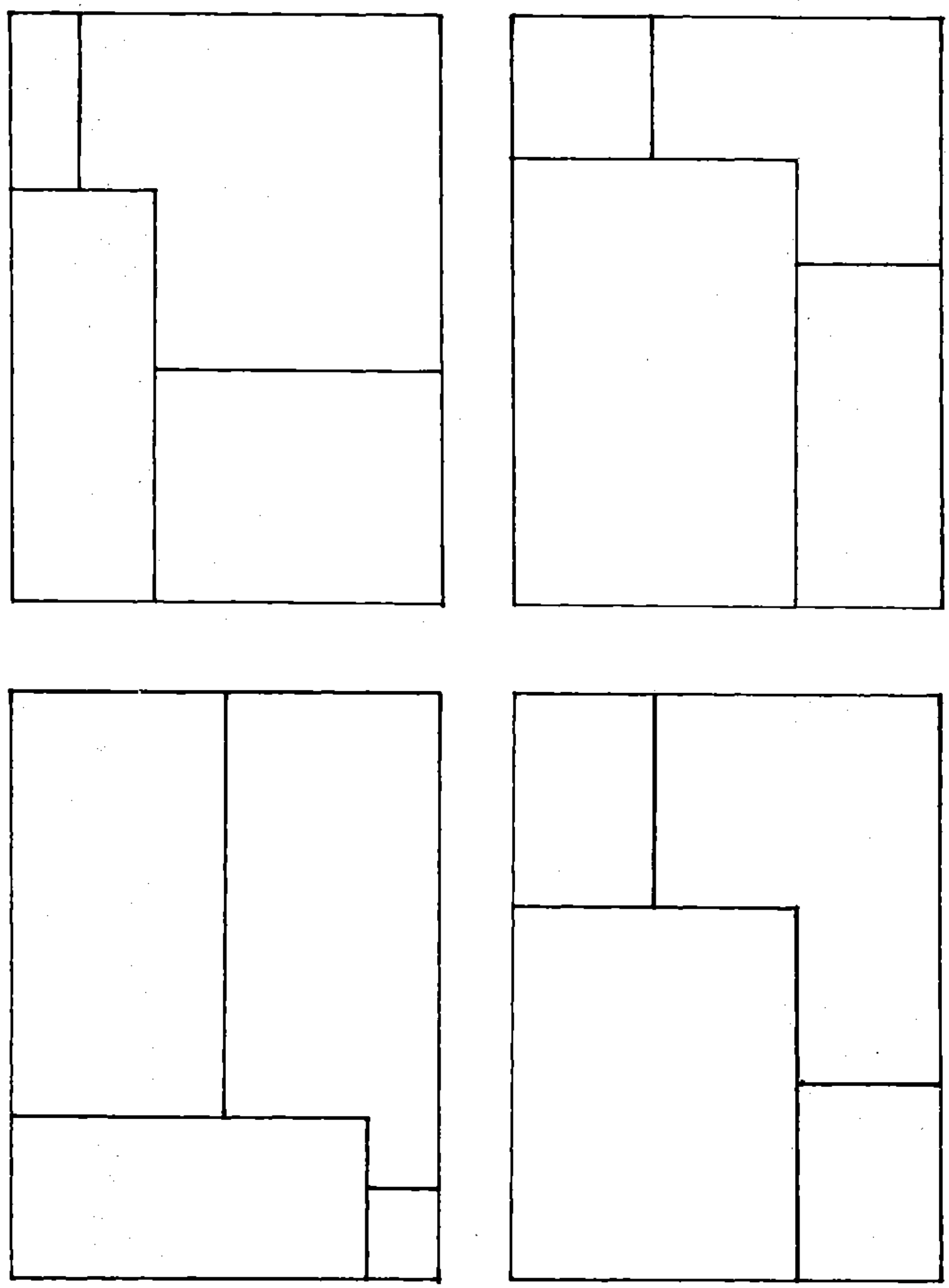


Figure 2-1. Layouts with the Same Abstract Template.

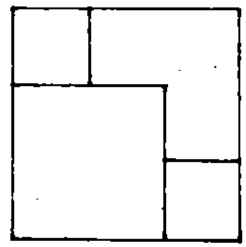


Figure 2-2. Abstract Template for Examples.

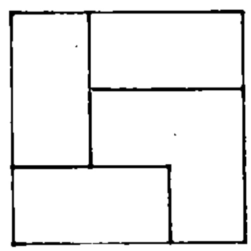


Figure 2-3. Example Template.

A B B

A C C

D D C

We see that the matrix form of the template contains the same information as the pictorial form. Indeed, the pictorial form may be immediately derived from the matrix form, as shown in Figure 2-4.

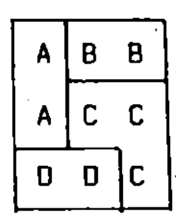
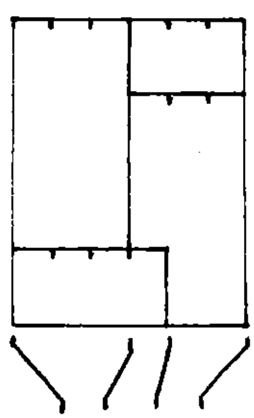


Figure 2-4. Derivation of Pictorial Form from Matrix Form.

From here on, the matrix and pictorial forms of templates are used interchangably. The pictorial form is easier to grasp visually, while the matrix form makes it easier to understand the operation of the layout algorithms.

A template contains one or more template items; in the final layout, each template item will correspond to one of the news items to be laid out. We define a *marked template* as one in which the correspondence between the template items and the news items they represent is explicitly labelled in the template, and an *unmarked template* as one in which no such correspondence is given.

A template in the matrix form also contains some number of columns. Each of these template columns corresponds to one or more news columns in the final layout, as shown in Figure 2-5. A *widened template* is one which contains exactly the same number of columns as does the layout (i.e., as does the newshole), so that each template column corresponds to exactly one news column. An *unwidened template* is one without this restriction.



A B B

A C C

D D C

Figure 2-5. Interpretation of Abstract Template Columns.

Consider the layout in a non-rectangular newshole shown in Figure 2-6. A *fitted template* is one which has the same overall shape as the newshole, as for the widened fitted template:

```
A A A B B B
A A A C C
D D D D C
```

whereas an unfitted template is one which does not explicitly state the interaction between the newshole shape and the news items, such as:

```
A A A B B B
A A A C C C
D D D D C C
```

Of course, for rectangular newsholes, there is no difference between fitted and unfitted templates.

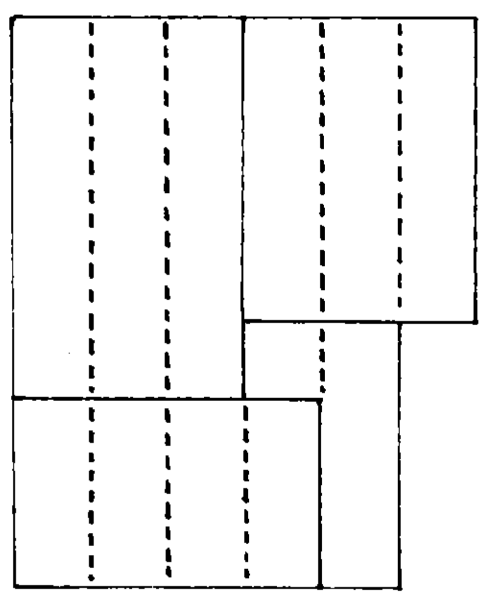


Figure 2-6. Layout in a Non-Rectangular Newshole.

A fitted, widened, marked template will be called a *concrete template,* because the correspondences of template items to news items, of template columns to layout columns, and of the shape of the template to the shape of the newshole are explicitly represented as part of the template. An unfitted unwidened unmarked template will be called an *abstract template.* In this thesis, we will be concerned primarily with the use of concrete templates.[1] Unless otherwise qualified, the term "template" in this report will refer to a concrete template.

2.3 Layout Equations

Given a concrete template, how can a layout be generated? Since the only information held in a layout but not in the template is that of the numerical dimensions of the layout, the problem is to find appropriate dimensions to add to the template so as to produce the layout. A layout has two sets of dimensions: *width* and *height.* Since a concrete template is by definition widened, the width dimensions are represented explicitly in the template (i.e., each template column corresponds to precisely one layout column), and are thus known, leaving only the variable height dimensions to be determined.

Each row of a template correspond to some unknown height dimension, as shown in Figure 2-7. We can symbolically represent the height dimensions by a set of *height variables,* one for each row:

```
A A A B B B     h1
A A A C C C     h2
D D D D C C     h3
```

1. Abstract templates will be used in Appendix IV, however, in the discussion of the "template library" approach presented by Kan in his thesis.

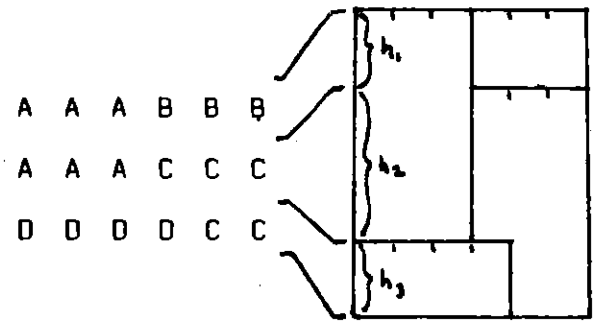


Figure 2-7. Interpretation of Template Rows.

The problem is to find appropriate numerical values for the height variables. These values can then be used to produce the layout. What, then, defines those values of the height variables which are appropriate? The approach taken in this thesis is to generate a set of simultaneous equations from the template. These equations will collectively state the necessary conditions on appropriate values of the height variables.

The first thing we note is that the sum of the height variables must equal the height of the newshole.[2] We can represent this restraint symbolically as:

$$h_1 + h_2 + \dots + h_n = \text{height of newshole} \qquad (2\text{-}1)$$

We term this equation a *height equation*.

Also, the height variables must always be non-negative:

$$h_1, h_2, \dots, h_n \geq 0 \qquad (2\text{-}2)$$

Another set of restraints on the height variables can be derived from the sizes of the news items. Each news item has a certain known minimum *area*, and must be assigned at least that much *space* on the page. It might be

2. The problem of non-rectangular newsholes is discussed in Section 2.4.1.

assigned more, usually without any great harm, with the excess simply remaining some form of white space. Alternatively, if there is more text or other matter available for optional use, it can be run.

For each news item, then, we can generate an *area equation*, which states that, for each item:

$$\text{space assigned in layout} \geq \text{area (item)} \tag{2-3}$$

For example, in the template shown above, the area equations are:

$$3\,h_1 + 3\,h_2 \geq \text{area (A)} \tag{2-4}$$
$$3\,h_1 \geq \text{area (B)} \tag{2-5}$$
$$3\,h_2 + 2\,h_3 \geq \text{area (C)} \tag{2-6}$$
$$4\,h_3 \geq \text{area (D)} \tag{2-7}$$

These area equations, combined with the height equation:

$$h_1 + h_2 + h_3 = \text{height of page} \tag{2-8}$$

and the non-negative condition, form a set of *layout equations*. Any values of h_1, h_2 and h_3 must satisfy these equations in order to correspond to a valid layout.

For the given layout equations, consider the layout problem given in Table 2-I. Here there is 9000 column-points of space available, but the areas of the news items sum to only 8600 column-points. The solution of the layout

---

Table 2-I. Example Layout Problem.

height of page = 1500	[i.e., 1500 points]
area (A) = 3000	[i.e., 3000 column-points]
area (B) = 1500	
area (C) = 2500	
area (D) = 1600	

equations is represented by the shaded region in Figure 2-8. The variable h_3 has been eliminated in this graph by use of the height equation:

$$h_1 + h_2 + h_3 = 1500 \tag{2-9}$$

The four area equations for the four template items are marked A, B, C, and D.

Since the layout equations are all linear equalities or inequalities, a simplex linear programming algorithm is sufficient for finding a solution to the layout equations. There is no function which needs to be maximized within the basic feasible region defined by the layout equations, so a linear programming algorithm which returns an arbitrary point in the region is adequate. The only caveat is that, because of the nature of linear programming, the solution returned will be at some vertex of the basic feasible region. In our

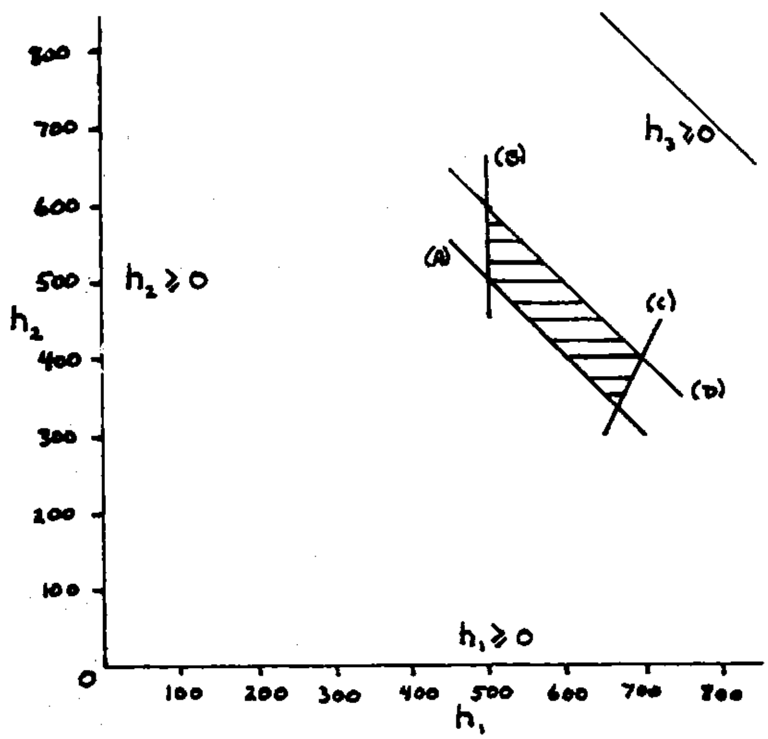


Figure 2-8. Solution of Layout Equations.

interpretation, then, some of the area equations will be satisfied as equalities (i.e., those news items will have zero leading), while others will be far from equalities (i.e., these news items will be leaded more than average).[3]

In this example, the layout equations a set of solutions. However, this may not always be the case. Most concrete templates, it develops, will have no corresponding layouts; their layout equations will therefore have no solutions. A simplex linear programming algorithm is capable of detecting such a case.

The layout equations, then, can be seen to represent the necessary conditions for a layout to be derived from a given template. A linear programming algorithm can determine whether the equations have a solution or not, and, if so, can generate the specific height dimensions for the layout.

2.4 More Layout Equations

The layout equations described in the previous section (the area equations and the single height equation), plus the non-negative condition, are necessary conditions on appropriate values of the height variables, but they are not necessarily sufficient. There are other necessary conditions which also need to be stated in the layout equations. Fortunately, these new equations are themselves in the form of linear inequalities, so that the linear programming approach can still be used to solve the resulting set of layout equations.

2.4.1 Non-Rectangular Newsholes

Consider the concrete (and therefore fitted) template:

---

3. This topic is discussed more fully in Chapter 8.

```
A  A  B  B  B  B

A  A  C  C

D  D  D  C
```

corresponding to the non-rectangular newshole shown in Figure 2-9. Here, the one height equation of the rectangular-newshole case must be replaced by a set of height equations:

$$h_1 = 800 \tag{2-10}$$
$$h_2 + h_3 = 700 \tag{2-11}$$

In general, for non-rectangular newsholes, there will be a height equation for each "level" in the newshole.

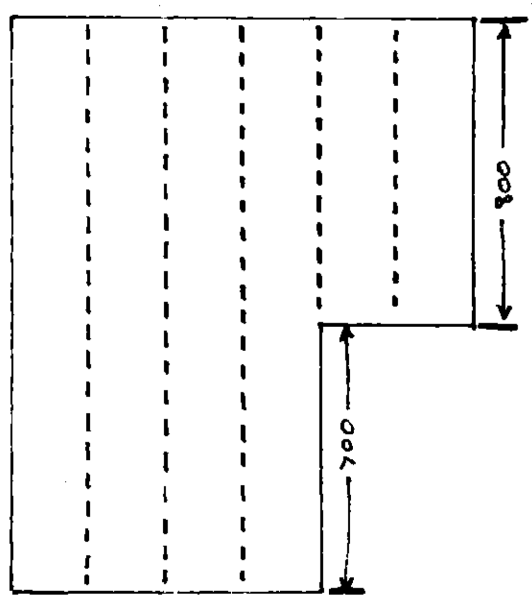


Figure 2-9. Example Non-Rectangular Newshole.

2.4.2 Headlines

Stories have headlines, and the dimensions of a story's headline must be taken into account in the layout equations. We shall assume the area needed for the headline is included in the total area needed for the story, as stated in the area equation. That is, the story's headline is not represented explicitly in the template, but is considered part of the containing template item.

There is, however, another restriction that must be stated as a layout equation. Consider the headline. We assume that we know its width, since the template is a widened template, and its height, from the numerical definition of the layout problem.[4] Not only must the space assigned to the story be great enough to accomodate the headline, it must also be great enough to allow it to be constrained to a rectangular form. That is, while a headline placed as in Figure 2-10 is acceptable, the one shown in Figure 2-11 is not, even though the numerical area assigned to each is the same, since the space assigned to headlines must be rectangular.

This condition can be stated as a layout equation. Consider the template:

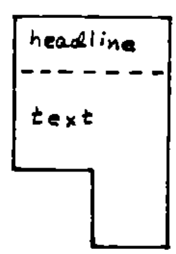


Figure 2-10. Example of Acceptable Headline Placement.

4. If there is more than one possible headline height for the given headline width, then the minimum of these should be used, since we are stating only necessary conditions on the space allocated.

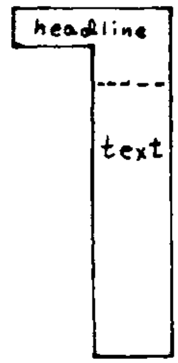


Figure 2-11. Example of Unacceptable Headline Placement.

A A A B B B

A A A C C C

D D D D C C

For story C, the headline equation is of the form:

$$h_2 \geq \text{height of headline of C} \tag{2-12}$$

This headline equation is in addition to the area equation for C:

$$3\,h_2 + 2\,h_3 \geq \text{headline area of C} + \text{text area of C} \tag{2-13}$$

This headline equation guarantees only that there will be enough space to run a rectangular headline. It does not rule out the possibility of a resultant layout such as the one shown in Figure 2-12 with no text running under part of the headline. This is considered an acceptable layout at some newspapers. For newspapers where such layouts are not used, the headline equation can be changed to:

$$h_2 \geq \text{height of headline of C} + \text{increment} \tag{2-14}$$

where the increment is the minimum height of text which absolutely must run

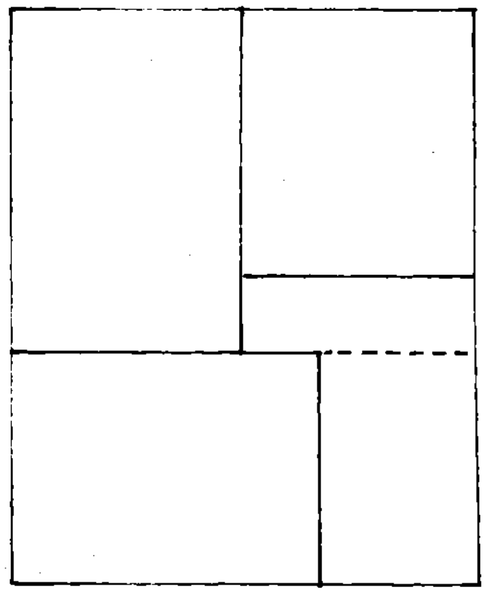


Figure 2-12. Questionable Layout.

underneath the headline. This can be either an absolute amount (e.g., two lines of text) or some amount relative to the headline size (e.g., 50% of the headline height).

The left-hand side of the headline equation will not always be a single height variable. In the template:

```
A A A B B B
C C C B B B
C C C D D D
C C E E D D
```

the headline equation for C is:

$$h_2 + h_3 \geq \text{height of headline of C + increment} \tag{2-15}$$

for some appropriate value of the increment. In general, the left-hand side of a headline equation is a sum of height variables, starting from the top of the template item, and terminating at the point where the template item ceases to be rectangular.

A corollary of this fact is that rectangular stories do not need their headline equations explicitly included in the set of layout equations. It is obvious that there can be no problem in providing a rectangular space for a headline within a rectangular story. Thus, in the above template, stories C and D need headline equations, while A, B and E do not, since their headline equations would simply be redundant: if we already require that:

$$3\ h_1 \geq \text{headline area of A + text area of A} \tag{2-16}$$
$$[\text{headline area of A = three times headline height of A}] \tag{2-17}$$

then the headline equation:

$$h_1 \geq \text{headline height of A + increment} \tag{2-18}$$

is unnecessary.

Strictly speaking, this headline equation might not be automatically satisfied if the text area becomes relatively small compared to the increment. This situation, however, can be avoided by the choice of the increment as an appropriate function of the text area.

2.4.3 Independent Pictures

There are two types of pictures: *independent pictures,* which are not associated with any story on the page, and *dependent pictures,* which are. The layout equation for an independent picture is simple. First, we note that the template item corresponding to an independent picture must be rectangular, and that the width of the template item must correspond to the width of the picture. With this given, the *picture equation* is simply:

$$\text{height assigned to picture} \geq \text{height (picture)} \qquad (2\text{-}19)$$

where height(picture) represents the minimum possible height the picture can be cropped to, since the picture equation states only a necessary condition. In the template:

```
A A A B B B
A A A C C C
D D D D C C
```

if A is a picture, its layout equation is simply the picture equation.

$$h_1 + h_2 \geq \text{height (A)} \qquad (2\text{-}20)$$

No separate area equation is needed for an independent picture, since it would state exactly the same condition, merely multiplied by the width of the picture.

Pictures typically have a *cutline* (or *caption*) beneath them, of the same width as the picture. The height of this cutline can be considered as part of the height of the picture for the purposes of the layout equations. The same is true for a headline appearing above the picture.

2.4.4 Dependent Pictures

Pictures which are associated with a story and which are to appear as part of the story are more difficult to handle. As with headlines, we assume that the location of a dependent picture or pictures within the template item corresponding to a story is not stated directly within the matrix form of a template, but is defined by a *story-picture format*, present as an adjunct to the template. The exact type of equation to be generated depends on the story-picture format chosen. The following formats are typical.[5]

2.4.4.1 Format 1

Format 1 is shown in Figure 2-13. The format is simply that of a picture to the left of the remainder of the story. Although the text shown in the figure is rectangular, this does not need to be the case. Of course, however, the picture must be rectangular. For the purposes of the layout equations, the two parts of the format, the picture and the headline-text combination, can be considered separately, with a picture equation for the picture, and an area equation and a headline equation for the remainder of the story. The area equation should include the area of the text plus the area of the headline, but not the area of the picture. Consider the template:

```
A A A A B B
C C C A B B
```

where story A, with a two-column picture, is to be laid out in Format 1. The picture will be placed in the first two columns of item A; this is acceptable since these columns define a rectangular space. For story A, the layout equations are:

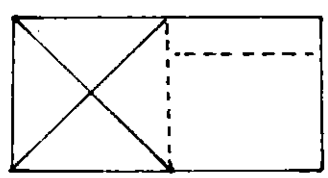


Figure 2-13. Format 1.

5. By extension, we can consider the headline-text combination present in a story without independent pictures to represent the single possible "format" for such a case; the same is also possible for independent pictures.

$$h_1 \geq \text{picture height} \tag{2-21}$$
$$2\,h_1 + h_2 \geq \text{text area} + \text{headline area} \tag{2-22}$$
$$h_1 \geq \text{headline height} + \text{increment} \tag{2-23}$$

2.4.4.2 Format 2

In Format 2, shown in Figure 2-14, the picture appears directly above the headline, and is of the same width. In this case, the conceptual approach is to think of the combination of picture and headline as if it were a single very tall headline, and generate the layout equations appropriately. The text, of course, need not be rectangular. Consider the template:

```
A A A B B B

A A A B B C
```

where story B, with a three-column picture, is to be laid out in Format 2. The layout equations are:

$$h_1 \geq \text{picture height} + \text{headline height} + \text{increment} \tag{2-24}$$
$$3\,h_1 + 2\,h_2 \geq \text{picture area} + \text{headline area} + \text{text area} \tag{2-25}$$

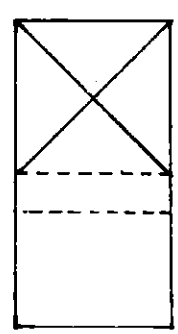


Figure 2-14. Format 2.

2.4.4.3 Format 3

In Format 3, shown in Figure 2-15, the picture appears to the right of the headline, the headline has text running beneath it, and there may be text running beneath the picture. In the template:

```
A B C C C C

D D D C C C

E E E E E E
```

where story C, which has a picture two columns wide, is to be laid out in Format 3, the layout equations are:

$$h_1 \geq \text{headline height} + \text{increment} \qquad (2\text{-}26)$$
$$h_1 + h_2 \geq \text{picture height} \qquad (2\text{-}27)$$
$$4\,h_1 + 3\,h_2 \geq \text{text area} + \text{headline area} + \text{picture area} \qquad (2\text{-}28)$$

Here, we are requiring that the first two columns, which hold the headline, have sufficient minimum height (here, h_1) for it, while requiring that the last two columns, which hold the picture, have sufficient minimum height (here, h_1+h_2) to do so. The use of the increment in the headline equation requires that there be enough space to run some text beneath the headline; the lack of such an increment in the picture equation means that, although it is possible for there to be room for text underneath the picture, this is not required.

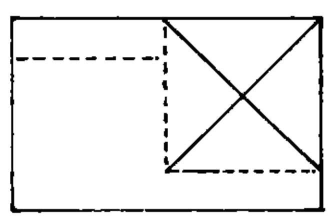


Figure 2-15. Format 3.

2.4.4.4 Format 4

Format 4, shown in Figure 2-16, is very similar to Format 3. The only difference is that the headline runs above the picture, slightly modifying the picture equation. With the template:

```
A B B C C C

A B D D D D

E E E D D D
```

laying out story D, with a two-column picture, in Format 4 gives the layout equations:

$$h_2 \geq \text{headline height} + \text{increment} \tag{2-29}$$
$$h_2 + h_3 \geq \text{headline height} + \text{picture height} \tag{2-30}$$
$$4\,h_2 + 3\,h_3 \geq \text{text area} + \text{headline area} + \text{picture area} \tag{2-31}$$

The only difference from Format 3 is in the right-hand side of the picture equation.

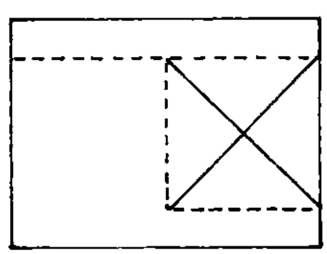


Figure 2-16. Format 4.

2.4.4.5 Other Formats

Other story-picture formats are possible, not only for stories with only one dependent picture, but also for those with more. The above examples, though, should be sufficient for showing how to develop the form of the layout equations needed for arbitrary formats, and for stories with an arbitrary number of pictures.

2.5 Example

Consider the concrete template:

```
A A B B B
C C C B B
C C C D D
```

where the news items are defined in Table 2-II. Item A and item C are stories with no dependent pictures. Item B is a story with one dependent picture, which will be laid out in Format 2. Item D is an independent picture. The newshole is five columns wide and 1160 points tall.

The resultant layout equations are shown in Table 2-III. The first three equations state the non-negative condition on the height variables. The next equation is the height equation, requiring that the sum of the height variables be equal to the height of the newshole. The area equation for item A requires

---

Table 2-II. News Item Sizes for Example.

item	text size	headline size	picture size
A	589	2x80	-
B	1121	3x40	3x270
C	2252	3x40	-
D	-	-	2x261

Table 2-III. Layout Equations for Example.

$$h_1 \geq 0$$
$$h_2 \geq 0$$
$$h_3 \geq 0$$

$$h_1 + h_2 + h_3 = 1500 \qquad \text{[height eqn]}$$

$$2\,h_1 \geq 589 + 2\times80 \qquad \text{[area eqn for A]}$$

$$3\,h_1 + 2\,h_2 \geq 1121 + 3\times40 + 3\times270 \qquad \text{[area eqn for B]}$$
$$h_1 \geq 270 + 40 \qquad \text{[picture-headline eqn for B]}$$

$$3\,h_2 + 3\,h_3 \geq 2252 + 3\times40 \qquad \text{[area eqn for C]}$$

$$h_3 \geq 261 \qquad \text{[picture eqn for D]}$$

One solution to these equations is:

h_1 = 389.33 points
h_2 = 509.67 points
h_3 = 261.00 points

---

that the space allocated to A, namely $2\,h_1$, be large enough to hold the text and headline area of A. Story A needs no headline equation, because it is rectangular in shape. The area equation for item B requires that the space allocated for it, $3\,h_1 + 2\,h_2$, be large enough to hold the text, headline, and picture areas of B. Also, a picture-headline equation for B is necessary, requiring that the rectangular space at the top of the template item have sufficient height, here h_1, for the picture and the headline to be allocated rectangular spaces; for the sake of simplicity, the increment was left out of the right-hand side of this equation. The area equation for item C requires that the space allocated to C, namely $3\,h_2 + 3\,h_3$, be large enough to hold the text and headline area of C. The picture equation for item D requires that its height, h_3, be large enough to hold the picture. The layout based on the given solution to the equations is shown in Figure 2-17.

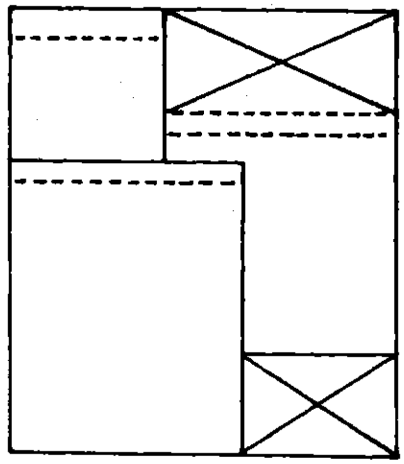


Figure 2-17. Final Layout for Example.

2.6 Summary

This chapter has introduced the concept of a template. A template contains the same non-numerical information as does a layout, but without specifying the numerical dimensions. Given a template corresponding to a given layout problem, it is possible to generate a set of layout equations whose solution, if any, is the set of numerical dimensions needed to transform the template into a layout.

The following chapters consider the problem of how these concrete templates are to be supplied.

Chapter 3 - Template Grammar

3.1 Definitions

Chapter 2 presents a template-layout procedure that produces a layout from a known concrete template, if a layout with that template is possible. In the general layout problem, however, no such template will be available a priori. It will be necessary somehow to produce one or more templates based on the given layout problem, and then to test each of these templates for correspondence to a layout. Testing can stop when a layout has been achieved, or, alternatively, a number of layouts can be generated, with the layout person deciding when to stop this process, and then choosing from among the layouts produced up to that point.

When generating a number of likely templates, not all possible templates need be considered. Some, such as the one shown in Figure 3-1, would be considered illegal at any newspaper, while others, like the one shown in Figure 3-2, are definitely legal. When generating templates, we should restrict ourselves to the legal ones only. What, then, defines a legal template?

In the theory of linguistics, there is the concept of a *grammar,* which

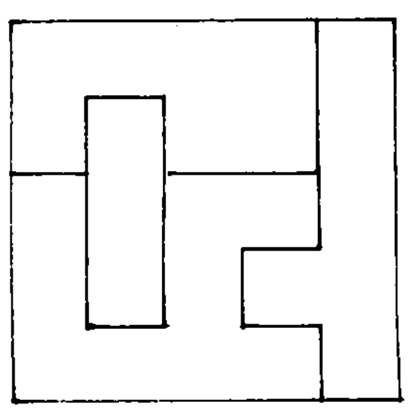


Figure 3-1. Illegal Template.

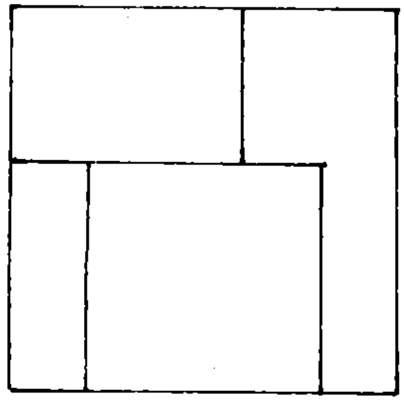


Figure 3-2. Legal Template.

---

defines the syntax of a language. The grammar consists of rules which specify the set of legal constructs (e.g., sentences) in the language; a construct is legal if, and only if, it belongs to the set of legal constructs thereby defined.

By analogy, then, we can develop a *template grammar* which will specify which templates are legal and which are not. Two types of grammar will be considered. The first, a *descriptive grammar*, gives rules for determining if an arbitrary template belongs to the set of legal templates. The second, a *generative grammar* based on the first, gives rules whereby the set of legal templates may be generated.

3.2 Descriptive Grammar

Unfortunately, not all newspaper personnel would agree on what constitutes a legal layout. To the extent that this disagreement involves geometrical issues, this implies that not all newspapers would agree on what constitutes a legal template. Even so, a careful study of layouts in a variety of newspapers shows that there are some common rules for legal templates which can be stated. Individual newspapers may wish to make changes to these rules.[1] Still, these rules do a good job of stating what features concrete templates at different newspapers have in common. There are seven rules, as follows.

Rule 1. The shapes of template items must never become larger as one scans from top to bottom.

For example, template item shapes such as the ones shown in Figure 3-3 are illegal, while those shown in Figure 3-4 are legal. In other words, a template item must have a unique flat top.

Rule 2. Each template item must be contiguous.

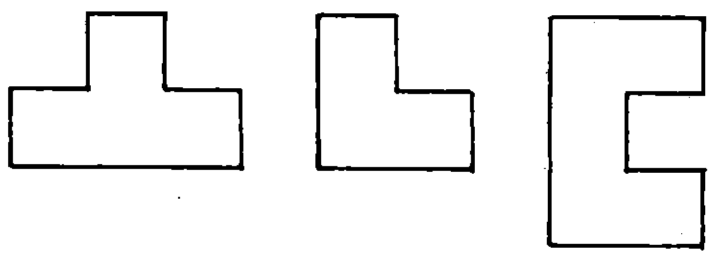


Figure 3-3. Illegal Template Item Shapes.

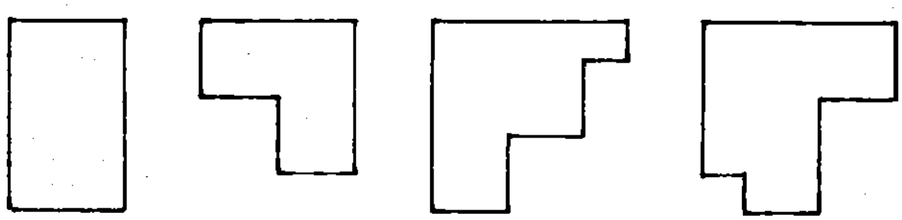


Figure 3-4. Legal Template Item Shapes.

1. Better yet, those newspapers could change their layout rules to conform to those given here. The latter is especially advisable considering that the simplicity of the generative grammar, as given in the next section, depends on the exact details of this descriptive grammar.

Repl. pg 51

Obviously, items should not jump about on the same page.

Rule 3. There must be one and only one template item for each news item.

This rule ensures that the template is a properly marked template.

Rule 4. The width of the top of a template item must be a possible one for the associated news item.

For example, a story whose only possible headlines are four and five columns wide should not have an associated template item with a one-column top. An independent picture always has exactly one possible width for its template item. For stories with dependent pictures, the choice of the story-picture format determines the possible top widths for the template item.

Rule 5. Each story with dependent pictures must have associated with it an acceptable story-picture format.

The story-picture formats given in Chapter 2 are all acceptable. Others are also acceptable, still others are not. There seems to be no simple general rule for determining which are which without utilizing human judgement. Thus, if template generation is to performed automatically, the layout program should have access to a list of pre-approved story formats to use, either built in or provided by the layout person.

Rule 6. Template items corresponding to pictures must be rectangular.

This includes that, for template items in Format 1, or similar formats, the part of the template item which will hold the picture must be rectangular.

Rule 7. The template items must be placed together in such a way as to produce a template which has the same width and the same overall shape

as the newshole,[2] with no holes left.

These seven rules collectively specify what will be considered in this thesis the set of *legal templates*. They require that the template be a concrete template, that the template items correspond correctly to their news items, and that template items have only certain legal shapes.

The purpose of these rules is only to outlaw those templates which should definitely be avoided. They are not intended to make any further distinctions between the various levels of quality found among the legal templates. The issue of template quality is discussed in Chapter 7.

3.3 Generative Grammar

The above descriptive grammar enables us to test a given template for legality. Moreover, this grammar can be transformed into a generative grammar which can be used to generate legal templates.

This generative grammar will give rules for generating templates incrementally, one template item at a time. At an intermediate stage in this process, we will have constructed a *partial template*. At the very beginning, the partial template is empty. At the end, the partial template becomes the final template. Each step of the template-generation procedure involves the addition of a single template item, corresponding to some particular news item, to the partial template, thereby obtaining a new partial template. At any step, there might be more than one possible item to add, and more than one possible way to add it. These rules do not tell which possibility to choose; they only state which possibilities are legal.

This generative grammar is based on a simple observation about the set of

2. I.e., they must produce a fitted widened template.

legal templates as defined in the previous section. Consider such a template:

```
A A A B B B

A A A C C C

D D D D C C
```

The set of allowable shapes for template items is such that if we know where in the template the tops of the items are to go:

```
A A A B B B

- - - C C C

D D D D - -
```

it is possible to construct the remainder of the template mechanically. Therefore, when generating templates, it is necessary only to choose the positions of the tops of the stories: the rest of the template can be automatically filled in later. For the remainder of this report, templates may be represented in this latter manner, so as to make it easier for readers to locate the tops of the template items and thereby see how the templates were generated.

To lay out an item in a partial template, some legal top width is chosen, then a top of that width is inserted somewhere in the template. The template above could be generated in the four steps shown in Figure 3-5. In fact, this template could be generated in a total of 24 ways (i.e., 4! ways), depending on the order in which the items are added.

In the generation of templates for a non-rectangular newshole, the conceptually simplest approach is to begin with an empty template marked with the location of the ads, and to modify the above procedure so that tops may not be laid out inside the marked regions. The template:

```
- - - B B B


- - - B B B
- - - C C C


A A A B B B
- - - C C C


A A A B B B
- - - C C C
D D D D - -
```

Figure 3-5. Generation of Example Template.

```
A A A B B B
    A B B B
    A C C
    D D C
```

could be generated as shown in Figure 3-6, where ads are marked with a "\$".

To generate a template, then, one starts with the empty template, and repeatedly adds individual items to it, until all the items have been used. Each time a template item is added, an appropriate top and an appropriate format must be chosen. If this is done, the resulting final template will be guaranteed

```
- - - - - -
$ $ - - - -
$ $ - - - $

- - - B B B
$ $ - - - -
$ $ - - - $

- - - B B B
$ $ - - - -
$ $ - C C $

A A A B B B
$ $ - - - -
$ $ - C C $

A A A B B B
$ $ - - - -
$ $ - C C $
$ $ D D - $
```

Figure 3-6. Generation of Example Non-Rectangular Template

to follow most of the rules for legal templates listed in the descriptive grammar. Some rules, however, will have to be checked independently. For example, the descriptive grammar requires that there be no holes left in the template. The only way that this could happen, given our interpretation of item shapes in terms of the tops of items, is for the very top line of the template not to be filled in. This condition must be checked. It is also necessary to check that the parts of the template corresponding to pictures are rectangular. If these criteria are satisfied, then a template built in the manner described is a legal one.

3.4 Summary

This chapter has introduced the class of legal templates, and has outlined a number of rules for recognizing legal templates and for generating them. These rules for template generation state only what templates *can* be generated, not which templates *should* be generated. This dichotomy can be compared to the relationship in linguistics between syntax and semantics. A grammar for English may tell you how to form correct sentences, but it cannot tell you what to say.

The issue of template syntax is developed more fully in the next two chapters, and then template semantics (the problem of determing which templates to generate) is considered.

Chapter 4 - Layout Trees

The preceding chapter outlines a scheme whereby an arbitrary template may be generated by a series of incremental operations. This chapter further develops this idea, using an analogy with game-playing programs.

4.1 Definition

Many computer programs have been written to play games such as checkers and chess against human opponents [Newborn 75]. Although these programs differ in detail, they all use the concept of a *game tree* in generating moves.

The nodes of a game tree are all of the board positions achievable in the game, and the edges are the moves which relate one position to another. Figure 4-1 shows part of the complete game tree for chess. The root node of the tree corresponds to the board at the beginning of the game. The nodes at the first *ply* represent the 20 possible board positions after White makes his first move; the edges leading to them represent the 20 possible moves. Similarly, the nodes at the second ply represent the 400 possible board positions after Black has replied with his first move, and so forth.

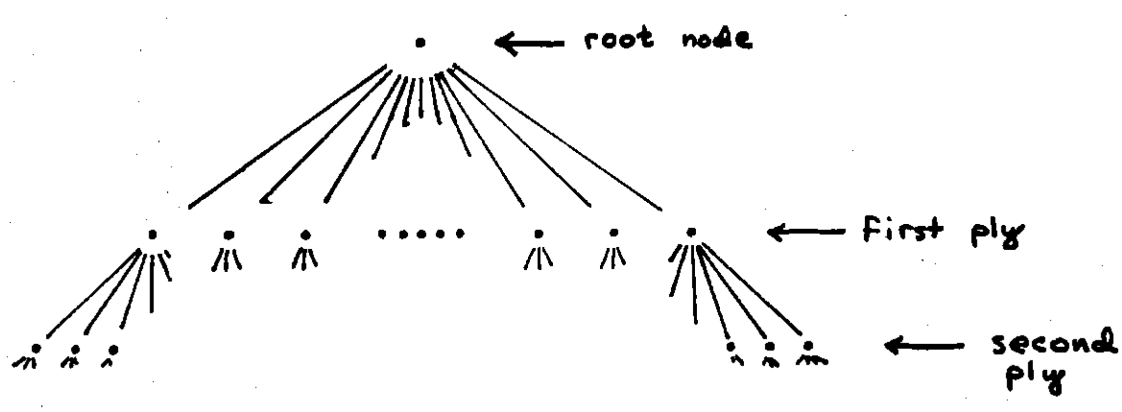


Figure 4-1. Part of the Complete Game Tree for Chess.

The complete game tree for chess is finite, since the rules for chess set an upper limit on the length of a game. Even so, it is estimated that there are approximately 10^{120} possible paths through the tree [Shannon 50]. Obviously, the complete game tree for chess will never be enumerated;[1] it remains forever a conceptual entity, which, still, has intellectual value. Playing a game of chess can be compared to following a path downward through this tree; the players alternate in choosing edges (moves). Some of the nodes (positions) in the tree represent a win for White; some represent a win for Black. White's aim is to force a path to one of his winning nodes, and vice-versa for Black. Unfortunately, a chess-playing program will not have an overview of the entire tree; it will be able, though, to "mentally" explore some neighborhood of the tree for each of its moves, starting from the then-current position, and use the information thereby gained to determine the seemingly-best move.

By analogy, one can consider a given news layout problem to be a game, although a solitaire one. We call its game tree a *layout tree*. The playing board is the template being constructed. At the beginning, the template is empty. Each move consists of adding an item to the template, leading to a new node in the layout tree corresponding to the new position. The final positions are those where all of the items have been added; these occur in the tree at the n^{th} ply, where n is the number of items to be laid out on the page.

Some of these final positions, as we have seen in the last chapter, will be syntactically illegal. These count as losses. Others will be syntactically legal, but semantically invalid. These are the templates without associated layouts, and also count as losses. The wins are those legal templates which do have associated layouts.

---

1. Assuming that certain current assumptions about the nature of the physical universe are correct.

The problem is to move through the tree to a winning position, thereby obtaining a layout. As in chess, the complete tree for a given layout problem is far too large to enumerate. It is necessary to search through the tree based on imperfect knowledge of where the search will lead. There are, however, some ways of increasing that knowledge, as described below.

4.2 Partial Layout Equations

As we have seen, the final nodes in the layout tree correspond to completed templates. Given one of these templates, we can generate a set of layout equations which tell us whether or not there is a layout based on that template, and, if there is, give its dimensions.

Each of the intermediate nodes in the layout tree corresponds to a partial template. Given one of these, it is possible to generate a set of *partial layout equations,* based on the partial information available, which tell whether there is any possibility of this partial template giving rise to a completed template which has a layout. Consider, for example, the partial template:

```
A - - - - -
- - - - F F
- C C C C -
```

A set of partial layout equations for this partial template is shown in Table 4-I. If there exist any h_1, h_2, and h_3 which satisfy these equations, then there is the

---

Table 4-I. Partial Layout Equations.

$h_1 + h_2 + h_3 =$ height of page	[partial height eqn]
$h_1 + h_2 + h_3 \geq$ area (A)	[partial area eqn for A]
$4\, h_3 \geq$ area (C)	[partial area eqn for C]
$2\, h_2 + h_3 \geq$ area (F)	[partial area eqn for F]
$h_2 \geq$ headline height (F)	[partial headline eqn for F]

possibility that a completed template with an associated layout can be built from this partial template. If not, then there is no such chance, since the partial layout equations necessarily state a strictly weaker condition on the existence of height variables than will any of the possible eventual sets of final layout equations.[2] The actual values of the height variables which satisfy the partial layout equations are irrelevant; what matters is whether any such height variables exist. The problem being solved here is not to determine partial numerical dimensions for the layout, but to determine if any such layout can exist. No actual numerical dimensions can necessarily be inferred from a partial template, since it is not yet known where the remaining template items will be inserted. Different subsequent choices will force different layout positions of the items already placed.

In this example, if news item A is too large to fit into one column, then the partial layout equations will fail. Similarly, if news items C and F are too large to fit together in the manner specified by the template, the equations will also fail. If the equations do succeed, there is still the possibility that the partial template might not lead to any layouts, because of the particular mix of news items remaining to be inserted into the template. However, this will be because of conditions not expressible in the partial layout equations.

A partial template whose set of partial layout equations succeeds is termed *valid*, as is a legal completed template whose set of final layout equations succeeds;[3] all other templates are *invalid*. By analogy, the various nodes of the layout tree for a particular layout problem can be considered valid or invalid, as can the particular moves leading to them. Since the partial layout

---

2. Adding an item to a template does not, in general, simply add new layout equations to the set of partial layout equations, but various existing equations may also change.

3. All partial templates are considered legal, since they are constructed by legal moves. In general, only a final template can be considered definitely illegal for such reasons as a non-rectangular shape for a picture, since such faults in a partial template might be corrected by the time it becomes final.

equations state necessary conditions on the existence of an eventual layout, no invalid node can have valid descendants (although the reverse can obviously be true). Therefore, any subtree of the layout tree originating from an invalid node will be uniformly invalid; in particular, all of the templates corresponding to final nodes in that subtree will be invalid and will not lead to layouts. Such a subtree can be totally ignored by any search procedure. In terms of our analogy, this means that, when adding an item to a partial template, the layout equations of the new template should be checked for satisfiability. If they fail, the move is invalid and should not be considered.

Similarly, any illegal completed templates can be removed from the tree, and the final moves leading to them considered invalid. With the use of lookahead, it is sometimes possible to determine whether or not intermediate positions can possibly lead to legal final templates; those that cannot should be considered invalid.

4.3 Shape of the Layout Tree

Although there will be a different layout tree for each layout problem, some generalities can be made about the shapes of layout trees.

There are more possible moves early in layout than there are later on. At the beginning, there are simply more news items remaining to be laid out. Toward the end, there are fewer remaining, and the requirements for legal completed templated must also be dealt with, potentially reducing the number of possible placements for each remaining item.

There is a higher probability that a legal position early in the tree will be valid than for positions later in the tree. The root position is always valid, of course. Positions at the first ply have small sets of partial layout equations, which state relatively weak conditions, and therefore stand a good chance of being valid. For positions at subsequent plies, the sets of partial layout equations change, growing in size as they do, and become more and more difficult to fulfill, causing a decrease in the probability of a valid position.

4.4 Basic Search Strategy

To find a layout, it is necessary only to search through the layout tree until a valid final node is located. This, unfortunately, is more easily said than done, because of the great size of the tree. The basic approach to search strategy taken in this thesis[4] is to start at the root node of the layout tree and move downward, making only legal moves, and checking the validity of each move by generating and attempting to solve each set of partial layout equations. This procedure may lead directly to a layout. Alternatively, there may come an intermediate point where there are either no legal moves possible, or where there are no valid moves among the legal ones. Either case can be termed a *dead end*. Upon encountering a dead end, the search procedure will backtrack, trying other possible paths. Eventually, if there are layouts, one will be found. This procedure should be monitored by the layout person to catch the rare cases in which there are no layouts possible with the given items on the given page, as well as the cases where the number of layouts is so small that finding one is like finding a needle in a haystack, becoming excessively difficult.

The amount of time that this search procedure takes depends on the number of positions in the tree it must consider. One obvious conclusion is that the effectiveness of the procedure depends on the effectiveness of the partial layout equations generated at each step. The more *blind alleys* (paths leading only to dead ends) that can be detected at their offset, the less time will be required to generate a layout.

---

4. This approach is developed more fully in Chapter 6.

4.5 Summary

The concept of a layout tree provides a convenient conceptual framework for understanding the layout problem. Generating a template is equivalent to moving downward through the tree; producing a final layout can be performed by searching through the tree, generating partial templates along the way, until a valid final template is reached.

The following chapter considers certain modifications of the rules for legal moves which make the search procedure more effective.

Chapter 5 - Direction of Layout

5.1 Canonical Orderings

As was shown in Chapter 3, a single template can be built in a variety of different ways, corresponding to the variety of possible orders in which the template items can be inserted into the partial templates. With n items to be laid out, there will be n! possible orders of insertion. These n! orderings will correspond to n! distinct paths through the layout tree, terminating at n! final nodes, each of which will correspond to the same final template.

Since the order of insertion of the items into the partial templates does not matter, restricting any particular final template to be constructed in a single particular order will reduce the size of the layout tree, potentially making searching easier. In this case, it matters not how you play the layout game, but whether you win or lose. These changes will represent changes in the rules of the layout game, making it potentially easier to win. Of course, the winning positions will still correspond to the same set of valid legal final templates. Each valid legal final template will appear once and only once in the restricted layout tree, and there will be only one path leading to each such template.

We define a canonical ordering based on the matrix form of the template. An item whose top appears on a template row higher than the top of another item will appear earlier in the ordering than the other item. For items whose tops appear on the same template row, the ones appearing the further left will appear earlier in the ordering. For example, for the template:

```
A A A B B B
A A A C C C
D D D D C C
```

the canonical ordering will be A, B, C, D. This ordering, which represents a total ordering on the items, will be called the *top-down ordering*. This ordering goes from left to right, top to bottom.

Another possible canonical ordering is the *bottom-up* ordering, which we define to be precisely the reverse of the top-down ordering. This ordering proceeds from right to left, bottom to top.

Either of these two possible orderings will satisfy the goal given above of there being only one possible path through the layout tree to a given final template. In this way, these orderings represent additions to the template grammar, in that they restrict the set of possible moves. As will be seen, each of these orderings has its unique advantages. Layout using the bottom-up ordering will be considered first, since it is the easier to understand.

5.2 Bottom-Up Layout

5.2.1 Equation-Generation Method

5.2.1.1 Basic Method

Consider building the template:

```
A A D D D

C C C D D

C C C B B
```

bottom-up. This will proceed as shown in Figure 5-1. The template items are inserted in the order B, C, D, A. The unique advantage of bottom-up layout lies in the fact that it is possible to generate the layout equations of the items as soon as they are inserted into the partial templates. For example, the area equation for B is:

```
- - - B B


C C C - -

- - - B B


- - D D D

C C C - -

- - - B B


A A D D D

C C C - -

- - - B B
```

Figure 5-1. Bottom-Up Generation of Example Template.

$$2\,h_1 \geq \text{area}\,(B) \qquad \textbf{(5-1)}$$

where height variables are numbered from the bottom up, so that h_1 is the height variable corresponding to the lowest row of the template under construction.[1] This area equation can, of course, be easily generated from the final template, but it can also be generated in the same form at the very first point that item B is added to the partial template, because of the bottom-up direction of layout. The other area equations, which can also be generated in

1. This reversal of the usual numbering scheme will be used throughout the discussion of bottom-up layout.

turn, are:

$$3\ h_2 + 3\ h_1 \geq \text{area (C)} \qquad (5\text{-}2)$$
$$3\ h_3 + 2\ h_2 \geq \text{area (D)} \qquad (5\text{-}3)$$
$$2\ h_3 \geq \text{area (A)} \qquad (5\text{-}4)$$

As can be seen, each of these area equations can be generated as soon as the corresponding template item is inserted into the template. This is because the exact symbolic shape of each template item is known immediately, due to the interaction between the bottom-up direction of layout and the template grammar; the symbolic shape of a template item does not depend on the placements of the following items.

Headline equations and picture equations can be generated with equal simplicity. None of the layout equations will need to be changed as further template items are inserted, with the exception of the height equation, which will have to be changed so as to incorporate the new height variables.

5.2.1.2 Height Equation

At an intermediate point in template generation, the set of height variables will be incomplete. Therefore, the height equation at an intermediate point cannot refer directly to the sum of the height variables, but must state that the sum of the subset of the height variables that have been introduced up to that point, plus some dummy variable h_n, equals the height of the newshole. The dummy variable h_n represents the sum of the height variables yet to be introduced. In the above example, for instance, after the first template item is inserted, the height equation is:

$$h_1 + h_n = \text{height of newshole} \qquad (5\text{-}5)$$

In case of a non-rectangular newshole, only a little more processing is required for generating the multiple height equations. Consider the newshole shown in Figure 5-2. We divide the newshole into a number of *levels*, one for each of the height equations that will be generated. In this case, there are three levels, leading eventually to three height equations. Layout begins in the bottom level. When deciding where to place the top of an item, one possibility is to terminate the current level and move to a higher level, possibly skipping over some number of intermediate levels. At any point, the variable h_n refers to the sum of the height variables yet to be introduced in the current level.

Template construction in the above newshole might progress as in the following sequence. We start with:

$ $ E E $ $

The current height equation is now:

$$h_1 + h_n = 350 \qquad (5\text{-}6)$$

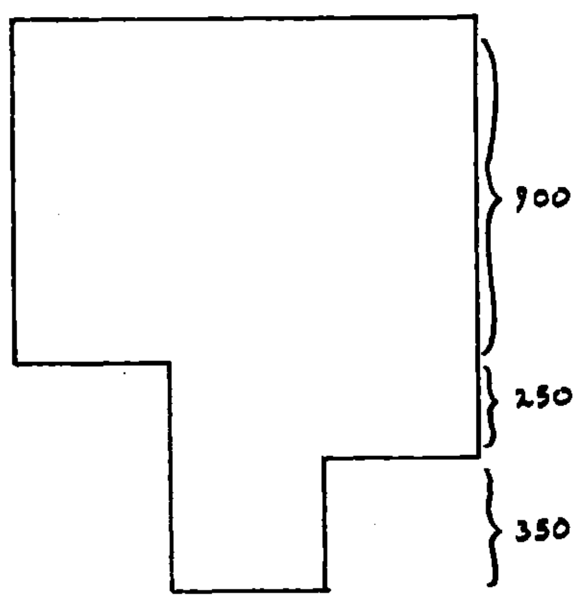


Figure 5-2. Non-Rectangular Newshole.

At this point, let us suppose that the next higher level is chosen. The height equation:

$$h_1 = 350 \tag{5-7}$$

is final, and a new one is started. The partial template becomes:

```
$ $ D D D D
$ $ E E $ $
```

and the new height equation is:

$$h_2 + h_n = 250 \tag{5-8}$$

Above, the next-level operation was performed in such a way as to require that the top of E, already placed in its level, be at the same height as the top of its level in the final layout. Another possibility is to perform the next-level operation as in:

```
- - C C C -
$ $ - - - -
$ $ D D D D
$ $ E E $ $
```

which allows there to be some distance between the top of D and the top of its level in the final layout. Here, the height equation:

$$h_2 + h_3 = 250 \tag{5-9}$$

is final; the height variable h_3, which may or may not be zero in the final layout, represents the distance between the top of D and the top of its level. A new height equation:

$$h_4 + h_n = 900 \qquad (5\text{-}10)$$

is started, and template construction continues.

In summary, before a template item is inserted into a partial template, one or more next-level operations may be performed. These may specify whether the item previously inserted is or is not to be constrained to be at the same height as the top of its level.

5.2.1.3 Items on the Same Row

Consider the two partial templates:

```
- - E E C C
```

and

```
- - E E - -
- - - - C C
```

The effective difference between these two partial templates is that, in the first, the top of item E is constrained to be at the same height as the top of C, whereas no such constraint exists in the second partial template. The first partial template can be viewed as a special case of the second, since the height variable h_2 in the second partial template can take on the particular value zero, making it equivalent to the first in such a case. If any final template built with the first of these partial templates can achieve a layout, so can an equivalent template built with the second. Moreover, it may be the case that layouts exist based on the second partial template that have no counterpart based on the first partial template, since the height variable h_2 in their final templates is effectively constrained to be greater than zero. Thus, we can consider the second partial template to be a generalization of the first. In fact, the first partial template represents so special a case that it is reasonable to outlaw such partial templates, in order to increase the proportion of valid nodes in the layout tree.

In certain situations, however, final templates built with the first partial template may have no equivalents built with the second. Consider the partial template:

```
- - D D D -
- - E E C C
```

where D represents a picture. Since D requires a rectangular space, it can be placed above E and C only if E and C are constrained to be at the same height. Similarly, the top row of a final template will usually contain a number of items, and all of these will of necessity be on the same row.

Even so, there can never be any reason for allowing more than one item top to occur on the same line of a partial template if the tops are non-adjacent. Partial templates such as:

```
- E E - C C
```

are never desirable, since they can lead to neither of the special cases mentioned. They can therefore be considered illegal, potentially increasing the proportion of valid nodes in the layout tree.

5.2.1.4 Lookahead

The top row of a template is the last row to be filled in when using bottom-up layout. By the rules of the template grammar, there must be no holes in the top row. If partial templates are not carefully chosen, though, it might become impossible to fill the top row with the item or items remaining. A lookahead procedure can help to forecast such a situation early enough to avoid it.

For example, in the partial template:

```
- - C C -

- B B B -
```

if there is only one item, A, remaining, then it is necessary for A to have a possible top width of five; otherwise, the top of the template cannot possible be filled in. If there are several template items remaining, there must be some subset with possible top widths totalling five. Similarly, with the partial template:

```
- - - C C

- B B B -
```

either a width of five or a width of three must be achievable.

5.2.2 Augmented Partial Layout Equations

As shown above, generating the partial layout equations is straightforward when proceeding bottom-up. Another advantage of bottom-up layout is that the partial layout equations can be augmented to state a stronger condition than is possible without an inherent direction of layout. The partial layout equations, as shown so far, make no statement related to the items remaining to be inserted into the template.

With bottom-up layout, we know that the remaining items will be inserted above the items already present. This knowledge can lead to an additional equation. With the partial template:

```
- - C C C -

- D D D - -
```

we can generate an additional area equation for the remaining items:[2]

$$6\ h_n + 3\ h_2 + 2\ h_1 \geq \text{area of remaining items} \tag{5-11}$$

This equation is in addition to the area equations:

$$3\ h_1 \geq \text{area (D)} \tag{5-12}$$
$$3\ h_2 + h_1 \geq \text{area (C)} \tag{5-13}$$

Note that if these three equations are added together, we obtain:

$$6\ h_n + 6\ h_2 + 6\ h_1 \geq \text{area of items} \tag{5-14}$$

which is to be expected. This shows that the new equation effectively requires that the total amount of space available in the newshole be sufficient to hold all the news items. This equation, in fact, states a much stronger condition.

This new equation can be called the *remaining-items equation.* To understand its operation, a basic concept is needed. In a given layout problem, there will be more newshole space available than the sum of the areas of the news items. We define the amount of *residue* to be the difference between these two quantities.

At the beginning of template construction, there is positive residue. The remaining-items equation is:

$$\text{total space available} \geq \text{total area of items} \tag{5-15}$$

This is true, since the residue is non-negative.

---

2. If the newshole is non-rectangular, then the numerical sum of the sizes of all levels higher than the current one should be added into the left-hand side of this equation. This is because these levels form part of the space available for the remaining items.

When the first item is added to the template, the symbolic "space available" splits into two parts: the part assigned to that item, and the part left for the remainder of the items. The "space available" is symbolically decreased by the area of the item inserted,[3] and the "area left" is numerically decreased by the same area, so that the residue remains the same.

At all points in template construction, the remaining-items equation requires the residue to be non-negative. After the first template item is inserted, though, the residue can begin to shrink. Consider, for example, the partial template:

```
B B B B - -

D D C C - -
```

The amount of space allocated for D is constrained to be exactly equal to the amount of space allocated for C. If the areas needed for these two items are unequal, then this partial template represents a forced transfer of some of the residue from the "space available" into the space for the smaller of items C and D. The remaining-items equation requires that the remaining residue be non-negative after such a transfer.

Residue can decrease because of a great number of template constructs. If the residue ever becomes negative (strictly speaking, if the maximum value of the residue ever becomes negative), then the partial template is invalid. This will be marked by the failure of the augmented partial layout equations. In such a case, the amount of space available can not possibly be large enough to hold the remaining items. The augmented partial layout equations therefore provide the ability to detect unsatisfiable situations before they would be detected by unaugmented partial layout equations.

3. Strictly speaking, it is decreased by a symbolic quantity, whose achievable minimum in this case is the area of the item.

5.2.3 An Efficient Equation Solution Method

All of the layout equations are simple linear equalities or inequalities. As stated in Chapter 2, a simplex linear programming technique is adequate for solving them. There is an even simpler and much faster method which can be used to solve the class of layout equations generated by bottom-up layout.

This method is primarily intended for determining only whether a set of layout equations or partial layout equations has a solution. As it turns out, this method does produce values of height variables as a by-product, but the simplex method would tend to produce ones of a better quality. That is, better in the quality of the layout produced. This method, if used for final layouts, would produce layouts with the bottommost items leaded as little as possible and the topmost items leaded as much as possible. The simplex method would tend to assign leading more evenly among the items. Still, this method can be used to check sets of partial layout equations, and to quickly check sets of final layout equations before performing the slower simplex method to obtain the numerical dimensions to be used for the final layout.

Consider an arbitrary set of partial layout equations. The layout equations for the items already added to the partial template say only that the space allocated to these items must meet certain minimum requirements, such as area, space for headline, etc.; the finalized height equations, if any, state similar requirements. The other equations are the current height equation and the remaining-items equation. These require that the space left must also meet certain requirements.

For an arbitrary partial template, such as the one shown in Figure 5-3, one way of satisfying these requirements is to find height variables to minimize the space assigned to the items already present, and to simultaneously maximize the space left. Consider the items at the h_1 row. If their tops are numerically

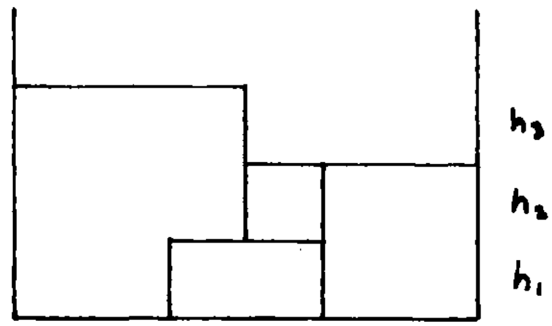


Figure 5-3. Partial Template.

placed as low as they can possibly be, based on these items' layout equations, thus minimizing h_1, this cannot possibly force the tops of any of the other items to be numerically higher than they might otherwise be. This is obvious for the partial template shown. It is also true in general, because of the restrictions on legal shapes of template items.

Similarly, consider the items at the h_2 row. If their tops are now placed as low as possible, minimizing h_2 with known h_1, this cannot cause the tops of any of the other items to move up numerically. Moreover, the choice of minimum h_1 necessarily leads to minimum h_1+h_2 at this point.

This process can continue up to the highest row yet used. This assignment of values to the height variables will simultaneously maximize h_n and maximize the space available for the remaining items. If these quantities are sufficient, then the values of the height variables represent solutions of the partial layout equations. Otherwise, there are no solutions.

This process can be incorporated in the template-building procedure as follows. Each time a new height variable h_i is introduced, it is assigned a value, initially 0. When an item is added whose top is on the height variable's row, the layout equations for that item are generated, and tested for truth with the current value of h_i. If necessary, h_i is increased until the equations become true. The finalizing of an old height equation might also make to necessary to increase h_i. If increasing h_i causes h_n to become negative or the remaining-items equation to become false, then the partial template is invalid; otherwise, it is valid.

Consider an example. The newshole is rectangular, six columns wide and 1500 points tall. Thus, h_n is initially 1500. We shall assume there is 8600 column-points of news to fit into the 9000 column-points of newshole. Story D, with area 1600 column-points, is inserted:

```
D D D D - -
```

This introduces a new height variable h_1. Since we know that:

$$4\ h_1 \geq 1600 \tag{5-16}$$

(considering only area equations here), this sets h_1 to 400, which is the minimum possible. Doing so decreases h_n to 1100 and the available space to 7400. Next, story C, with area 2500, is inserted:

```
- - - C C C

D D D D - -
```

This introduces a new height variable h_2. Since we require that:

$$3\ h_2 + 2\ h_1 \geq 2500 \tag{5-17}$$

this sets h_2 to 566.67, which is (2500-2x400)/3, decreasing h_n to 533.33, and the available space to 4900. Story B, with area 1500, is inserted:

```
- - - B B B

- - - C C C

D D D D - -
```

This introduces h_3, and, since we require that:

$$3\ h_3 \geq 1500 \tag{5-18}$$

we set h_3 to 500, reducing h_n to 33.33, and the available space to 3400. Last, story A, with area 3000, is inserted:

```
A A A B B B
- - - C C C
D D D D - -
```

This requires that:

$$3\ h_3 + 3\ h_2 \geq 3000 \tag{5-19}$$

but $3\ h_3 + 3\ h_2 = 3200$ already, so h_3 does not need to be increased further. All of the items have now been inserted, and h_n is still positive, as is the residue, which is all that remains of the space available (in this case, in fact, none of the residue was used up). This signifies that the final template is valid, as were, of course, all of the partial templates.

As an example of an invalid partial template, consider item C having been inserted as:

```
- - C C - -
D D D D - -
```

This would have set h_2 to 1250, forcing h_n negative.

5.3 Top-Down Layout

5.3.1 Equation Generation Method

5.3.1.1 Basic Method

Top-down layout is quite different. Consider the following partial template with one item:

```
A A A - -
```

Here, there is no immediate knowledge of the symbolic shape of the item A just inserted into the template. It could have any of a number of possible final shapes, depending on the placements of the succeeding items below A. None of the advantages associated with the presence of such knowledge in bottom-up layout exist here in top-down layout.

Even so, it is still possible to generate partial layout equations. Consider what is known in the above partial template. The space allocated for A, although not yet known for certain, will be at least 3 h_1. We can express this as:

$$3\ h_1 + x_A \geq \text{area (A)} \qquad \textbf{(5-20)}$$

where x_A symbolically represents the part of the space eventually to be allocated to A which is not contained in the 3 h_1. We know that:

$$x_A \leq 3\ h_n \qquad \textbf{(5-21)}$$

where h_n, of course, is defined here by:

$$h_1 + h_n = \text{height of newshole} \qquad \textbf{(5-22)}$$

since area A cannot occupy more than three columns at any lower point in the template.

In the subsequent partial template:

A A A B B

similar partial layout equations for B can also be generated:

$$2\ h_1 + x_B \geq \text{area (B)} \quad (5\text{-}23)$$
$$x_B \leq 2\ h_n \quad (5\text{-}24)$$

In the partial template:

A A A B B

- - C C C

the partial layout equations for C can be generated in the same manner:

$$3\ h_2 + x_C \geq \text{area (C)} \quad (5\text{-}25)$$
$$x_C \leq 3\ h_n \quad (5\text{-}26)$$

For A and B, however, matters now become more complicated. For one thing, the space for B is now completely known, so B's layout equations can be generated in final form:

$$2\ h_1 \geq \text{area (B)} \quad (5\text{-}27)$$

plus any necessary headline and picture equations. The area equation for A becomes:

$$3\ h_1 + 2\ h_2 + x_A \geq \text{area (A)} \quad (5\text{-}28)$$
$$x_A \leq 2\ h_n \quad (5\text{-}29)$$

since A will be at most two columns wide at any lower point. Additionally, more is known about the shape of A. We know, for instance, that its headline equation, if any, should be:

$$h_1 \geq \text{headline height of A} + \text{increment} \qquad (5\text{-}30)$$

Also, we know that the shape of A will be non-rectangular, making it illegal if A is a picture. If A contains a dependent picture, the generation of a picture equation might now be possible, depending on the story-picture format chosen.

The point is that information about the exact symbolic shapes of template items is accumulated slowly in top-down layout. Layout equations must be generated and modified as the construction of the template progresses. As more information becomes known concerning the shapes of items, these shapes should be checked against the set of legal shapes.

5.3.1.2 Special Cases

Special cases such as those described in the presentation of bottom-up layout can be handled similarly in top-down layout.

Non-rectangular newsholes can be divided into a number of levels in the same way as in bottom-up layout. Here, there is a choice as to whether the item being inserted is to be in a lower level, and, if so, whether its top is or is not to be constrained to be at the same height as the top of that new level. The definition of the x-variables should be changed appropriately. Finally, in cases such as the partial template:

```
A A A B B B

C C C C - -
```

in a newshole such as the one shown in Figure 5-4, partial headline and picture equations for C should be generated until its shape is more certain than it is in this partial template.

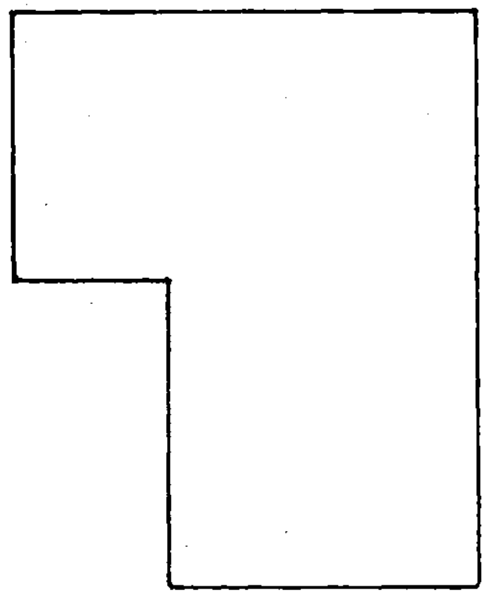


Figure 5-4. Non-Rectangular Newshole.

As in bottom-up layout, there is never any reason for the tops of two non-adjacent items to appear on the same template row.

Since the top row of the template is filled in first in top-down layout, there is no need for complicated lookahead procedures to assure the absence of holes in the top row, as is needed in the bottom-up case. Holes at the bottom, of course, cannot occur, due to the nature of the template grammar.

5.3.2 Augmented Partial Layout Equations

As in bottom-up layout, it is possible in top-down layout to augment the partial layout equations to take into account the set of items remaining. Consider the partial template:

A A A - -

Here, we change the partial layout equations for A to:

$$3\ h_1 + x_A \geq \text{area}\ (A) \tag{5-31}$$
$$x_A + f_A = 3\ h_n \tag{5-32}$$

The latter of these equations can be seen to be equivalent to the formerly defined x-variable equation in terms of the restraints it places on x_A, since all variables in linear programming are constrained to be non-negative.[4] This new equation symbolically divides the space left below A (the 3 h_n) into that which will be used for the remainder of the symbolic shape of A (the x_A), plus that which will be used for other items (the f_A).

These changes in the x-variable equations make it possible to generate a remaining-items equation. In this case, it is:

$$f_A + 2\ h_1 + 2\ h_n \geq \text{area of remaining items} \tag{5-33}$$

The 2 h_1 + 2 h_n appears in this equation because that part of the template is not under any template item, and thus is totally free for the allocation of the remaining items. For the partial template:

```
A A A B B
```

the remaining-items equation would be:

$$f_A + f_B \geq \text{area of remaining items} \tag{5-34}$$

In:

```
A A A B B

- - C C C
```

it is:

---

4. In case of a non-rectangular newshole, the numerical sum of the areas of the lower levels should be added to the right-hand side of this equation.

$$f_A + f_C \geq \text{area of remaining items} \qquad (5\text{-}35)$$

Of course, the layout equations for B in this latter partial template will no longer include references to x_B or f_B, since the exact symbolic shape of B is known.

5.3.3 Absence of an Efficient Equation Solution Method

For bottom-up layout, there exists an efficient method available for solving the layout equations generated, in the order generated. No analogous method exists for top-down layout.

Consider the partial template:

```
A A A A B B

- - - C C C
```

where the item areas are as shown in Table 5-:1, and the newshole height is 1500 points. Given this partial template, it might seem that it would be simple to generate a value for h_1, namely 500, the minimum possible one. Consider, however, the final template:

Table 5-I. Example Item Areas.

area (A) = 3800 column-points
area (B) = 1000 column-points
area (C) = 900 column-points
area (D) = 1200 column-points
area (E) = 1500 column-points

```
A  A  A  A  B  B

-  -  -  C  C  C

E  E  E  -  -  -

-  -  -  D  D  D
```

Here, it develops that there is only one solution possible, which is shown in Table 5-II; this solution has a greater-than-minimum h_1. On the other hand, the minimum h_1 is achievable in this final template:

```
A  A  A  A  B  B

-  -  -  C  C  C

-  -  -  E  E  E

D  D  D  -  -  -
```

which is also based on the same partial template. Indeed, the minimum h_1 possible with the partial template corresponds to the only solution with this final template.

The point here is that it is not possible in top-down layout, based only on a partial template, to determine universally acceptable values for the height variables which can be used regardless of the placement of the remaining items. Instead, the set of partial layout equations must be kept and completely checked at the point of insertion of each new item, without such a shortcut as is possible with bottom-up layout. It is interesting to note that the only

---

Table 5-II. Solution to Layout Equations.

$$h_1 = 800$$
$$h_2 = 200$$
$$h_3 = 100$$
$$h_4 = 400$$

exception to this general rule is when all template items are rectangular. This is because rectangles are the only legal template item shapes which are still legal upside-down.

One computational shortcut that is possible, given a partial or final template constructed top-down, is to reconstruct the template bottom-up, checking for validity in doing so. This method, of course, works perfectly as a validity check for a final template constructed top-down (or, indeed, in any manner); for partial templates, though, it cannot yield a correct value for the space remaining, because of the mismatch in layout directions. Thus, for partial templates, this method can be used only for checking a subset of the necessary conditions stated in the layout equations. Still, the procedure is fast, and, if it fails, then the more complicated linear programming method need not be tried, because it would certainly fail also. If the procedure succeeds, then linear programming must still be used to check the validity of the layout equations augmented with the remaining-items equation.

For final templates, this shortcut always gives the same result as linear programming. For partial templates, the correlation grows less as the template grows less complete.

5.4 A Comparison

The previous sections outlining the use of bottom-up layout and of top-down layout strongly suggest that the layout game is inherently easier to play when proceeding bottom-up than when proceeding top-down. The important point of comparison between them, though, is to determine which one makes the layout game easier to win. The answer, surprisingly, is that it is much easier to win the layout game when proceeding top-down than when proceeding bottom-up.

Consider the layout problem given in Table 5-III. This example is taken directly from Page 8 of the February 9, 1977 issue of Tech Talk, an M.I.T. administrative newspaper of tabloid size (this page is shown in Figure I-1 in Appendix I). In this example, as in others in this report which are based on actual newspaper layouts, a number of possible headline sizes are included in the layout problem listed, all based on the single headline size actually used in the layout, so as to make the problem more representative of actual layout problems.

The complete layout trees for this example layout problem have been experimentally enumerated, both for the bottom-up layout case and the top-down layout case. There were found to be exactly 66 distinct layouts possible for this layout problem; therefore, the layout trees for both bottom-up layout and for top-down layout have exactly 66 valid legal final nodes. What matters, though, is how difficult those 66 nodes are to reach.

The layout tree, ignoring descendants of invalid nodes, contains 4317 nodes in the bottom-up case, but only 1011 nodes in the top-down case. More statistics are shown in Tables 5-IV and 5-V. We see that proceeding top-down gives a significantly smaller layout tree for this example than does

---

Table 5-III. Tech Talk Layout Problem.

name	text	headline	picture
Engineering	2252	3x40, 4x40	-
Dancing	1121	2x80, 3x40, 4x40	3x270
Symposium	589	2x60, 3x30, 4x30	-
AARP	-	-	2x261

The newshole is rectangular, 5x1160.
Formats 1, 2, and 3 are legal format choices for the Dancing story.
The increment in the headline equations is equal to the headline height.
The convexity rule from Section 7.2 is an absolute rule.

Table 5-IV. Bottom-Up Layout Tree Statistics for Tech Talk Example.

	Ply 1	Ply 2	Ply 3	Ply 4
No. of posns w/valid final descendants:	11	44	64	
No. of posns w/valid descendants:	22	141	64	
No. of valid posns:	22	235	424	66
No. of posns:	23	332	2174	1787

Table 5-V. Top-Down Layout Tree Statistics for Tech Talk Example.

	Ply 1	Ply 2	Ply 3	Ply 4
No. of posns w/valid final descendants:	6	17	40	
No. of posns w/valid descendants:	7	30	40	
No. of valid posns:	9	36	154	66
No. of posns:	9	38	291	672

proceeding bottom-up. The top-down tree contains fewer dead ends and blind alleys than does the bottom-up tree. As a measure of this, consider a tree-searching algorithm which starts at the root node, and repeatedly moves to some valid descendant node selected at random, stopping either when a valid final node has been reached or a dead end has been detected.[5] The probability of success of a single pass of this simplistic search algorithm can be derived from the information in the two tables; it is 0.091 with bottom-up layout and 0.168 with top-down layout.

As noted in Section 5.3.3, there is a very efficient equation solution method possible with bottom-up layout which has no analogue under top-down layout. This method reduces the amount of time necessary to check a given node in the layout tree for validity. This time reduction, however, is wiped out by the relatively greater number of nodes that must be checked in bottom-up layout.

5. More sophisticated tree-search algorithms are the subject of the following chapter.

Generating these layout trees required 359 seconds bottom-up but only 287 seconds top-down,[6] even with the more efficient equation solution method possible with bottom-up layout. The difference in these figures is made even more significant by the fact that the implementation of bottom-up layout used to generate the layout tree was a drastically stripped version which was suitable only for generating the bottom-up layout tree as quickly as possible, while the one for top-down was a relatively inefficient general-purpose top-down layout program adapted for generating the layout tree.

There are many sources of the advantages of top-down layout over bottom-up layout. For example, the top of the page is laid out first with top-down layout. This eliminates the need for the complicated and less-than-perfect look-ahead procedures that are necessary with bottom-up layout for ensuring that the top of the page can be filled in at all. Furthermore, since the top of the page "governs" the rest of the page, a partial template going top-down tells more about the overall nature of the layout than a partial layout going bottom-up. Top-down layout places the stories at the bottom of the page last. There are a number of possible legal placements for these items; this increases the probability of some valid placement, and helps to avoid a dead-end late in the tree. Bottom-up layout, on the other hand, places the last items at the top of the page, where there are only a few placements possible if the top of the template is to have no holes, thus reducing the probability of one of these few placements leading to a valid final template.

Dead ends can be detected earlier with top-down layout than with bottom-up layout. Experience with both top-down and bottom-up layout algorithms shows that this is true not only for the Tech Talk example, but also

---

6. These timing figures are, of course, based on a particular implementation running on a particular machine. More information on the timing characteristics of this implementation are given in Chapter 9.

in the general case. The number of nodes in the complete layout tree grows exponentially with the number of items.[7] Since dead ends are detected earlier with top-down layout than with bottom-up layout, and since there are simply fewer dead ends to find, the ratio between the number of nodes to be examined with bottom-up layout and with top-down layout grows exponentially with the number of items. On the other hand, the ratio between the speeds of the equation solution methods of bottom-up layout and top-down layout grows at only a polynomial rate,[8] which becomes insignificant. This suggests that, for any layout problem of equal or greater complexity than the Tech Talk example, top-down layout will be faster than bottom-up layout. For simpler problems, bottom-up layout may be faster, but, since the amount of time taken by either direction of layout should be relatively small, it would not be unreasonable to use top-down layout for these cases too.

This discussion has been based on the overall shape of the layout trees for bottom-up layout and top-down layout, showing that it is reasonable to believe that top-down layout will be faster than bottom-up layout. In actual practice, this is true. The problem with bottom-up layout is that it is difficult to control its path toward a layout, because of the shape to its layout tree. It will often achieve partial templates which look distinctly improbable to a human observer, because of their geometrical shapes, but which are nonetheless valid at that

---

7. It is this fact that makes it effectively impossible to generate the complete layout trees for any examples larger than the Tech Talk example. Another example was tried, which also had four items, but with a greater relative choice of possible headline sizes (and therefore of possible top widths). This example lead to 3080 distinct layouts, with 14038 nodes in its layout tree in the top-down case, requiring over 4500 seconds (one and a quarter hours) to generate. The bottom-up layout tree was not attempted. An attempt at a complete top-down layout tree for an example with five items was halted partway through; it is estimated that it would have required several hours to exhaustively enumerate. This exponential increase is related to the exponential increase in the number of possible templates, explored in Appendix II.

8. That is, the equation solution method for bottom-up layout takes constant time per item, and thus linear time overall, while linear programming can be cleverly implemented to take quadratic time per item, and cubic time overall.

ply. These partial templates later turn out to have no valid final descendants. In top-down layout, however, a human observer is usually unable to predict the probability of success of a partial template much better than do the partial layout equations.

5.5 Summary

Choosing a fixed direction of layout not only drastically reduces the size of the layout tree, but also allows the partial layout equations to be augmented by a remaining-items equation. This strengthens the condition on valid intermediate positions, assisting in the earlier detection of dead ends. Although the bottom-up layout game is easier to play, the top-down layout game is easier to win.

The following chapter will consider the problem of efficient searching of the top-down layout tree.

Chapter 6 - Layout Tree Searching

Once the form of the layout tree has been made more certain by the choice of top-down layout, it is possible to study the effectiveness of the various possible tree-searching procedures. All of the search strategies presented in the chapter have been implemented and tested. The analyses presented are based on the experience thereby obtained.

6.1 Exhaustive Search

One search procedure is *exhaustive search*. An exhaustive search procedure simply searches the entire layout tree, in a depth-first (backtracking) search, looking for valid final nodes. Remember that moving down through the tree corresponds to adding to the partial template under construction, combined with checking the new partial layout equations for satisfiability, while backtracking up the tree corresponds to undoing a previous choice, returning the partial template to a former valid state before another possibility is tried. When a valid final node has been found, the corresponding layout is output. We assume that the layout person has control over whether the search procedure will then terminate, or whether it will continue to search for further valid final nodes.

This search procedure has the advantage of simplicity. It does, however, present certain problems. A layout tree is usually too large to search completely in any reasonable amount of time. Assume that, at each ply, the search procedure cycles through the various possible valid moves, trying each in turn. If there are only three or so items on the page, this search procedure will be able to traverse all the possible paths through the layout tree in a reasonable amount of time. If there are more than about three items, not all of the tree can be searched within the time available. That part of the tree reachable by an exhaustive search procedure will all be based on the same move at the first ply (since the procedure will not have exhausted the consequences of that move yet), the same move at the second ply, and so forth, with only the moves on the last three or so plies examined fully. This will result in a noticable lack of variety in the final templates tested for validity, since the top of the page will be identical in each of them. This has two consequences.

First, all the layouts produced will look quite similar. If the top of the page resulting from the single early choices that could be tried looks good, then this is no problem. If, however, it represents an esthetic "mistake", then the layouts produced will tend to be equally unacceptable.

Secondly, all the templates generated will have similar layout equations. Experience shows that valid nodes tend to cluster together in various parts of the tree, instead of being evenly distributed. This relates to the fact that various early choices in building the templates are better than others for reasons not expressible by the dichotomy of simple validity or invalidity of partial templates. If that part of the tree visited first happens to be relatively dense in valid final nodes, exhaustive search will be effective. If that part of the tree has no valid final nodes, then exhaustive search will fail because of a pragmatic "mistake" having been made early in the search.

It is possible to design a tree-searching algorithm so that it can make certain judgements concerning the probability of the validity of descendants of a given node in the layout tree. One such judgement, of course, is in the distinction between validity and invalidity of a node; a more sophisticated judgement is discussed in Section 6.5. It is also possible to design the tree-searching algorithm so that it can make certain judgements concerning the eventual esthetic qualities of layouts based on descendants of a given node. One such judgement is related to the rule of trying to place the more important news items nearer to the top of the page than the less-important ones.[1]

The use of such judgements of perceived template quality can improve the efficiency of searching, but neither of these judgements is foolproof. It will be still possible to make mistakes in choosing the early moves which, under exhaustive searching, cannot be taken back. Paradoxically, exhaustive searching can fail because it can not search enough of the tree. Instead of exhaustively searching what must be at most a very small and closely-related part of the tree, it might be better to try to search incompletely through a greater part of the tree. Such a search strategy will not work as well as exhaustive search does in the best case, but it should represent an improvement over the worst case.

6.2 Basis for Random Searching

Only a small part of the complete layout tree for any reasonably large layout problem will be practical to search, no matter what particular search strategy is used. The search strategy must select, from the set of all moves that could be made, those that will actually be considered. How are these moves to be selected?

In this thesis, the premise is that the moves are selected at random, with probabilities related to their perceived quality. This may seem a strange choice at first. It might seem more reasonable always to use those moves which have the highest perceived quality, without running the risk associated with a lower quality one. There are, however, reasons why such an analysis is incorrect.

As mentioned in the last section, the perceived quality of a move is not a

1. Others esthetic judgements are discussed in Chapter 7.

perfect indication of its actual worth, to which it is related only statistically. In other words, there is uncertainty over the conditions governing the choice of a move. Game Theory[2] is a branch of mathematics which deals with decision-making in uncertain situations. It shows that, in many situations where one has a choice of strategies to be pursued, selecting randomly from the strategies according to mathematically derived odds can lead to a higher degree of success than simply following the most reasonable single strategy. This suggests that such an approach might also be reasonable in searching a layout tree, where there is incomplete information about the quality of possible moves. Unfortunately, a direct application of game-theoretic analysis to this problem seems impractical, precluding the possibility of deriving optimal odds for the various moves. It is necessary to use rule-of-thumb measures of move quality for this purpose. In this way, we can at least increase the probability of success of a partial tree search.

If a probabilistically directed partial search of the tree fails to produce a layout, the search can be repeated until a layout is found. Alternatively, the layout person may choose to intervene after a reasonable amount of time has passed. If a probabilistic search of the layout tree fails to produce a layout after a reasonable amount of time, it is more reasonable to conclude that there are either no layouts in the tree, or very few, than it would be with other search techniques. If layouts do exist, as is usually the case, this procedure will eventually find some. One important measure associated with such a search procedure is the amount of time required to find the first layout, or the first few. The use of a probabilistic search algorithm can be expected to improve the variance of this amount of time as compared to the exhaustive-search case, since it is not as badly affected by the clustering of the valid

2. Game Theory, despite its name, is only indirectly related to the study of games such as chess, and to elements of that theory, such as game trees.

nodes.[3] Thus, it will increase the probability that a layout will be found within the fixed amount of time available. Also, the layouts achieved should prove less similar to each other than those produced in the exhaustive-search case, increasing the set of real choices available to the layout person.

6.3 Breadth-1 Search

The *breadth-1* search procedure was briefly introduced in the last chapter. This procedure starts at the root node of the layout tree and attempts to form a single continuous path to a valid final node by continuously adding purely random valid moves. Eventually, either a valid final node is reached, in which case the corresponding layout is output, or a dead end is reached, in which case the algorithm starts again from the beginning.

Discarding the entire path thus constructed might not be the optimal strategy here. However, it does have the advantage of simplicity, and is not totally unjustified. It is known that the path leads nowhere, and it is not known which particular move or sequence of moves is to blame for this. Starting an entirely new path represents cutting our losses, rather than continuing to spend time trying to patch the path.

This algorithm will eventually produce layouts. As might be expected, it performs less well as the number of news items grows. Its implementation, however, is quite simple. At each intermediate point, there is somepartial template representing the *current position*, from which a new position is chosen by randomly selecting from among the set of possible legal moves until a valid

3. Rabin [Rabin 76] has studied the class of probabilistic algorithms, and shows that the time performance of many algorithms can be improved through the use of randomization. For example, certain sorting algorithms have good average-case running times, but bad worst-case times. If the inputs to such an algorithm are first randomized, then all sets of inputs of a given size will have an expected running time equal to the average running time of the algorithm for sets of input of that size.

one is found.

6.4 Breadth-N Search

One improvement on breadth-1 search is to let the set of current positions kept at each ply be of size N, for some constant N possibly greater than 1. Here, each step of the search involves choosing N valid moves from the moves possible from the N current positions. (Of course, there will be only one current position, the empty template, at the start, and exceptional numbers of dead ends may force a smaller number of current positions at other plies.) At the end of a successful search, there will be N final layouts produced. This search strategy will be called *breadth-N* search. If N=1, then breadth-N search is the same as breadth-1 search.

On the average, we can see that this procedure will take approximately N times as much time as breadth-1 search, ignoring the occurrence of dead ends. On the other hand, dead ends do occur frequently, and are the source of the potential superiority of breadth-N search over breadth-1 search.

Consider a layout tree for five items, where each valid node has a 0.5 chance of being a dead end, except for the root node and final nodes. With breadth-1 search, there is a chance of only 0.5^4, or 0.0625, that a layout will be found in one pass, requiring an average of 16 passes to find a layout (the 4 in the exponent comes from the fact that there are four moves following the first). With breadth-2 search, the probability that both of the current positions will be a dead end is only 0.25, giving a 0.75 chance of success at single ply, and a 0.75^4, or 0.316, chance of success in one pass, requiring an average of only 3.16 passes. Thus, a doubling of the amount of work performed at each ply can lead to the number of passes through the tree being better than halved. Further increases in N lead to further overall improvements, up to a point, after which the probable number of passes becomes so close to unity that diminishing returns set in.

The above analysis is quite simplistic. To confirm its results, an experiment was performed in which the complete top-down layout tree for the Tech Talk example given in Chapter 5 was used to simulate the breadth-N search procedure for various values of N, thereby determining the expected amounts of time that would be required. This required the simulation of several hundred searches for each N, more than would be practical if the complete layout tree were not available. Figure 6-1 shows the expected times as a function of N. Time is expressed in seconds, estimated from performance measurements of the layout program. The curve represents the amount of time required to achieve one layout.[4] The vertical lines for each N

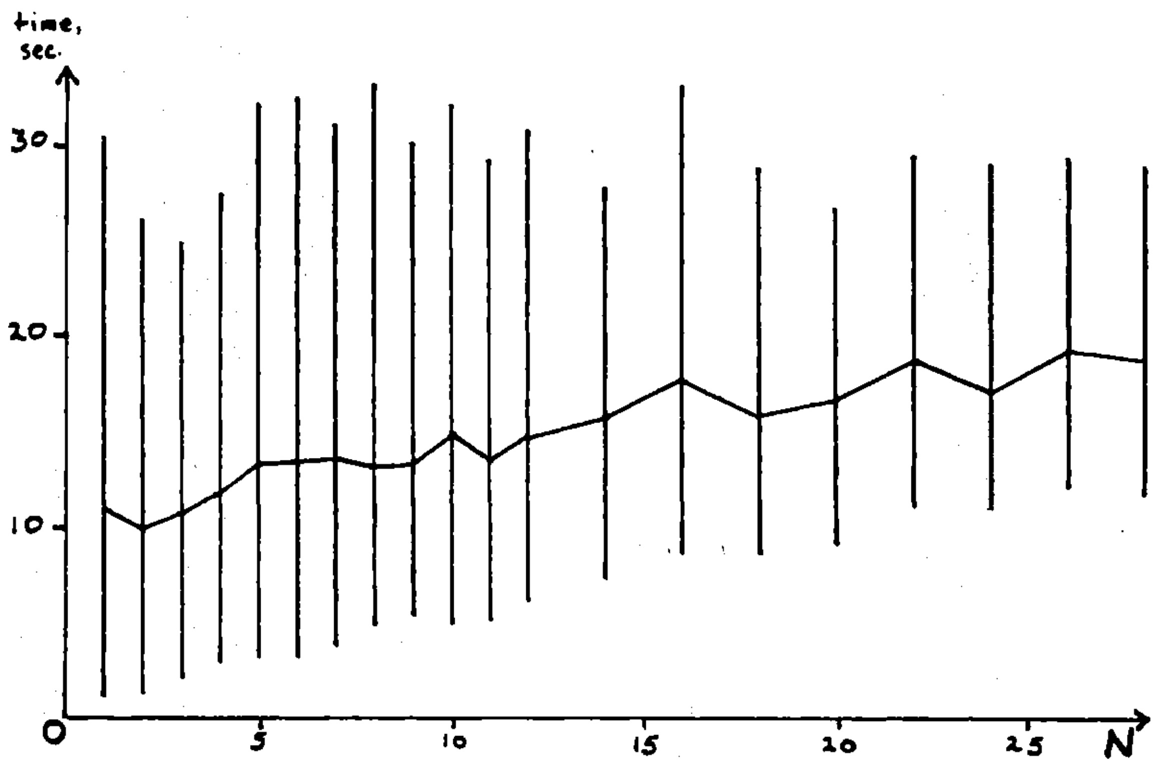


Figure 6-1. Average Times for Breadth-N Search for Tech Talk Example.

4. Of course, for large N, not much more time would be required to achieve some number of layouts after the first.

represent 90% bounds for the distribution of the times.

As can be seen, the expected time reaches a minimum here at N = 2, and then increases with increasing N.[5] The variance in the amount of time required decreases with increasing N. Thus, breadth-N search can be seen to represent an improvement on the less-general breadth-1 search.

For the sake of interest, part of the equivalent graph for the bottom-up layout tree for the Tech Talk example is shown in Figure 6-2. For this example, it achieves approximately the same minimum expected time, but at a

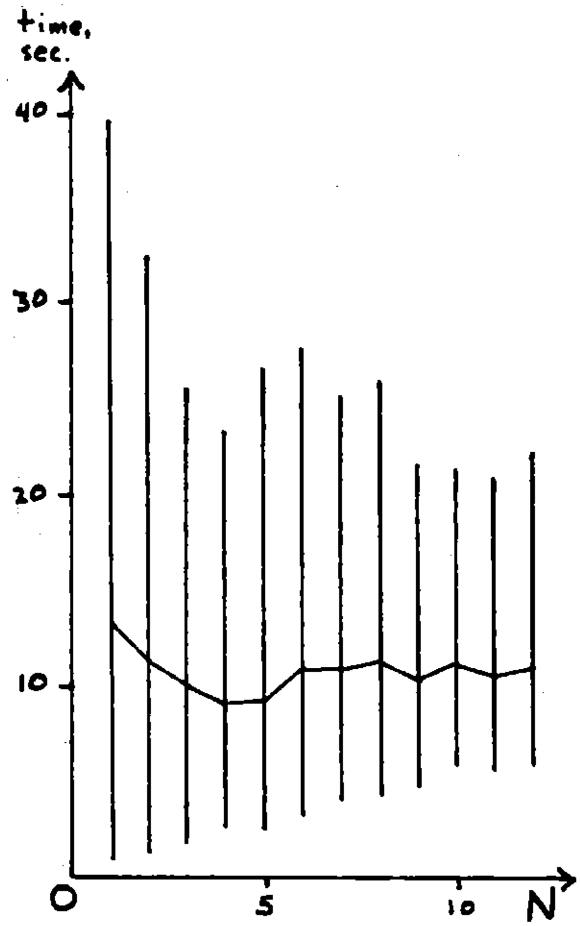


Figure 6-2. Average Times for Breadth-N Search for Tech Talk Example (Bottom-Up Layout).

5. The irregularities in the curve are probably caused by the finite sample size.

relatively greater N, reflecting the greater proportion of dead ends in the bottom-up layout tree. It is believed that the optimum value of N also grows as the number of items increases.

In the above analysis of breadth-N search, it was stated that an increase in the number of current positions would lead to a certain decrease in the probability of failure at a given ply. There was an implicit assumption that the probabilities of failure of each of the individual positions were statistically independent. To the extent that this is true, the analysis is correct. At the other extreme, if the probabilities of failure were perfectly correlated, then breadth-N search will be no more successful than breadth-1 search, but will require N times as much work. In general, some correlation between the probabilities of failure of the current positions will lead to some decrease in the effective breadth of search.

This consideration suggests that there should be some attempt made to keep the set of current positions uncorrelated. Consider breadth-2 search. At the first ply, there will be two current positions, from which two new positions are selected. These two current positions are "brothers" in the layout tree. If the two new positions at the second ply are based on different old positions, then they will be "cousins" in the tree to each other. If they are both based on the same old position, however, then they will themselves be brothers. For the Tech Talk top-down layout tree, analysis shows that the coefficient of correlation of eventual failure is 0.289 between brothers but only 0.008 between cousins.

This leads to differences in effectiveness between the possible ways to select the next N positions. One possibility is to assign all of the descendants of the N current positions equal weights. Another is to divide the descendants into N sets, based on which of the N current positions they are derived from, give each of these sets equal aggregate weight, and give the members of each set equal weight among themselves. In other words, to pick a move to check for validity, it will be necessary first to pick at random a current position on which to base it, and then to pick at random one of the legal moves from that position.

Certain of the current positions will have more valid moves than will others, usually because they simply have more legal moves. Here, the former "one-bucket" approach will tend to give more weight to such such positions than will the latter "N-bucket" approach. On the other hand, if certain positions have a higher proportion of valid moves than others (that is, if they are more "robust"), then N-bucket move selection will tend to give them higher weight. Experience and experimentation show the N-bucket approach to work better. One-bucket move selection has too great a tendency to select multiple descendants of the same prolific current position; prolificy is less of an indication of quality than is robustness.

An improvement that works even better is to similarly subdivide the moves in each of the N sets, depending on which item is being inserted by the move. This keeps moves involving independent pictures, which have only one possible top height, from being swamped by moves involving stories, which can have several possible top heights. Otherwise, moves inserting independent pictures have much lower aggregate probability than moves inserting stories, producing a tendency toward leaving pictures to be inserted last, which is unpleasing esthetically as well as difficult practically.

A further possible improvement to ensure that brothers occur as seldom as possible might be never to pick more than one move from the same current position, unless this becomes necessary because of the occurrence of dead ends. Such a method might reduce the number of passes necessary, but, experience shows, requires a roughly equivalent increase in the amount of time spent at each ply. Some of the positions are simply difficult to get valid moves from, being close to failure though not yet failing (this concept is explored more closely in the following section). The expense of squeezing the inferior moves from these positions does not seem to be reasonable.

6.5 Breadth-(N_1,N_2) Search

In the previous chapter, the concept of residue was introduced, as a measure of the difference between the amount of space available for the remaining items and the sum of the areas of those news items. The amount of residue decreases monotonically as one moves downward on a path through the layout tree. If the amount of residue ever becomes negative, this signals an invalid position. Figure 6-3 shows how the average amount of residue decreased in one particular breadth-10 search of a layout tree for a layout problem with 12 items (such a large value of N as 10 is used to provide a smoother graph). The first few items are all laid out on the top line. The amount of residue cannot decrease while they are being laid out, except for very small amount related to new-found knowledge of exact headline sizes.

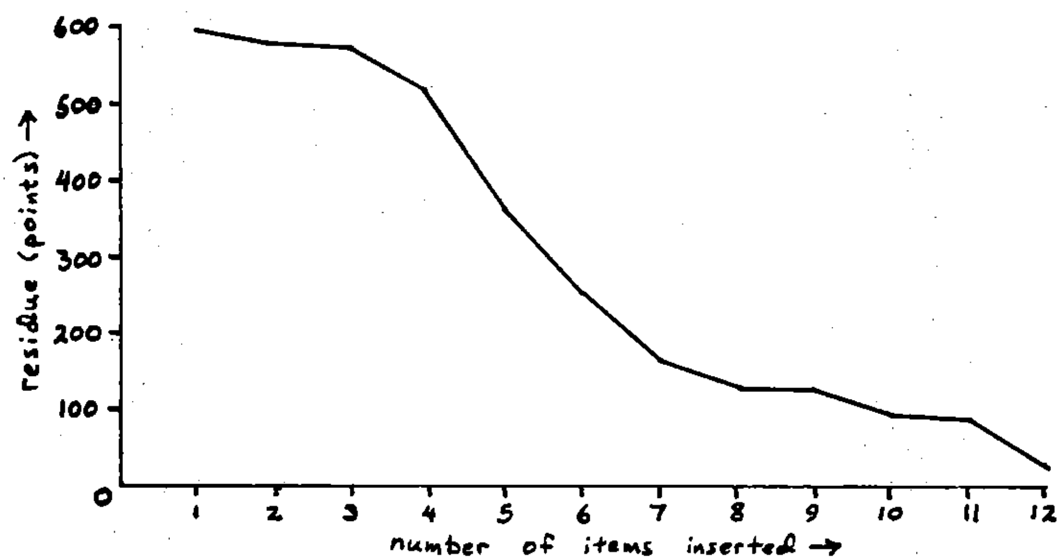


Figure 6-3. Decrease of Amount of Residue in Breadth-10 Search.

After the first few items, the amount of residue begins to drop almost linearly, until it approaches zero. At this point, many moves would lead to negative residue; these, being invalid, are ignored. Still, the amount of residue moves asymptotically closer to zero, making it harder and harder to find a valid move.

During the linear section of the curve, no effort is being made to try to minimize the amount of residue lost, which is why the curve approaches zero so rapidly. If the amount of residue associated with a position could be directly computed, however, such information could be used to avoid this situation. To compute the residue, we can remember that the remaining-items equation is of the form:

$$\text{space remaining} \geq \text{area remaining} \qquad (6\text{-}1)$$

for symbolic "space remaining" and numerical "area remaining". This equation is used to form a set of augmented partial layout equations, which are then checked to see if any solution exists. A different approach would be, instead of including this equation explicitly in the set of layout equations, to use linear programming to maximize the "space remaining" within the constraints given by the other equations.[6] The known total area of the remaining items can then be subtracted from this quantity to compute the residue. If the residue is non-negative, then the position is valid. Moreover, the exact amount of residue is thereby known and can be used to judge the quality of the position.

A relatively large amount of residue remaining can be viewed as an indication that a position is more likely than average to have valid descendants. To understand this, consider that a position has some number of possible legal sequences of moves following it. Each of these uses up some amount of residue. The greater the amount of available residue associated with the

6. This is done implicitly by the bottom-up equation solution method discussed in the previous chapter.

current position, the greater the proportion of these move sequences that will be valid.

Another measure of position quality that can be used in conjunction with the amount of residue is the total area of the remaining items. In general, templates with the larger items placed at the top of the page are more successful than those with the large items placed at the bottom of the page.[7] Thus, a template with a smaller "area remaining" should tend to be chosen over templates with larger ones. The smaller items will be easier to place than would larger ones. One numerical measure of position quality that experience has proven useful is (residue)/(area remaining). When this expression is maximized, this tends to maximize the amount of residue while minimizing the area remaining. This function will be used as the criterion function related to the probability of valid descendants of a position. Note that, since residue = space remaining - area remaining, maximizing the criterion function is the same as maximizing (space remaining)/(area remaining), which seems reasonable, considering the overall relation in layout problems between this factor and ease of layout. This relation is explored in more depth in Appendix II. Note that this criterion function predicts that it is harder to solve layout problems with news items all of the same size, or nearly so, than it is to solve problems with the same number of items but where there is a great dissimilarity in sizes, since the maximum decrease of the "area left" in the first case is less than in the second. Experience shows this to be the case.

The availability of a criterion function leads to *breadth-(N_1,N_2)* search. Here, there are N_1 current positions, and there are N_2 valid descendants chosen, where $N_2 \geq N_1$; the value of the criterion function is computed for each.

---

7. This criterion is related to the numerical probability of template success, and should not be confused with the esthetic criterion whereby important news items should be placed near the top of the page, although these two criteria usually work toward the same goals.

Finally, the N_1 of these with the highest values of the criterion function are chosen as the N_1 new current positions. Of course, if N_1=N_2, then breadth-(N_1,N_2) search is equivalent to breadth-N search.

This procedure gives roughly the same results as would be obtained by selecting N_1 valid moves with all valid moves being weighted according to their values of the criterion function, but requires the computation of the criterion function for only N_2 valid positions. Checking for validity, of course, takes time, so it is desirable to limit checking to as few positions as possible.

Figure 6-4 shows the results of using breadth-(10,15) search. As can be seen, the amount of residue decreases significantly more slowly with breadth-(10,15) search than with breadth-10 search, since the effort to keep the residue large begins earlier. The presence of upward slopes on this curve may seem confusing at first, since adding to a partial template can only decrease the residue left, not increase it. Actually, this curve represents the average amount of residue in the ten positions at each ply. At plies where

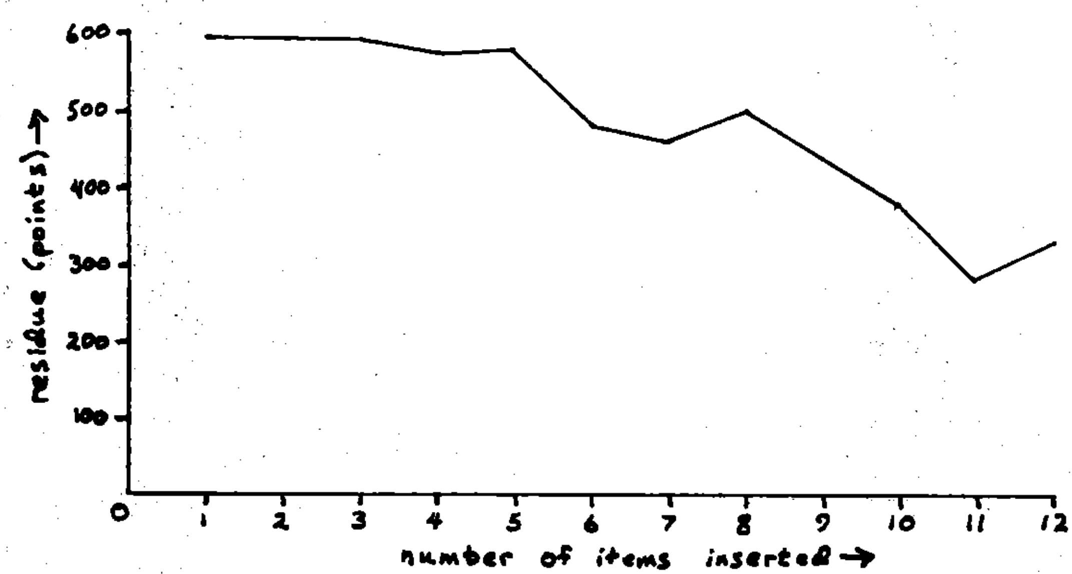


Figure 6-4. Decrease of Amount of Residue in Breadth-(10,15) Search.

proportionately many of the positions are derived from a relatively small set of high-residue positions in the previous ply, the average amount of residue can, and sometimes does, increase.

This consideration points out that breadth-(N_1,N_2) search does tend toward the selection of somewhat correlated positions at a given ply, where the correlation is related to the values of the criterion formula. This correlation is related to a correlation between the probabilities of failure of the positions, and thus works partially at cross purposes to the principle expressed in the previous section of selecting uncorrelated positions. Still, moderate use of breadth-(N_1,N_2) search can result in faster layout.

Figure 6-5 illustrates the expected amount of time required for breadth-(N_1,N_2) layout for various values of N_1 and N_2, for the Tech Talk example. These figures were obtained in a similar way to the breadth-N figures. The minimum is reached here at (N_1,N_2) = (2,4), with an average time

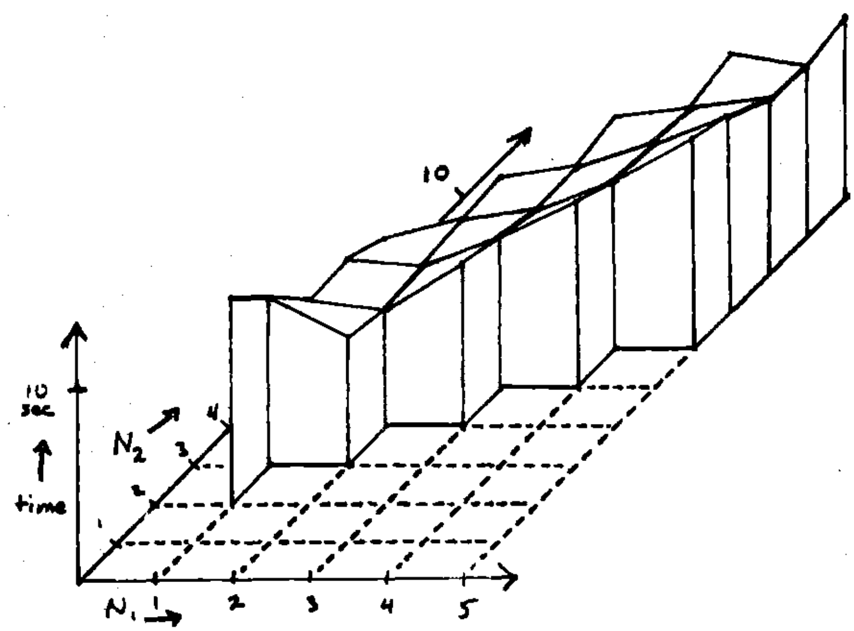


Figure 6-5. Average Times for Breadth-(N_1,N_2) Search for Tech Talk.

of 6.1 seconds. This shows that breadth-(N_1,N_2) search can provide a significant improvement over the less general breadth-N search.

6.6 Other Approaches

Other approaches to searching layout trees are possible. However, none of the alternatives tested seem as well adapted to the particular nature of layout trees as the ones described above. It seems that any efficient searching algorithm should be based on one of these.

One approach that has not been completely tested is the possibility of variable-breadth search, where the breadth of the search can vary with the depth of the ply. This would let the breadth increase toward the bottom of the tree, where proportionately more work is needed to avoid dead ends. Such an increase can be effective, but it seems difficult to formulate general rules of when and by how much the breadth should be increased for the best results.

A limiting case of the above is to increase the breadth toward infinity near the end of the search, which is the same as having a randomly ordered exhaustive search. Doing so for the final ply is effective, since further layouts can thus be achieved cheaply, though they will be similar to the earlier ones. Doing so for forming the next-to-final ply is also effective,[8] but doing so any earlier requires too many positions to be checked.

6.7 Summary

This chapter has examined the problem of searching through a layout tree. A simple exhaustive search is shown to be inappropriate, and a partial probabilistic search to be more promising. A series of probabilistic search

8. In such a case, keeping lists of all the current positions will be too space-consuming, since their number will be so great. Instead, a backtracking search should be used.

techniques was considered, culminating in a breadth-(N_1,N_2) search with a fast exhaustive search at the end. Experience and experimentation show this method to be the best of the ones considered. Success in achieving layouts was the measure of quality of a search technique used in this chapter.

Now that it is clear that layouts can be reached by searching the layout tree, the next chapter is devoted to the related problem of achieving esthetically pleasing layouts.

Chapter 7 - Template Esthetics

7.1 Basic Procedure

Preceding chapters have shown how layouts for a given layout problem may be generated by searching the layout tree for valid final nodes. The procedures shown, although adequate for their task of finding some layout, made no special effort to find a "good-looking" layout.

The rules of template grammar given in Chapter 3 are hard-and-fast rules on template construction, which may never be violated. They are few in number and provide only enough restrictions on the form of templates to allow the layout procedures to operate. There are other rules which one might wish to state, which make more subtle judgements on template quality. Such rules may be broken, but layouts will tend to be of better quality if they are not. On the other hand, these rules give only a partial indication of template quality.

The approach used to implement such rules will be to have the moves selected for consideration tend to be moves judged esthetically good. Thus, we will in effect be combining the criterion function described in the last chapter, which judges a partial template on a numerical basis related to its perceived ability to produce layouts, with esthetic criteria related to how good-looking the template is perceived to be. This could present a problem if these criteria tended to oppose each other. In actual practice, this does not seem to be the case. All of the esthetic criteria considered here seem at worst to make no real difference in the probability of a successful layout, and many seem to improve the probability. This interesting fact is possibly due to the tendency on the part of newspaper readers (and especially editors) to consider commonly used layout types as esthetically pleasing, coupled with the tendency on the part of newspaper layout people to produce layouts which are easy to achieve.

This chapter considers a representative set of possible rules governing template esthetics. These rules make their judgement based solely on the form of the template or partial template. Thus, they cannot incorporate certain judgements based on numerical properties of the eventual layout. For example, it is considered poor form to have a headline placed exactly half-way down the first page of a section, since that is where the fold in the paper occurs. It is hard to see how such a consideration could be detected in the framework of basing esthetic criteria on properties of the templates alone.

As was argued in the last chapter, rules for move selection are best used as the basis for the assignment of probabilities in a probabilistic search of the layout tree. One satisfactory way to implement these esthetic rules is to use them to assign a numerical quality factor to each legal move, and to use this factor to control the random selection of legal moves for testing for validity.[1] Each time a legal move is to be chosen, the probability that a given move will be selected is proportional to its quality factor. This bias will affect the order in which the moves are chosen, and will tend to have the more highly rated moves chosen before the lower rated ones. Since the search procedures discussed in the previous chapter require only some certain number of valid moves, this method will tend to make the chosen moves be more esthetically pleasing than average.

These esthetic rules will be rules on the form of final templates, and will list certain bad features which should not be present. These bad features can be detected when they are incorporated into partial templates. For each newly constructed partial template, there might be some set of bad features present in it which were not detectable in its immediate predecessor. Each of these bad features can be considered to have some numerical weight, associated with

1. In case of a multiple-bucket selection procedure, such as described in the last chapter, each bucket should have assigned to it a derived aggregate quality.

its severity; the worse the feature, the greater the weight. The sum of these weights can be used to generate the quality factor for each position. One formula for the quality factor that has been found useful is 1/(1+sum). Thus, the poorer a position, the greater the sum of the weights for the position, thus the less the quality factor, and the less the probability that the position will be selected.

The rules listed here are only part of the set of possible esthetic rules. In experimental implementations, the choice of which rules to include, and what weights to assign to them, was based on experience. Further research is needed to understand this problem better.

7.2 Item Shapes

Templates containing extremely unusual item shapes are undesirable. For example, in the template shown in Figure 7-1, the topmost item has an obviously undesirable shape. This item violates the *convexity rule,* which states that there should not be such holes in the shapes of items (i.e., no horizontal line drawn through an item should pass through the item twice or more).[2] Violations of the convexity rule should be given great weight, except

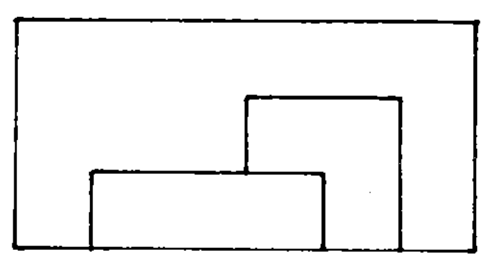


Figure 7-1. Template with Item of Unusual Shape.

2. The presence of this feature could not be detected at the point when the second item was inserted, since there would be no way of knowing that the third item would not be inserted immediately to the right of the second item, extending to the end of the row. Such a placement would result in a shape for the top item that did not violate the convexity rule.

possibly for the common case shown in Figure 7-2, where one item is centered beneath the other and the overall shape of the two items is rectangular.

Another example of a strange template item shape is the top item in the template shown in Figure 7-3. It is hard to say exactly how bad a layout based on this template will look, since the numerical dimensions are not known, but it is reasonable to assign some weight against moves that drastically reduce the width of a template item.

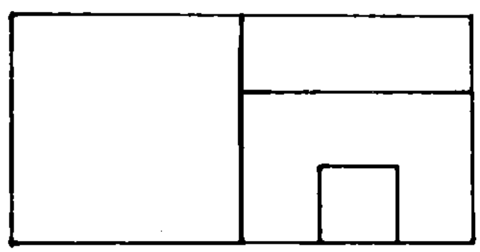


Figure 7-2. Possible Exception to the Convexity Rule.

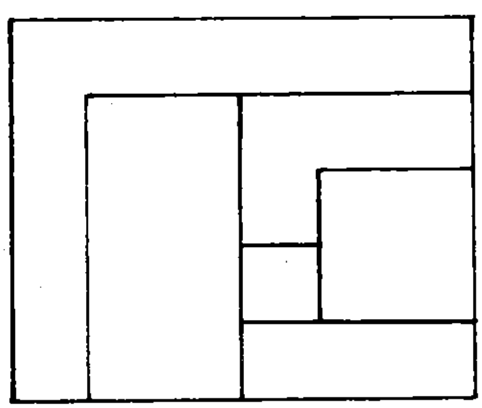


Figure 7-3. Template with Item of Unusual Shape.

7.3 Importance Ranking

The relative importances of news items can be provided to the layout program in the hope that the eventual layout will have the more important items near the top of the page, and the less important items near the bottom. Taking into account the fact that the more important items tend to be the larger, and that the breadth-(N_1,N_2) search procedure already tends to place larger items nearer the top of the page, it would be possible to let this suffice as a treatment of importance ranking. Otherwise, giving some weight to the presence of unimportant items near the top of the template is reasonable.

7.4 Headline Bumping

In a layout such as the one shown in Figure 7-4, the headlines of the two stories on the top row are said *bump,* since they are directly adjacent. Bumping headlines are undesirable visually. Such a condition can usually be detected when constructing the template, and should be assigned a weight. On

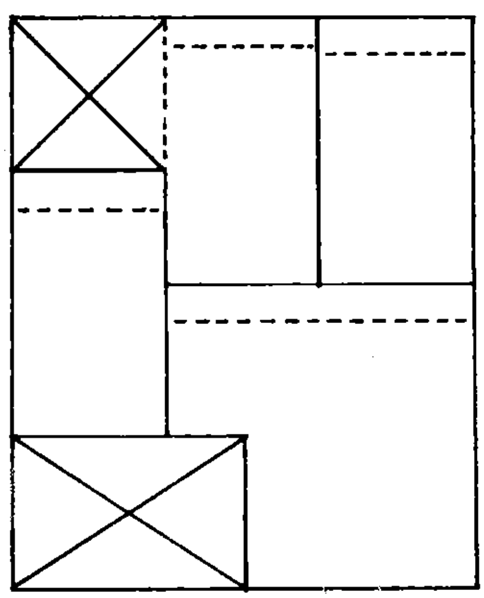


Figure 7-4. Example of Headline Bumping.

the other hand, given only a template such as:

```
A A B B B

C C C - -

- - - D D
```

it is impossible to tell what the relative heights of the tops of C and D are. If h_2 is zero or near-zero in the final layout, this will cause the headlines to bump, but this cannot be detected in the template, since it depends on numerical knowledge found only in the layout and not in the template. However, such an occurrence tends to be relatively improbable, especially with the asynchronous layout procedure discussed in Chapter 8.

7.5 Pictures

Templates with pictures that touch each other are less desirable than they would be otherwise. Cases such as the one shown in Figure 7-5 can be

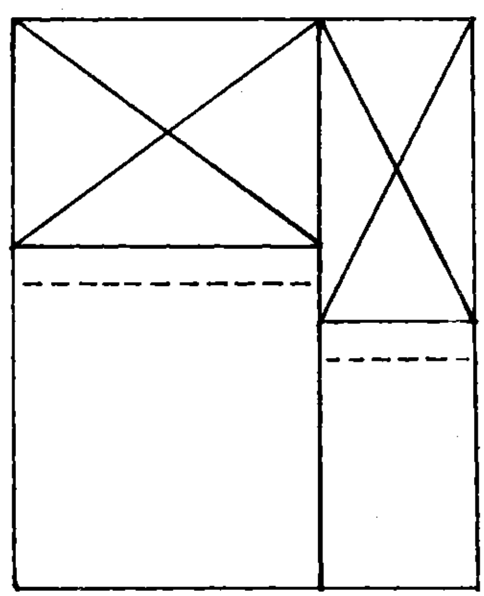


Figure 7-5. Example of Picture Bumping.

detected by the same procedures that detect headline bumping. That is, the top of each item can be considered to be either headline or picture at the left, and headline or picture at the right. A story with no dependent pictures is headline-headline, an independent picture is picture-picture, a story in Format 1 is picture-headline, and so forth. Headlines bump against headlines, and pictures bump against pictures. Other undesirable cases include such layouts as the ones shown in Figure 7-6, in which pictures are positioned adjacent to each other in other manners.

Caution should be exerted when choosing the weights associated with the violations of these rules, or of any of the esthetic rules. Since there are so many bad template features potentially associated with pictures, this can produce a tendency not to choose partial templates containing pictures, since such partial templates tend to be judged of lower quality, such lower quality being difficult to avoid. One partial solution to this problem is also to include esthetic rules against leaving pictures to be inserted last, but such a rule might

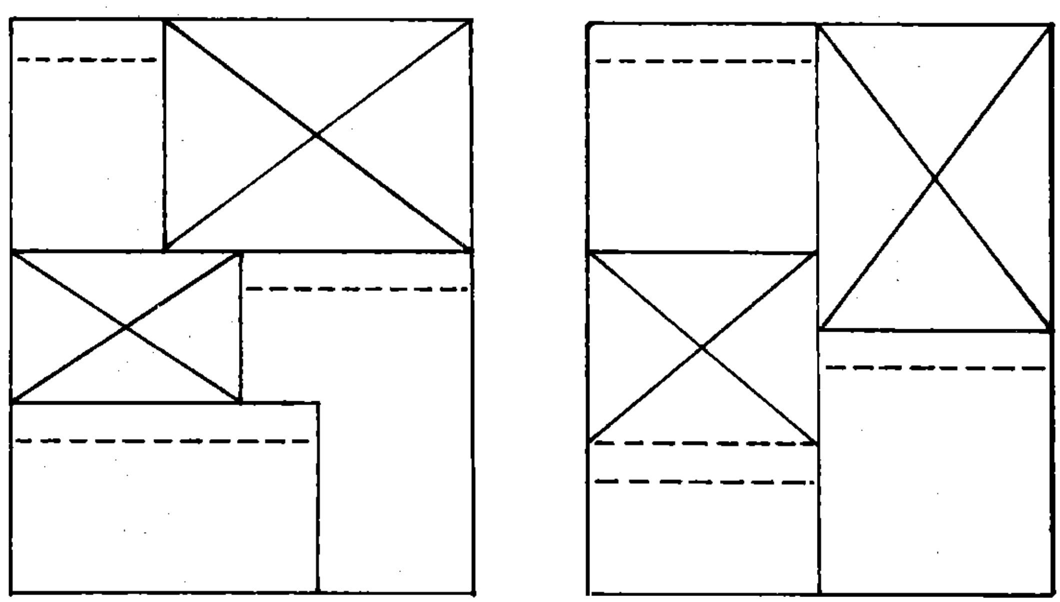


Figure 7-6. Examples of Undesirable Picture Placements.

prove difficult to control.

7.6 Summary

This section has presented a basic procedure whereby template selection may be directed toward the production of esthetically pleasing templates. The definition of such templates is embodied in a set of esthetic rules, the violations of which are assigned various weights. These rules may be broken, but only at the cost of a reduced probability that such templates will be selected.

The exact choice of rules and their associated weights is an implementation decision. Implementations to date have made this decision based solely on experience.

Chapter 8 - Asynchronous Layout

8.1 Introduction

The layout equations associated with a particular template place constraints on the set of layouts possible with the template. This chapter considers certain additional layout equations which outlaw a certain class of undesirable layouts. The additional constraints thereby incorporated, however, make the generation of layouts very difficult, since the constraints are so strong as to drastically reduce the number of valid nodes in the layout tree.

This chapter also introduces a change in the method of generating layout equations from a template, in order to make the relationship between template and layout more flexible and to allow a single template to correspond to a greater class of layouts. This change leads to an increase in the number of valid nodes in the layout tree sufficient to allow the additional layout equations mentioned above to be used.

8.2 Leading Factors

The preceding chapters have presented a method whereby a given set of news items can be laid out in a given newshole. In the resulting layout, each news item is allocated an amount of space which is greater than or equal to the area it is known to need. This allocation of space includes a distribution of the available residue among the various items; the residue present in a particular layout item will represent leading or some other form of unused space.[1] Each item in the layout will have an associated *leading factor,* which indicates the ratio between the amount of residue assigned to the item and the area of the item. For example, if an item with an area of 1000 column-points is assigned

1. Technically, only text areas can be leaded. However, analogous procedures are possible with headlines and, to an extent, pictures.

1050 column-points of space, this corresponds to 50 column-points of residue, or a 5% leading factor.

In general, we can observe that layouts with extreme discrepancies among the leading factors of the various items on the page will be less pleasing visually than layouts where the leading factors are more nearly equal. As we have seen, there are typically many possible solutions to the linear programming problem represented by a set of layout equations, one solution for each point inside the basic feasible region. Of course, all of these layouts will be based on the same template, and will therefore differ from each other only in the particular values of the height variables. Of these solutions, some will be more desirable than others, because of their particular distribution of leading factors. Certain mathematical techniques (such as coefficient ranging or quadratic programming) can be used in conjunction with linear programming to produce final layouts where the leading factors of the items are as nearly equal as possible within the constraints imposed by the layout equations.

In some cases, this will not be good enough. Even the particular solution to the layout equations that minimizes the differences among the leading factors might still represent an undesirable layout, because of unfortunate constraints imposed by the layout equations. Consider, for example, the template:

```
A A B B B C

D D D - - -

- - - E E E
```

Assume that the average leading factor on the page (i.e., the ratio between the total residue on the page and the sum of the areas of the news items) is 5%, and that news item B is equal in area to news item A. This template, however, requires that the space allocated to B be at least 50% greater than that allocated to A, since the space allocated to B is $3\ h_1 + 2\ h_2$, whereas the space allocated to A is only 2 h_1. It is possible, although improbable, that layouts could exist for such a layout problem and for such a template, but in any such layout, B's leading factor will be at least 50% (since A's leading factor is at least 0%). Any leading factor so much larger than the average leading factor on the page can be judged totally unacceptable.

8.3 Additional Layout Equations

Templates which have no layouts with appropriate leading factors can be considered to have no layouts at all. Such templates can be outlawed by the use of additional layout equations, which set maximum leading factors permissible for the various items. Thus, the only layouts generated will be layouts with acceptable leading factors.

Such an additional layout equation, for a typical item, will be of the form:

$$\text{space allocated for item} \leq \text{maximum space allowable} \tag{8-1}$$

where the right-hand side of the equation is some constant which represents the maximum amount of space which should be allowed to be allocated to the item. A reasonable value is an amount 15% greater than the maximum area possible for the item,[2] resulting in a maximum leading factor of 15%. Of course, the leading factors of most of the items in the layouts produced can be expected to be less than 15%, which is simply the maximum leading factor deemed at all acceptable.

Remembering that the area equation for the item is of the form:

$$\text{space allocated for item} \geq \text{minimum space allowable} \tag{8-2}$$

2. For items where text can be cut, or where pictures may be cropped, or where headlines may be rewritten, there is no one area associated with the item, but rather a range of values.

where the minimum space allowable is simply the minimum possible area for the item, we can consider Equation 8-1 as also being a type of area equation. The two area equations for each item both have the same symbolic left-hand side, but differ in their numerical right-hand sides. Together, these two area equations set numerical bounds on the amount of space that can be permitted to be allocated for the item.

In addition to the inclusion of the new area equations in the set of layout equations, a new remaining-items equation along the same lines must also be included in sets of partial layout equations. This remaining-items equation is of the form:

$$\text{space remaining} \leq \text{maximum space allowable} \qquad \text{(8-3)}$$

where the maximum space allowable is the sum of the maximum amounts of space allowable for each of the items remaining.

8.4 A Problem

The use of these additions to the set of layout equations will lead only to layouts with reasonable leading factors. Unfortunately, experience shows that they tend in general to make it extremely difficult to find such layouts. In fact, it was found impossible to determine experimentally any reliable figures on how much more difficult the new equations make the layout problem; these equations make layouts so rare that, for most layout problems tested, no layouts at all could be generated within a reasonable period of time. It is to be expected that the addition of restrictions on possible layouts will make layouts harder to find; what is unexpected is that these restrictions on maximum leading factors make layouts so much harder to find.

The inclusion of the additional area equations, by making the set of layout equations much more restrictive, vastly increases the number of dead ends in the layout tree, thereby reducing the occurrence of successful layouts. The approach taken in this thesis is to make certain changes in the nature of the layout equations which, by independently increasing the incidence of successful layouts, help to offset this effect. These changes are discussed below.

8.5 Definition

In almost all templates, certain items appear above certain others. For example, in the template:

```
A A A B B B

C C C - - -

- - - D D D
```

A appears above C, and B appears above D. For any template, these relationships define a *partial ordering* on the positions of the tops of the items. Obviously, if item X appears above item Y in the template, it must also do so in the layout. This partial ordering, like any other, is transitive, in that if X is above Y and Y is above Z, then X is defined to be above Z.

The layout equations as presented so far, however, define a *total ordering* on the positions of the tops of the items. For example, for the template above, even though C is not above D, C's top is constrained by the layout equations to be above (or at the same height as) D's top in the final layout. *Asynchronous layout* is a manner of generating layout equations in such a way that unnecessary relations, such as the relation between the positions of the tops of C and D, are not included in the necessary conditions stated in the layout equations; this is opposed to *synchronous layout*, presented in the previous chapters, where these relations are included as necessary conditions. With asynchronous layout, the above template can be used to generate layouts where D's top is positioned higher than C's top, as well as the reverse case (and of course the special case where they both have the same height).

The method used is a change in the nature of the layout equations. Instead of a simple sequential set of height variables h_1, h_2 and h_3 used in the synchronous case, the asynchronous layout equations are based on other types of height variables. For the template shown above, let the height variable h_{AC} represent the vertical distance between the top of A and the top of C. Similarly, let h_{BD} represent the vertical distance between the top of B and the top of D. The height variable h_{C*} represents the distance between the top of C and the bottom of the newshole, and h_{D*} represents the distance between the top of D and the bottom of the newshole. These height variables do not represent the heights of template rows as in the synchronous case; rather, they represent the heights between the tops of items and either the tops of the items directly below or the bottom of the newshole.

For this template, the asynchronous layout equations are shown in Table 8-I. The first equation is the non-negative condition on the asynchronous height variables. The next two equations are the height equations; two separate height equations are needed here. The next four equations are the area equations for the four items. It may be seen that these asynchronous layout equations embody the same constraints on the layouts produced as do

---

Table 8-I. Asynchronous Layout Equations for Example.

$$h_{AC}, h_{BD}, h_{C*}, h_{D*} \geq 0$$

$$h_{AC} + h_{C*} = \text{height of newshole}$$
$$h_{BD} + h_{D*} = \text{height of newshole}$$

$$3\ h_{AC} \geq \text{area (A)}$$
$$3\ h_{BD} \geq \text{area (B)}$$
$$3\ h_{C*} \geq \text{area (C)}$$
$$3\ h_{D*} \geq \text{area (D)}$$

the equivalent synchronous layout equations, except that they allow D's top to appear higher than C's (since h_{AC} may be less than h_{BD}, although h_1 is never less than h_1+h_2). In short, the relative heights of items in the template do not unnecessarily carry over into the layout; layout items may have any reasonable heights, which are not totally "synchronized" with the relative heights of the template items.

Consider a more complicated example. For the asynchronous template:

```
A A A B B B

C C C - - -

- - - - D D

E E E E - -
```

the asynchronous layout equations are shown in Table 8-II. The first equation states the non-negative condition on the height variables. The next two equations are the height equations. One is necessary for each item which

---

Table 8-II. Asynchronous Layout Equations for Example.

$$h_{AC}, h_{BD}, h_{BE}, h_{CE}, h_{D*}, h_{E*} \geq 0$$

$$h_{AC} + h_{CE} + h_{E*} = \text{height of newshole}$$
$$h_{BD} + h_{D*} = \text{height of newshole}$$

$$3\ h_{AC} \geq \text{area (A)}$$
$$h_{BE} + 2\ h_{BD} \geq \text{area (B)}$$
$$3\ h_{CE} \geq \text{area (C)}$$
$$2\ h_{D*} \geq \text{area (D)}$$
$$4\ h_{E*} \geq \text{area (E)}$$

$$h_{BE} \geq \text{headline height of B + increment}$$
$$h_{BD} \geq \text{headline height of B + increment}$$

$$h_{AC} + h_{CE} = h_{BE}$$

touches the bottom of the newshole.[3] The next five equations are the area equations for the five template items. The next two equations are the headline equations for B (only item B has a non-rectangular shape); since both E and D appear under B, and since their relative heights are not known, two headline equations are necessary. The last equation is a new type of equation, called a *synchronization equation;* when E is inserted into the template, it cuts across two independent sections of the template (that is, it lies beneath both B and C), necessitating a synchronization equation to state that the height of the top of E is the same in both sections. (Indeed, the height equations may themselves be considered a special form of synchronization equation, forced by the occurrence of the bottom of the newshole.)

In the asynchronous layout equations shown, no statement is made about the relative heights of the tops of C and D, or of D and E; such restrictions are considered unnecessary. On the other hand, the asynchronous layout equations do require that, for example, the top of B be higher than the top of E, since such a restriction is obviously required by the fact that B is above E in the template.

8.6 Evaluation

Synchronous templates, since they define total orderings, place restrictions on the relative heights of all of their items. They are often inappropriate for particular layout problems because of the relative sizes of the news items; in such cases, they might lead either to no layout, or to excessively leaded ones. To understand the connection between these two possibilities, consider that excessive leading is caused by the transfer, under the control of the layout equations, of an excessive amount of residue into one or more of the layout

3. If the newshole is non-rectangular, there must be a separate height equation, along with a separate height variable, for each distinct place that an item touches the bottom of the newshole.

items. If the layout equations force the transfer of more residue than is available, then there is no solution. If they force no more than is available, they can still cause excessive leading factors in one or more of the items.

The equivalent asynchronous templates are more flexible in the sets of layouts they can produce, and in the sets of item sizes they are appropriate for. Therefore, asynchronous layout is inherently more efficient than synchronous layout. To understand this, consider that each asynchronous template can correspond to a number of synchronous templates, where each synchronous template represents some permutation of the relative heights of certain items. Since the asynchronous template will be valid if any of these synchronous templates are valid, it has a greater chance of being valid than a single one of the synchronous templates. Interestingly enough, even though each asynchronous template can correspond to more than one synchronous template, the total number of asynchronous templates is the same as the total number of synchronous templates, since they have the same template grammar. This is because more than one asynchronous template can have the same set of layout equations, making such asynchronous templates equivalent. Thus, the same asynchronous template may be possible to generate in a number of ways, corresponding to alternative orderings of the placement of the template items. Even though these items are placed top-down in the template, they are not necessarily being placed top-down in the resulting layout. Although this leads to a repetition of equivalent templates in the layout tree, this causes no real problem here when probabilistic search, which considers only part of the tree in any case, is used.

Asynchronous layout owes its efficiency also to the fact that it allows fewer ways to force the transfer of residue into layout items. For example, in the synchronous template:

```
A  A  A  B  B  B

C  C  C  -  -  -

-  -  -  D  D  D
```

if item D were larger than item C, this would force residue into item C (since item C is constrained to have at least the same amount of space allocated to it as item D). On the other hand, this particular case of forced transfer does not occur in asynchronous layout.

We see, then, that asynchronous layout simplifies the search for layouts, since it increases the occurrence of valid nodes in the layout tree. The use of asynchronous layout can help to counteract the decrease in the occurrence of valid nodes caused by the use of additional layout equations to set maximum leading factors. In fact, we can expect asynchronous layout to be especially useful here, since it lowers the probability of an unallowable residue transfer which would result in a violation of the new layout equations. Experience shows that this is indeed the case, and that that a combination of asynchronous layout with the additional layout equations is effective, and that layout programs using the two techniques together are only slightly slower than those using neither. This is considered a reasonable price to pay for the assurance of appropriately leaded layouts.

There are some miscellaneous special problems caused by the use of asynchronous layout, relating to the fact that the template bears a less direct relation to the resulting layout. For example, in an asynchronous template such as:

```
A A A A A A

B B - - - -

- - C C - -

- - - - D D
```

the shape of A in the layout will not necessarily be that shown in the template, depending on the relative heights of B, C, and D. In particular, the shape of A can violate the convexity rule of Section 7.2 if C is placed higher than both B and D in the final layout. Similarly, in the asynchronous template:

```
A A B B B

C C - - -

- - D D -

E E - - -
```

where items C and D correspond to pictures, it is impossible to tell whether C and D will border on each other in the final layout. These examples show that the great computational advantages of asynchronous layout are combined with a slight decrease in the ability to make esthetic judgements on asynchronous templates; both the advantage and the disadvantage stem from the more flexible relationship between template and final layout.

8.7 Summary

Restrictions on the maximum amounts of space allocated to layout items can be expressed as additions to the set of layout equations. Such equations are practical to solve only in the case of asynchronous layout, which makes the connection between template and layout more flexible.

Chapter 9 - Examples of System Operation

9.1 The Experimental Implementation

To test the validity of the approaches to news layout described in the preceding chapters, several experimental implementations were prepared at various stages of the research. These implementations were programmed in MacLisp, a dialect of Lisp, running on a DECsystem-10 computer.

The most recent, the most complete, and the most effective of these experimental implementations uses top-down asynchronous layout, with breadth-(N_1,N_2) search of the layout tree; for the sake of experimental flexibility, this implementation allows the user to choose N_1 and N_2 dynamically at each ply. All of the layout equations described in the preceding chapters are implemented, including the additional area equations described in Chapter 8 for setting maximum leading factors. All of the particular esthetic rules listed in Chapter 7 are included. For the sake of simplicity, this implementation incorporates only story-picture formats 1, 2 and 3; therefore, it allows stories to have at most one dependent picture. This implementation was used to generate the example layouts in this chapter.

As shown below, this implementation has proven its ability to produce layouts for the problems given it, taking less time than it is estimated would be required if the layout were done by hand. This is especially impressive considering the relatively slow instruction execution rate (about 300,000 instructions per second) of the particular computer used.

9.2 Example Layouts

This chapter shows the results of applying the experimental implementation to several news layout problems taken from actual newspapers. For each of these example problems, a page from a newspaper was selected, and a layout problem prepared from it. The layout program then operated on the layout problem to produce a set of possible layouts. One layout was chosen manually, and a page was pasted up using the specified layout and news items cut from the original page.

For each story to be laid out, the maximum leading factor for text was set at 15%.[1] A set of possible headline widths and heights was generated from the headline used in the original layout; the maximum leading factor for headlines was approximately 15%, based on a formula related to the number of lines in the headline. Picture heights were constrained to be at most 10% larger or smaller than they appeared in the original layout, to simulate the range of cropping factors typically possible. It is believed that these constraints are, if anything, less flexible than those found in real newspaper layout. For example, in actual practice, headlines can easily be made larger or smaller as needed, text can be cut, pictures can be cropped more, and so forth.

9.2.1 Boston Globe Examples

This section shows layouts generated for pages taken from various daily editions of the Boston Globe.

1. That is to say, 15% was the absolute maximum leading factor that could be tolerated. Actual leading factors in the generated layout could be expected to be somewhat less.

9.2.1.1 First Example

Figure 9-1 shows a layout generated for the items on Page 3 of the Boston Globe on January 10, 1978. The sandwich (the boxed copy appearing in the text area) in the First-Blacks story was included in the text length in the layout problem. This layout was produced in 19 seconds of CPU time. In all, four layouts were produced in one pass of the layout algorithm, the first being generated after 11 seconds of CPU time and the last after 19 seconds, at which point the pass terminated. In this example, as in the others, the values of N_1 and N_2 were chosen large enough so as virtually to guarantee a layout being achieved in one pass, making the amounts of time for the various examples more nearly comparable. Of course, use of the optimal values of N_1 and N_2, if they were known, would result in a reduction in the expected time for a layout.

9.2.1.2 Second Example

Figure 9-2 shows a layout generated for the items on Page 3 of the Boston Globe on January 11, 1978. The kicker (the smaller headline appearing over the main headline) in the Councilman-Deficit story was included in the headline height in the layout problem. This layout was produced in 1 minute 53 seconds of CPU time. In all, two layouts were produced in one pass of the layout algorithm, the first being generated after 1 minute 52 seconds of CPU time; the pass required 2 minutes 7 seconds.

9.2.1.3 Third Example

Figure 9-3 shows a layout generated for the items on Page 3 of the Boston Globe on April 21, 1977. This layout was produced in 2 minutes 52 seconds of CPU time. In all, ten layouts were produced in one pass of the layout algorithm, the first being generated after 2 minutes and 40 seconds of

Turnover continues at Somerville city hall

By Kay Longcope
Globe Staff

First blacks on the block — worried but they won't run

By Ron Hutson
Globe Staff

Black families have lived on Codman Hill for years without serious incidents, a fact that has baffled both police and longtime residents of the Irish neighborhood.

City plans more BCH consolidation

By Richard A. Knox
Globe Staff

Hope for dragger all but lost after body is discovered at sea

By Alexander B. Hawes Jr
Globe Staff

Figure 9-1. Layout for First Boston Globe Example.

State auditor blasts lack of restraint in MBTA spending

Can-dating up for action soon

Mr. and Mrs. Dung Lam and latter's sister, Thu Duong (from left to right) watch the Lam children – Thi, 6, Quyen, 9, and Thu, 5 – dance in living room of their Brighton apartment. (Globe photo by Stan Grossfeld)

A family of 'boat people' in Boston

By Gloria Negri, Globe Staff

City spokesmen disagree

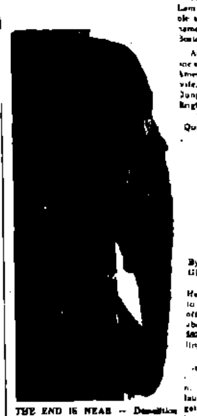
Councilman says Boston running up $15m deficit

By Peter Cowen, Globe Staff

EPA says Cape radar not a hazard

Globe Washington Bureau

Power drain slows down march of time



THE END IS NEAR – Demolition crews, ... (Globe photo by Bill Curtis)

Health council turns down Lahey clinic probe

By Richard A. Knox, Globe Staff

Figure 9-2. Layout for Second Boston Globe Example.

Beat changed through years — now it ends

By Bob Sales
Globe Staff

Student indicted in Wellesley robbery

By John F. Cullen
Globe Staff

Detectives Frank (left) and Fred McLean recount their years of police service. (Globe photo by Bill Curtis)

Gardner leaders suggest old hospital as prison site to provide needed jobs

Globe State House Bureau

CORRECTION

'Town meeting' in Watertown

Insurance protesters drown out Dukakis talk

By Nick King
Globe Staff

Gov. Dukakis addresses Watertown audience. (Globe photo by John Blanding)

They were right - it is an Indian burial ground

Board to report on Southie plans

Parents accuse police of threat to daughter

Quincy girl rejected Boston restaurant bill after saying she found hair in clam plate

By Mary Neter
Globe Staff

LORI McGLINCHEY ... taken to headquarters

Figure 9-3. Layout for Third Boston Globe Example.

CPU time; the pass required 4 minutes 2 seconds.

9.2.1.4 Fourth Example

Figure 9-4 shows a layout generated for the items on Page 3 of the Boston Globe on March 10, 1977. The format for the '73-Murders story was manually changed from Format 1, produced by the layout program, to the format shown for esthetic reasons. This layout was produced in 5 seconds of CPU time. In all, four layouts were produced in one pass of the layout algorithm, of which this was the first; the pass required 13 seconds.

9.2.1.5 Fifth Example

Figure 9-5 shows a layout generated for the items on Page 3 of the Boston Globe on May 14, 1977. This layout was produced in 2 minutes 20 seconds of CPU time. In all, four layouts were produced in one pass of the layout algorithm, of which this was the first; the pass required 2 minutes 28 seconds.

9.2.1.6 Sixth Example

Figure 9-6 shows a layout generated for the items on Page 3 of the Boston Globe on January 6, 1978. In an actual layout, lines would be inserted to better isolate the Patriarca picture from the story beneath it. This layout was produced in 1 minute 52 seconds of CPU time. In all, two layouts were produced in one pass of the layout algorithm, of which this was the first; the pass required 2 minutes 7 seconds.

Figure 9-4. Layout for Fourth Boston Globe Example.

Timilty slated for 2d US job

President Carter will nominate Sen. Joseph F. Timilty of Mattapan to head the National Commission on Neighborhoods. The announcement in Washington yesterday will give Timilty his second major Administration assignment from Carter.

Timilty has been serving on a committee established by Carter earlier this year to screen candidates and recommend a successor to FBI Director Clarence M. Kelley. The senator also ran Carter's presidential campaign in Pennsylvania last year.

The National Commission on Neighborhoods was created by the National Neighborhood Policy Act signed by Carter April 30 to study neighborhood problems and propose national strategy to revitalize the economy of urban areas.

The commission has been authorized to hold public hearings in cities around the country. Boston is expected to be one of the commission's major areas of activity because of the complexity of neighborhood problems here and the need for Federal assistance.

Timilty's nomination is expected to be submitted to the Senate for confirmation in early June.

The 20-member commission, whose members are nominated by the President and the leaders of Congress, will be given $1 million to make its one-year study, which will be submitted to Congress. Timilty and other public members of the panel will receive $100 a day plus travel expenses for their work.

COMING MONDAY

Should fathers earn welfare?

Should welfare fathers be required to work part-time to earn their welfare checks? Massachusetts Gov. Michael Dukakis thinks so, but his 'workfare' proposal has provoked torrents of criticism from people in and outside his administration. Monday's FORUM examines the pros and cons of 'workfare' and invites readers to send their opinions. Look for FORUM in its new place on the letters page.

THE PICKIN'S ARE PLENTIFUL – [illegible] on Boston's Fish Pier. (Globe photo by Bill Curtis)

Powers battles courthouse plan

Ex-Navy brigs as Hub jail urged

The Suffolk County Courthouse Commission recommended yesterday that US District Court Judge W. Arthur Garrity Jr. look into the use of former Navy brigs in Charlestown and South Boston as prisons for inmates of the Charles Street Jail.

In a five-page letter delivered to Garrity, commission chairman John E. Powers said "it would be a mistake" for the judge to insist that the 82 cells in the Suffolk County Courthouse subbasement be renovated at cost of $350,000 to house inmates who will be transferred when the antiquated Charles Street Jail closes Nov. 1 in accordance with Garrity's order.

Powers said the now unused brigs at the Fargo Building, South Boston, and the Charlestown Navy Yard could provide more than 300 cells. Boston, he said, now owns that Charlestown facility which was closed down by the Navy.

The chairman of the three member commission which has control of the courthouse said his letter listed more than 30 reasons why the subterranean city prison should not be converted to a jail. Powers declined to disclose these reasons until Garrity has an opportunity to study the letter.

Earlier this week, Powers said the courthouse commission would refuse to allow use of the cells because of insufficient light, ventilation, security and food service facilities. He said the prison, located two stories below the courthouse, provides no recreation area, has open plumbing and limited sanitary facilities.

The cells were installed almost 40 years ago when the court building was constructed. The area has been unused for about 10 years except for temporary detention of defendants awaiting court trials.

"Walpole State Prison is a country club compared to that hole," Powers said referring to the courthouse cells.

In writing to Judge Garrity, Powers said he was acting not only for the courthouse commission, but also "as agent for the Supreme Judicial Court."

Hair transplants reaching new height

By Herbert Black
Globe Staff

Transplanting hair from the fringes to the top of a bald or balding head has developed into an art form in which the doctor designs hair contours to strengthen good features, hide bad ones and may even suggest a beard or mustache to enhance the new graft.

Some candidates for a transplant may think they are buying a work of art because a transplant can run 200 grafts or more at about $11.50 a graft — or $2300 and up (each graft contains about 12 or 14 strands of hair). Some very bald individuals with fine fringe hair may require as many as 600 grafts.

Hair transplantation has reached new heights of sophistication and esthetics thanks to power-driven instruments that obtain and implant grafts quickly with minimal discomfort. Also, the skill developed by specialists during the past decade is important, according to Dr. Richard Stough of Hot Springs, Ark., in Boston this week to attend a convention of the American Academy of Facial Plastic and Reconstructive Surgery.

Stough, who is a specialist in hair transplants, said in an interview that his field is one of the fastest-growing areas of plastic surgery. He attributes a 20 percent gain in one year in part to acceptance of transplants as an economic and psychological asset in a youth oriented America.

"It is no longer considered just pandering to vanity," he said.

Wouldn't a hair piece be cheaper?

Not for a younger person, said Stough, an associate professor at the University of Arkansas Medical School. "In 10 years an implant would pay for itself when you consider a person needs two wigs at perhaps $400 each, has to have them cleaned periodically, groomed and eventually replaced," he said.

Stough, who is presenting an update on hair transplantation to 500 plastic surgeons today, said that the esthetic phase of hair transplantation involves "emphasis on frontal hairline configuration, placement of the hair-bearing grafts, implantation of blending hairs and discriminatory selection of styling and grooming aids."

He said there probably are 200 plastic surgeons in the nation who do hair transplants, but only about a dozen that do it as their only specialty, becoming the artists of the procedure.

In his lecture, Stough points out that symmetry and balance are the fundamentals of good appearance. Sometimes, he added, the natural hairlines will accentuate protruding ears, far-apart eyes or an asymmetrical nose. Restyling hair by grafting — and growing a beard or mustache — can improve on nature, he said.

Grafting may require six or eight sessions with the doctor, each session lasting one to three hours. Each graft is about the size of a match head.

Mass. eyes workfare in Utah

Three state officials are going to Utah to check out a mandatory work program for welfare recipients there that the Bay State officials hope can be a model for a similar program back home.

The trip is being made in compliance with Gov. Michael S. Dukakis's directive that a "workfare" program be implemented in Massachusetts.

The three officials, who are scheduled to be in Utah for two or three days next week, are Richard Anderson, workfare coordinator for Dukakis; Richard T. Sullivan, supervisor of the state WIN (work incentive) program in the Division of Employment Services, and Stephen W. Kane of the Welfare Dept.

They will be researching the Utah workfare program to see if it can be used as a model for a workfare program for unemployed fathers here.

Making welfare recipients work for their welfare payment is against the regulations of the Department of Health, Education and Welfare. However, the Utah program was approved without a waiver because it includes a training element in the mandatory work program.

While Laura Vail watches, her son, Bruce, 7, and a neighborhood friend, Scott [illegible], 10, cook atop a camp stove in [illegible] of Sudbury home. (AP photo)

Edison nears end of storm cleanup

Power was finally restored last night to most homes that have been without electricity since last Monday's storm caused a massive blackout affecting more than 500,000 customers in eastern Massachusetts and Rhode Island.

A spokesman for the Boston Edison Co. said power was expected to be restored by last night to approximately 1000 homes in the Framingham area and in Burlington, Lexington and Winchester.

He said that tree-trimming crews, many of them from Con Edison in New York, will continue to cut branches near power lines throughout the weekend.

A day of grace for sticker-less

If you haven't gotten around yet to getting that spring safety inspection sticker, don't panic. Although the deadline is May 15, that day, however, is a Sunday.

The Massachusetts Registry of Motor Vehicles said it is allowing motorists a day of grace, and so will not start enforcement on the non-stickered cars until Tuesday, May 17, according to Ernest D. Sabatino, supervisor of vehicle inspection.

IT'S STILL WINTER – [illegible] peak of New Hampshire's Mt. Washington. (Globe photo by [illegible])

Figure 9-5. Layout for Fifth Boston Globe Example.

Witness alleges harm to patient

Somerville mayor names brother to job

Revival plans follow fires

Hope in Fenway disaster area

BRA's Plaza expert leaving

Patriarca, 69, hasn't retired

Murder charge dismissed for retarded man

Motorists can keep rebates, court rules

Figure 9-6. Layout for Sixth Boston Globe Example.

9.2.2 Tech Talk Examples

This section shows layouts generated for Page 8 of the February 9, 1977 edition of Tech Talk, an M.I.T. administrative newspaper. The layout originally used by Tech Talk is shown in Figure I-1 in Appendix I. The same news items, as defined by Table 5-I in Chapter 5, were laid out in a number of different ways to illustrate the variety of layouts possible.

Figures 9-7 and 9-8 show two different layouts for the news items. These layouts were produced in separate passes of the layout program. Figures 9-9 and 9-10 show examples of layouts in non-rectangular newsholes.

9.3 Summary

This chapter has presented examples of the class of layouts possible with the layout algorithm presented in this thesis. The quality of the layouts produced can be seen to compare favorably with the quality of layouts on inside pages of typical newspapers. This is particularly impressive in light of the fact that the numbers of items in many of the layout problems solved were significantly greater then the number of items on a typical newspaper page.

Katie Earnshaw Mixes MIT Duties with Dancing Future

By KATHARINE S.C. JONES
Staff Writer

Katherine Earnshaw is a professional dancer, enrolled in a master's degree program at Connecticut College, and an administrative assistant in the MIT Department of Nuclear Engineering.

How does she manage to do so much? She attributes her success at combining the demands of a dancer's career with a full-time job to the flexibility of MIT and the support of her husband and her supervisor. She is married to William Earnshaw, a doctoral degree candidate in biology at MIT.

Ms. Earnshaw began working at MIT in 1977 as a clerk in nuclear engineering headquarters. Three evenings a week she went to modern dance classes. When she learned of an opportunity to study with dancer choreographer Dorothy Hershkowitz in the fall of 1975 she thought she would have to quit her job and work part-time elsewhere. Her supervisor, John Cochrane, administrative officer in the Department of Nuclear Engineering, suggested they rearrange her schedule so she would have time for daytime classes. Her schedule was adjusted and now she has a class four days a week with Ms. Hershkowitz plus one ballet class with the Boston Repertory Ballet.

Ms. Earnshaw estimates that she devotes nine to ten hours each week to dance—more when she's rehearsing for a public performance. She spends 35 hours a week working at MIT.

"I try to be sensitive to the pressures of my job," she said. "I'd cancel a dance class if something urgent came up." Although her schedule is hectic, "I'm much happier than I was two years ago," she said.

Her modern dance classes in the fall of 1976 were repertory classes, and resulted in the premiere of "Monday Morning Quarterback: An American Dance Masque 1976," a new work by Ms. Hershkowitz performed by Dance Circle on January 6-8 at the First Congregational Church in Cambridge. The dance was well received and highly acclaimed in the Boston press. It is "the best work yet by this talented choreographer," wrote Laura Shapiro of the Boston Globe. Iris M. Fanger of the Christian Science Monitor called it a provocative work which deserves to be seen again by a wider audience.

This request has been granted. "Monday Morning Quarterback" will be performed Friday through Sunday, February 11-13, at the Joy of Movement Center, 23 Main St., in Watertown. Tickets cost $3 and are available at the Joy of Movement Centers in Cambridge and Watertown.

In addition to performances in this work, Ms. Earnshaw is working on a new piece with Christy Nichols, dance teacher at Pine Manor Junior College, to be performed March 25 and 26.

Ms. Earnshaw will receive her master's degree in dance at Connecticut College in the summer of 1978. The program requires two summers and one academic year of training and study. She has completed two six-week summer sessions, and hopes to return for a full academic year in the near future. For each of the two summers she studied at Connecticut, she took a leave of absence from her job, and, when necessary, a temporary employee was hired to handle day-to-day office work.

Her ability to combine study and performance in dance with a full-time job has led Ms. Earnshaw to believe that "there are ways of working within the system at MIT to accomplish your career goals, and it's important for people to know that they can work something out."

DANCING A CHEERLEADERS' ROUTINE in "Monday Morning Quarterback" are members of the Dance Circle ensemble. That's Katherine Earnshaw on the far right. photo by George Simian

16 Named in Engineering School

The following appointments or changes of appointments have been announced in the School of Engineering.

Department of Electrical Engineering and Computer Science:

Dr. Cyril Leung, appointed as assistant professor until July 1, 1979. Dr. Leung, who received the B.Sc. (Eng.) Honours degree from Imperial College, London, completed his doctorate at Stanford University, where he was the 1975 King George VI memorial fellow, in 1976. His research interests are in the areas of communication, information and complexity theories.

Mr. Markus Zahn, appointed visiting associate professor Sept. 1, 1977. Dr. Zahn, who as an instructor of electrical engineering at MIT from July 1969 to July 1970, is associate professor in the Department of Electrical Engineering at the University of Florida, Gainsville.

Department of Chemical Engineering:

Dr. William M. Deen, appointed assistant professor in the department and in the Harvard-MIT Program in Health Sciences and Technology until July 1, 1979. Dr. Deen, who received the BS degree in chemical engineering from Columbia University in 1969, and the MS and PhD degrees in chemical engineering from Stanford University in 1971 and 1973, was adjunct assistant professor in the departments of medicine and physiology at the University of California, San Francisco, from January 1975 until June 1976.

Dr. John P. Longwell, change of appointment to professor in the Department of Chemical Engineering from visiting professor, beginning Jan. 1, 1977. Dr. Longwell, formerly senior scientific adviser in corporate and government research at Exxon Research and Engineering Co., is a past president of the Combustion Institute and a member of the National Academy of Engineering. He received the BS degree in mechanical engineering from the University of California in 1940 and the ScD in chemical engineering from MIT in 1943.

Dr. Frederick A. Putnam, appointed assistant professor in the Department of Chemical Engineering until July 1, 1979. Dr. Putnam received the BA in physics, cum laude, from Dartmouth College in 1969, the MS in chemical engineering from Case Western Reserve University in 1975 and the PhD in chemical engineering from Carnegie-Mellon University in 1976. His research areas of interest are interfacial phenomena, physical adsorptions of gases, adhesion and abhesion, solid surface characterization and thermodynamics and statistical mechanics.

Dr. Preetinder S. Virk, appointed associate professor in the Department of Chemical Engineering until July 1, 1979. Dr. Virk, who received the Doctor of Science degree from MIT in 1966, was an assistant professor of chemical engineering at MIT from 1966-1971 and an associate professor from 1971-72. From 1972 until he returned to MIT he was a professor of chemical engineering at the Indian Institute of Technology in Madras, India. He is a consultant for the Process Industries Group at Stone & Webster Engineering Corp.

Department of Civil Engineering:

Dr. Oral Buyukozturk, appointed associate professor in civil engineering until July 1, 1979. Dr. Buyukozturk received the MS in civil engineering in 1963 from Technical University of Istanbul, Turkey, the MS in structural engineering from Cornell University in 1969 and the PhD in structural engineering from Cornell in 1970. Dr. Buyukozturk formerly was adjunct associate professor of engineering at Brown University.

Department of Mechanical Engineering:

Dr. Adam C. Bell, change of appointment to associate professor from visiting professor until July 1, 1979. Dr. Bell, who received the BEng degree in 1963 from Nova Scotia Technical College, holds the SM, ME and ScD degrees from MIT. He became visiting associate professor in mechanical engineering at MIT in September 1975 after being on the engineering and applied sciences faculty at the State University of New York at Buffalo from September 1968—September 1975.

Dr. Peter W. Huber, change of appointment to assistant professor in mechanical engineering from lecturer until July 1, 1978. Dr. Huber, who received the SB and SM degrees in 1974 and the PhD in 1976, all from MIT, has been a research assistant and part-time teaching assistant at the Institute since 1972.

Dr. Alician V. Quinlan, appointment as Arthur D. Little Assistant Professor of Environmental Sciences and Engineering. Dr. Quinlan, who also is a Rockefeller Foundation Fellow in Environmental Affairs and an assistant professor of mechanical engineering, received the SB in life sciences in 1968 from MIT, the MS in physical oceanography in 1970 from the Institute of Marine Science, University of Alaska, and the PhD in environmental engineering in 1975 from MIT. Her main fields of interest are environmental engineering, element-cycle dynamics in aquatic and terrestrial ecosystems and reaction-transport dynamics in living systems.

Dr. Derek Rowell, change of appointment to assistant professor in mechanical engineering from Sponsored Research Staff and lecturer in mechanical engineering until July 1, 1979. Dr. Rowell received the BE degree in 1966 and the PhD degree in 1970, both from the University of Canterbury, Christchurch, New Zealand. He is director of the Sensory Aids Evaluation and Development Center at MIT.

Dr. Hendrik G. Stassen, appointment as visiting associate professor in mechanical engineering until Aug. 1, 1977. Dr. Stassen, associate professor in control engineering at Delft University of Technology, The Netherlands, since 1968, specializes in control engineering, man-machine systems and rehabilitation of the handicapped. He received the BSc degree in 1960, the MSc degree in 1964 and the Dr. of Technology degree in 1968, all from the Delft University of Technology.

Dr. Nak-Ho-Sung, change of appointment to assistant professor of mechanical engineering from research associate and lecturer until July 1, 1978. Dr. Sung received the BS in chemical engineering from Seoul National University, Korea, in 1964, the MS in physical chemistry from the University of Chicago in 1967 and the ScD in mechanical engineering from MIT in 1972. He joined MIT in 1967 as a research assistant in the Fibers and Polymers Division of the mechanical engineering department.

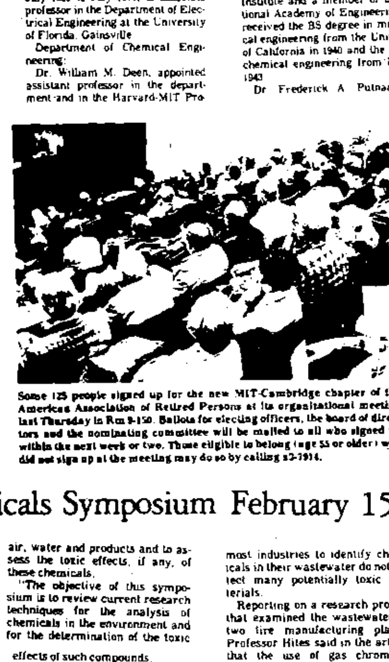
Department of Nuclear Engineering:

Dr. Mujid S. Kazimi, appointment as assistant professor in the Department of Nuclear Engineering until July 1, 1979. Dr. Kazimi received the B.S. in nuclear engineering from the University of Alexandria in 1969, the MS in nuclear engineering in 1971 from MIT and the PhD in nuclear engineering in 1973, also from MIT. From May 1974 until coming to MIT he was associate nuclear engineer, Department of Applied science, Brookhaven National Laboratory. His main fields of interest are nuclear reactor safety analysis, nuclear reactor design and optimization heat transfer and fluid mechanics in two-phase media and scientific advancement and technology transfer in developing countries.

Dr. Puthenveetil K. John, appointed as visiting associate professor part-time, in the Department of Nuclear Engineering, until July 1, 1977. An associate professor of physics at the University of Western Ontario since 1972, Dr. John received the BSc degree from Kerala University, India, in 1961, the MSc from Madras University in 1966 and the PhD from Ohio State University in 1963.

Department of Ocean Enineering:

Cmmdr. Clark Graham, change of appointment to adjunct professor from associate professor of ocean engineering until Sept. 1, 1979. Dr.



Some 125 people signed up for the new MIT-Cambridge chapter of the American Association of Retired Persons at its organizational meeting last Thursday in Rm 9-150. Ballots for electing officers, the board of directors and the nominating committee will be mailed to all who signed up within the next week or two. Those eligible to belong (age 55 or older) who did not sign up at the meeting may do so by calling x3-1914.

Toxic Chemicals Symposium February 15

A symposium on the analysis and assessment of toxic chemicals in the environment will be held at MIT Feb. 15.

The chairman of the symposium—which will begin at 9am in Kresge Auditorium—is Professor Ronald A. Hites of the MIT Department of Chemical Engineering. Professor Hites has been involved in major research projects on the application of analytical chemical techniques to the solution of environmental and energy-related problems.

"Concern about toxic chemicals in the environment has never been higher," Dr. Hites said.

"With increasing frequency, industrial, governmental and academic laboratories are being called upon to identify and quantitate potentially toxic chemicals in air, water and products and to assess the toxic effects, if any, of these chemicals.

"The objective of this symposium is to review current research techniques for the analysis of chemicals in the environment and for the determination of the toxic effects of such compounds.

"Most of the emphasis will be on organic compounds in water and their measurement by gas chromatographic mass spectrometry. The use of cellular toxicology and industrial epidemiology to assess toxic effects of chemicals also will be reviewed."

Professor Hites, in an article published in a recent issue of *Analytical Chemistry*, said the methods currently employed by most industries to identify chemicals in their wastewater do not detect many potentially toxic materials.

Reporting on a research project that examined the wastewater at two tire manufacturing plants, Professor Hites said in the article that the use of gas chromatographic mass spectrometry detected potentially toxic chemicals several miles from where one plant's wastewater entered a river.

The symposium is sponsored by the National Science Foundation's Division of Advanced Environmental Research and Technology.

Figure 9-7. First Tech Talk Example.

DANCING A CHEERLEADERS' ROUTINE in "Monday Morning Quarterback" are members of the Dance Circle ensemble. That's Katherine Earnshaw on the far right.

photo by George Suman

Katie Earnshaw Mixes MIT, Dancing

By KATHARINE S.C. JONES
Staff Writer

Katherine Earnshaw is a professional dancer, enrolled in a master's degree program at Connecticut College, and an administrative assistant in the MIT Department of Nuclear Engineering.

How does she manage to do so much? She attributes her success at combining the demands of a dancer's career with a full-time job to the flexibility of MIT and the support of her husband and her supervisor. She is married to William Earnshaw, a doctoral degree candidate in biology at MIT.

Ms Earnshaw began working at MIT in 1973 as a clerk in nuclear engineering headquarters. Three evenings a week she went to modern dance classes. When she learned of an opportunity to study with dancer/choreographer Dorothy Hershkowitz in the fall of 1975 she thought she would have to quit her job and work part-time elsewhere. Her supervisor John Cochrane, administrative officer in the Department of Nuclear Engineering, suggested they rearrange her schedule so she would have time for daytime classes. Her schedule was adjusted and now she has a class four days a week with Ms. Hershkowitz plus one ballet class with the Boston Repertory Ballet.

Ms Earnshaw estimates that she devotes nine to ten hours each week to dance—more when she's rehearsing for a public performance. She spends 35 hours a week working at MIT.

"I try to be sensitive to the pressures of my job," she said. "I'd cancel a dance class if something urgent came up." Although her shedule is hectic, "I'm much happier than I was two years ago," she said.

Her modern dance classes in the fall of 1976 were repertory classes, and resulted in the premiere of "Monday Morning Quarterback (An American Dance Masque 1976)," a new work by Ms. Hershkowitz choreographer," wrote Laura Shapiro of the Boston Globe. Iris M Fanger of the *Christian Science Monitor* called it a "provocative work which deserves to be seen again by a wider audience."

This request has been granted. "Monday Morning Quarterback" will be performed Friday through Sunday, February 11-13, at the Joy of Movement Center (23 Main St.) in Watertown. Tickets cost $3 and are available at the Joy of Movement Centers in Cambridge and Watertown.

In addition to performances in this work, Ms. Earnshaw is working on a new piece with Christy Nichols, dance teacher at Pine Manor Junior College, to be performed March 25 and 26.

She began her dance career rather late. Her first dance lesson was at Colby College where she earned the BA degree in sociology in 1972. Colby did not have a dance program, but she took all the courses offered in dance history, composition, and improvisation.

"Dancing was something I always wanted to do but it was always at the back of my mind," she said. "Having started so late, I'm now trying to make up for lost time."

Ms. Earnshaw began working toward the master's degree in dance at Connecticut College in the summer of 1975. The program requires two summers and one academic year of training and study. She has completed two six-week summer sessions, and hopes to return for a full academic year in the near future. For each of the two summers she studied at Connecticut, she took a leave of absence from her job, and, when necessary, a temporary employee was hired to handle day-to-day office work.

Her ability to combine study and performance in dance with a full-time job has led Ms. Earnshaw to believe that "there are ways of working within the system at MIT to accomplish your career goals, and it's important for people to know that they can work something out."

Toxic Chemicals Symposium Planned For February 15

A symposium on the analysis and assessment of toxic chemicals in the environment will be held at MIT Feb. 15.

The chairman of the symposium—which will begin at 9am in Kresge Auditorium—is Professor Ronald A. Hites of the MIT Department of Chemical Engineering. Professor Hites has been involved in major research projects on the application of analytical chemical techniques to the solution of environmental and energy-related problems.

"Concern about toxic chemicals in the environment has never been higher," Dr. Hites said.

"With increasing frequency, industrial, governmental and academic laboratories are being called upon to identify and quantitate potentially toxic chemicals in air, water and products and to assess the toxic effects, if any, of these chemicals.

"The objective of this symposium is to review current research techniques for the analysis of chemicals in the environment and for the determination of the toxic effects of such compounds.

"Most of the emphasis will be on organic compounds in water and their measurement by gas chromatographic mass spectrometry. The use of cellular toxicology and industrial epidemiology to assess toxic effects of chemicals also will be reviewed."

Professor Hites, in an article published in a recent issue of *Analytical Chemistry*, said the methods currently employed by most industries to identify chemicals in their wastewater do not detect many potentially toxic materials.

Reporting on a research project that examined the wastewater at two tire manufacturing plants, Professor Hites said in the article that the use of gas chromatographic mass spectrometry detected potentially toxic chemicals several miles from where one plant's wastewater entered a river.

The symposium is sponsored by the National Science Foundation's Division of Advanced Environmental Research and Technology.

16 Named in Engineering School

The Following appointments or changes of appointments have been announced in the School of Engineering.

Department of Electrical Engineering and Computer Science:

Dr. Cyril Leung, appointed as assistant professor until July 1, 1979. Dr Leung, who received the B.Sc. (Eng.) Honours degree from Imperial College, London, completed his doctorate at Stanford University, where he was the 1975 King George VI memorial fellow, in 1976. His research interests are in the areas of communication, information, and complexity theories.

Mr Markus Zahn, appointed visiting associate professor Sept. 1, 1977. Dr. Zahn, who as an instructor of electrical engineering at MIT from July 1969 to July 1970, is associate professor in the Department of Electrical Engineering at the University of Florida, Gainesville.

Department of Chemical Engineering:

Dr. William M. Deen, appointed assistant professor in the department and in the Harvard-MIT Program in Health Sciences and Technology until July 1, 1979. Dr. Deen, who received the BS degree in chemical engineering from Columbia University in 1969, and the MS and PhD degrees in chemical engineering from Stanford University in 1971 and 1973, was adjunct assistant professor in the departments of medicine and physiology at the University of California, San Francisco, from January 1975 until June 1976.

Dr. John P. Longwell, change of appointment to professor in the Department of Chemical Engineering from visiting professor, beginning Jan. 1, 1977. Dr. Longwell, formerly senior scientific adviser in corporate and government research at Exxon Research and Engineering Co., is a past president of the Combustion Institute and a member of the National Academy of Engineering. He received the BS degree in mechanical engineering from the University of California in 1940 and the ScD in chemical engineering from MIT in 1943.

Dr. Frederick A. Putnam, appointed assistant professor in the Department of Chemical Engineering until July 1, 1979. Dr. Putnam received the BA in physics, cum laude, from Dartmouth College in 1968, the MS in chemical engineering from Case Western Reserve University in 1975 and the PhD in chemical engineering from Carnegie-Mellon University in 1976. His research areas of interest are interfacial phenomena, physical adsorptions of gases, adhesion and adhesion, solid surface characterization and thermodynamics and statistical mechanics.

Dr. Preetinder S. Virk, appointed associate professor in the Department of Chemical Engineering until July 1, 1979. Dr. Virk, who received the Doctor of Science degree from MIT in 1966, was an assistant professor of chemical engineering at MIT from 1966-1971 and an associate professor from 1971-72. From 1972 until he returned to MIT he was a professor of chemical engineering at the Indian Institute of Technology in Madras, India. He is a consultant for the Process Industries Group at Stone & Webster Engineering Corp.

Department of Civil Engineering:

Dr. Oral Buyukozturk, appointed associate professor in civil engineering until July 1, 1979. Dr. Buyukozturk received the MS in civil engineering in 1963 from Technical University of Istanbul, Turkey, the MS in structural engineering from Cornell University in 1969 and the PhD in structural engineering from Cornell in 1970. Dr. Buyukozturk formerly was adjunct associate professor of engineering at Brown University.

Department of Mechanical Engineering:

Dr. Adam C. Bell, change of appointment to associate professor from visiting professor until July 1, 1979. Dr. Bell, who received the BEng degree in 1963 from Nova Scotia Technical College, holds the SM, ME and ScD degrees from MIT. He became visiting associate professor in mechancial engineering at MIT in September 1975 after being on the engineering and applied sciences faculty at the State University of New York at Buffalo from September 1968—September 1975.

Dr. Peter W. Huber, change of appointment to assistant professor in mechanical engineering from lecturer until July 1, 1978. Dr. Huber, who received the SB and SM degrees in 1974 and the PhD in 1976, all from MIT, has been a research assistant and part-time teaching assistant at the Institute since 1972.

Dr. Alician V. Quinlan, appointment as Arthur D. Little Assistant Professor of Environmental Sciences and Engineering. Dr. Quinlan, who also is a Rockefeller Foundation Fellow in Environmental Affairs and an assistant professor of mechanical engineering, received the SB in life sciences in 1968 from MIT, the MS in physical oceanography in 1970 from the Institute of Marine Science, University of Alaska, and the PhD in environmental engineering in 1975 from MIT. Her main fields of interest are environmental engineering, element-cycle dynamics in aquatic and terrestrial ecosystems and reaction-transport dynamics in living systems.

Dr. Derek Rowell, change of appointment to assistant professor in mechanical engineering from Sponsored Research Staff and lecturer in mechanical engineering until July 1, 1979. Dr. Rowell received the BE degree in 1966 and the PhD degree in 1970, both from the University of Canterbury, Christchurch, New Zealand. He is director of the Sensory ment as visiting associate professor in mechanical engineering until Aug. 1, 1977. Dr. Stassen, associate professor in control engineering at Delft University of Technology, The Netherlands, since 1968, specializes in control engineering, man-machine systems and rehabilitation of the handicapped. He received the BSc degree in 1960, the MSc degree in 1964 and the Dr. of Technology degree in 1968, all from the Delft University of Technology.

Dr. Nak-Ho-Sung, change of appointment to assistant professor of mechanical engineering from research associate and lecturer until July 1, 1978. Dr. Sung received the BS in chemical engineering from Seoul National University, Korea, in 1964, the MS in physical chemistry from the University of Chicago in 1967 and the ScD in mechanical engineering from MIT in 1972. He joined MIT in 1967 as a research assistant in the Fibers and Polymers Division of the mechanical engineering department.

Department of Nuclear Engineering:

Dr. Mujid S. Kazimi, appointment as assistant professor in the Department of Nuclear Engineering until July 1, 1978. Dr. Kazimi received the B.S. in nuclear engineering from the University of Alexandria in 1969, the MS in nuclear engineering in 1971 from MIT and the PhD in nuclear engineering in 1973, also from MIT. From May 1974 until coming to MIT he was associate nuclear engineer, Department of Applied science, Brookhaven National Laboratory. His main fields of interest are nuclear reactor safety analysis, nuclear reactor design and optimization heat transfer and fluid mechanics in two-phase media and scientific advancement and technology transfer in developing countries.

Dr. Puthenveetil K. John, appointed as visiting associate professor part-time, in the Department of Nuclear Engineering, until July 1, 1977. An associate professor of physics at the University of Western Ontario since 1972, Dr. John received the BSc degree from Kerala University, India, in 1951, the MSc from Madras University in 1955 and the PhD from Ohio State University in 1963.

Department of Ocean Enineering:

Cmmdr. Clark Graham, change of appointment to adjunct professor from associate professor of ocean engineering until Sept. 1, 1979. Dr. Graham received the BS degree from the U.S. Naval Academy in 1954, the SM in mechanical engineering from MIT in 1967 an the PhD in mechanical engineering in 1969, also from MIT. His principal fields of interest are high-performance marine vehicles, marine engineering, marine power systems and heat transfer.

Some 125 people signed up for the new MIT-Cambridge chapter of the American Association of Retired Persons at its organizational meeting last Thursday in Rm 9-150. Ballots for electing officers, the board of directors and the nominating committee will be mailed to all who signed up within the next week or two. Those eligible to belong (age 55 or older) who did not sign up at the meeting may do so by calling x3-7814.

Figure 9-8. Second Tech Talk Example.

16 Named in Engineering School

Katie Earnshaw Mixes MIT, Dancing

Toxic Chemicals Symposium Planned For February 15

Figure 9-9. Third Tech Talk Example.

Katie Earnshaw Mixes MIT, Dancing

16 Named in Engineering School

Toxic Chemicals Symposium February 15

Figure 9-10. Fourth Tech Talk Example.

Chapter 10 - Implementation Considerations

If the algorithms presented in this report were to be used in an operating newspaper production environment, they would probably be implemented on a relatively small computer system, such as a minicomputer system, for reasons of cost-effectiveness. There are special problems for such an implementation.

One problem is limited *memory size*. Many minicomputers have stringent inherent limitations on the possible size of memory. Even when this is not the case, the cost of the memory needed is an incentive to develop a space-efficient implementation. Another problem is limited *instruction set*. For example, most minicomputers do not include a standard floating-point package; this lack presents problems in the implementation of linear programming. *Speed of execution* is another possible problem, depending on the exact minicomputer to be used. This chapter examines each of these problems, and considers the impact of each on software system development.

10.1 Memory Size

As was mentioned in the previous chapter, the experimental implementation made for the purpose of testing the algorithms in this thesis was programmed on a large-scale computer. Because of the relative absence of memory constraints, this implementation was able to be very inefficient in its use of memory. For certain large layout problems, the data structures generated at intermediate points required almost 100,000 36-bit words for their storage. This amount can easily be decreased.

This section presents the major data bases used by the layout algorithms, and considers how each may be implemented efficiently. Data storage is expected to be the major factor in memory requirements for an implementation, with program storage being less significant.

10.1.1 Templates

In the experimental implementation, templates (and partial templates) were represented as data structures containing over twenty separate elements. Almost all of these elements, however, could be regenerated from a lesser amount of basic information.

The least information necessary to represent a template is a history of which items have been inserted, and where. This means a list of which news items have been inserted, listed in the order of insertion, along with a specification of the columns occupied by their tops. Additionally, it is necessary to note for each item whether its top is or is not required to be at the same height as the previous item's top, or at the same height as the top of a level, and it is necessary to specify the set of story-picture formats chosen. If this information is known, all other necessary data can be generated. In particular, the matrix form of the template can be generated, and from it, the layout equations.

As space permits, more data can be included in the representation of a template, which, although redundant, will increase the speed efficiency of the algorithm. In particular, state information concerning the template construction can be helpful. In the partial template:

```
A B B B B C

D D D - - -
```

where B is a picture, the next item inserted must begin in the fourth column, and must be constrained to be at the same height as D. This fact can be effectively encoded as a state variable. Other state variables might include the set of news items remaining, their total area, the total area of the items already present in the partial template, and so forth.

10.1.2 Tableaux

The linear programming tableaux can require a great amount of space for their storage. A tableau is represented as a two-dimensional array of numbers, where there is a row for each equation, and a column for each variable.[1] Of course, it is necessary to provide space for only one tableau at a time. The tableaux are not particularly sparse (i.e., most elements are non-zero, especially toward the end of the solution of the linear programming problem), so representation techniques for sparse arrays are not applicable here.

10.1.3 Possibility Lists

A possibility list can be used to store the set of possible moves based on the set of current positions. One structure for the possibility list is shown in Figure 10-1. Beneath each current position are listed the items which could be inserted, and below each of the items is listed the set of possible moves which could be made to insert that item. Each of the current positions should be represented in the format for a partial template; this information need not be repeated in the list of moves, leading to a space-efficient implementation which

current positions:

items:

moves:

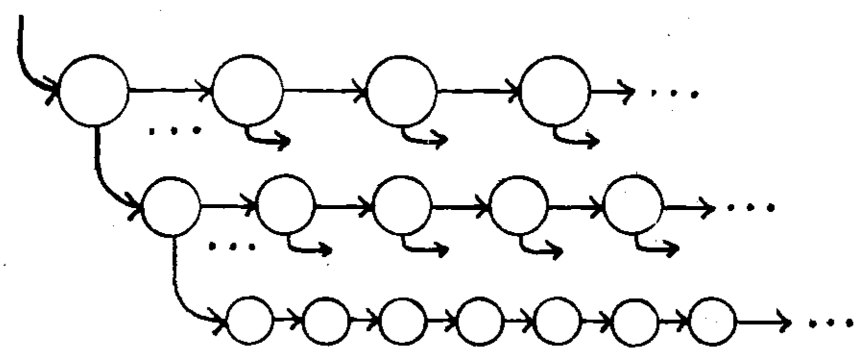


Figure 10-1. Format of Possibility List.

1. And for each "slack variable", implicit in each inequality.

is particularly desirable in light of the fact that there can be hundreds of moves possible.

Each move should also be associated with its quality factor, as described in Chapter 7. This factor can be used to control the selection of possible moves in the following fashion, assuming that all factors are between 0 (the worst) and 1 (the best).[2] When a set of moves is to be selected, a current position is chosen at random, then an item to be inserted into that position, then a move inserting that item. When such a randomly determined move is reached, a random number between 0 and 1 is generated, and compared to the quality factor of the move. If the random number is less than or equal to the quality factor, the move is removed from the tree and, if valid, inserted into the set of valid moves. If the random number is greater than the quality factor, the move is left untouched. This method can be seen to tend to examine higher-quality moves before lower-quality ones. It is superior to direct selection of moves with the probability of selection of each move equal to its quality factor in that it is unnecessary to compute and update aggregate probabilities for each of the current positions and items.

A move which has been selected is removed from the possibility list so that it will not be selected twice. A current position or an item for which there are no possible moves remaining should also be removed.

In breadth-(N_1,N_2) search, once a valid move is selected, it is placed in a set of valid moves, which will reach size N_2. These moves can perhaps best be represented in the partial template format, with their associated values of the criterion function.

---

2. Although some moves may have a factor of 1, no moves should have a factor of 0, since this will signify that the move is never to be chosen. Such a move should not be placed in the possibility list initially.

1Ø.2 Instruction Set

Minicomputers typically have a more limited repertoire of instructions than do larger computers. This limited instruction set is usually accompanied by a smaller word size. In general, neither of these limitations should prove a real problem. Most of the operations performed by the layout algorithm can be represented by logical operations on small fields, with few arithmetic operations required. The arithmetic operations which are needed are typically simple, and no difficulty should be caused by the smaller word size.

The exception to the above is the simplex algorithm for linear programming. The tableau will contain non-integral numerical values. If the machine being used has floating-point operations, then the floating-point number format can be used for these values. Experience shows that a floating-point number format with a 27-bit mantissa presents no real problems with the accumulation of round-off and truncation errors during linear programming. This should also be the case with a slightly smaller mantissa.

If floating-point operations are not present, they could be simulated through the use of software routines. It is probably more efficient, though, to implement routines for rational arithmetic, in which each number is represented in the form a/b, where a and b are integers. Assuming that rational numbers are always represented in reduced form (i.e., where a and b have a greatest common denominator of 1), the sizes of a and b should remain sufficiently small to be stored as 16-bit numbers. In particular, denominators seem to stay below 10 in value, assuming a six-column page. Numerators can be higher, especially when representing values of variables, which are measured in the relatively small units of column-points. If they are measured in larger units, this can cause the sizes of the numerators to shrink, possibly accompanied by some growth in the size of the denominators.

It is also possible, of course, to store numbers in a fixed-point format. It is not known what particular problems this might cause, but it is suspected that this might lower the accuracy of the numerical operations.

When numerical values are obtained by linear programming, these results, representing some number of points of height, might be fractional. In general, replacing these values by the next higher integral number of points is appropriate.

10.3 Speed of Execution

As was described in the last chapter, the test implementation was able to achieve a reasonable speed of execution, taking only a few minutes to solve large layout problems. It is believed that this speed can be improved upon in a production implementation.

The test implementation was written in Lisp, and run on a computer with only a 300,000 instruction-per-second execution rate. This basic rate can be greatly surpassed by modern minicomputers. Moreover, hand-coding of the layout procedures can produce a further improvement in their speed.

Monitoring of the test implementation shows that the majority of time is spent in generating possibility lists and in linear programming. Special attention should be paid to these sections.

One cause of the large amount of time spent in generating possibility lists is the large number of possible moves that must be enumerated. Not all of the possibility list, however, will typically be examined once it has been generated. A more time-efficient implementation would be to have the possibility list generated incrementally, while it is being examined. In other words, only a skeleton of the possibility list is generated initially; the set of moves for a given current position and a given item to be inserted is generated the first time that pair is selected. In cases where the number of such pairs possible is greater than the number that will be considered, such incremental generation can result in significant time savings, by generating only that part of the possibility list that is actually needed. Further compartmentalization of moves in the possibility list can result in further efficiency gains. It should be kept in mind, though, that changes in the structure of the possibility list might lead to undesirable biases in move selection.

Of course, the amount of time used to generate a single move should also be kept as small as possible. One way of doing this, as mentioned above, is to include redundant data in the representation of a template, trading off space efficiency for time efficiency.

The speed of linear programming can be improved through the use of careful coding techniques. It will be slower, however, if software simulation of arithmetic is necessary.

10.4 Summary

Special care must be taken when implementing the layout algorithms described in this report on a minicomputer system, because of the limitations of such machines. Consideration of these limitations can lead to implementations for which the first consideration is space efficiency, but where the time efficiency is also adequate.

Chapter 11 - Open Problems

There exist certain open research problems relating to the news layout procedures outlined in this report. Before these procedures can be implemented and used in a production environment, these problems must be considered and resolved. They relate both to understanding more fully the layout algorithms developed in this thesis, and to evaluating the utility of possible additions to the basic approach.

As was stated in the last chapter, production versions of the news layout program will probably be implemented on relatively small computer systems. Such systems, however, are seldom the best choice for basic development work. To the extent that these unresolved research problems deal with the layout algorithms themselves, independent of the particular implementation, this work can be performed on a large-scale computer system offering a wide range of program-development tools. Once the complete details of the layout algorithms to be used have been determined, these algorithms should be transferrable to the desired target computer system without creating any significant problems of a research nature.

This chapter presents a number of known research problems which should be considered. It begins with problems pertaining to features of the layout program described in this report, and leads into problems related to possible additional features.

11.1 Biases in Move Selection

Any particular method of move selection is based on some particular bias. Any such bias will have an effect on the relative probabilities of selection of the various possible final layouts. For example, in the experimental implementation, for the Tech Talk layout example presented in Chapter 5, the probability of selection of the final template:

```
E E E S S
E E D D D
E A A D D
```

is 0.14, while the probability of selection of:

```
D D D D D
D E E E E
A A E E E
A A S S S
```

is only 0.00089, assuming a totally random breadth-1 search. This is primarily because the partial template:

```
E E E - -
```

has far fewer possible next-moves than the partial template:

```
D D D D D
```

since the top row must be filled in in the first case, meaning that each individual next-move has a greater chance of being selected. A basis is needed for understanding which move-selection method produces the "best" results.

11.2 Tuning

An implementation of the layout algorithm will include many parameters which can take on various values. It has been shown that the particular values used for these variables can have significant impact on system efficiency. A process of *tuning* is necessary to achieve appropriate values for these parameters. Since each layout problem is different, it seems unlikely that perfectly optimum values could be determined, although certain uniformly good values might exist. More research is necessary to better understand this process.

One possibility in some cases is *automatic tuning*. This would have the system itself automatically adjust the parameters as necessary. For example, if multiple passes through the layout tree were necessary for a particular layout problem, the breadth of search could be increased on each pass in recognition of the apparent difficulty of the problem. Similarly, it might be possible to automatically tune the weights of the esthetic rules, based on feedback from the layout person on the esthetic quality of the layouts produced.

11.3 Internal Story Formats

There are more possible elements to a story than have been considered in this research; hence, it must be determined how the additional elements affect the generation of layout equations. These elements include secondary headlines *(kickers)*, boxed exerpts *(sandwiches)*, and so forth. All such story elements should be considered, and a disposition determined for each.

Stories with more than one picture present a problem. Experience shows that a small number of possible story-picture formats is sufficient for stories with only one picture. (In this way, the set of possible story-picture formats acts like the template library proposed by Kan, discussed in Appendix IV, except on a smaller scale; only three or four formats are needed.) For stories with more than one picture, it is expected that this number would have to be much larger, causing a vast increase in the number of possible moves when such stories are inserted. Instead, it is proposed that a set of appropriate formats be specified by the layout person for every story with more than one picture. This is particularly reasonable in light of the fact that the proportion of such stories is small.

In this case, and in the case when unusual story formats are desired, the layout person will have to provide a format to be used by the layout program. It would be desirable to have a very powerful general means for doing this, whereby even unusual formats could be described and used.

One partial alternative to the use of internal formats for generating layout equations might be to represent all item components directly in the matrix form of the template. Coupled with the use of asynchronous layout, it might be possible for such a representation to be no less general than the representation scheme used in this thesis.

11.4 Unusual Item Shapes

Chapter 3 presents a template grammar which, it is believed, is the simplest such grammar which does a good job of representing the set of legal templates. However, it will not necessarily describe the exact set of legal templates for any particular newspaper. If desired, the rules can be changed to reflect the local conditions at the newspaper. On the other hand, since the rules given in Chapter 3 lead to such a simple generative grammar, it might be more reasonable to leave these rules as they are.

As a example of a possible change to the template grammar, the template shown in Figure 11-1 contains an item of illegal shape. It is believed that in most newspapers such shapes are used only for the sake of expediency, and are not as desirable as shapes considered legal in this thesis. (On the other hand, the New York Times uses items of such a shape often.) Similarly, item shapes such as the illegal one in Figure 11-2 are used in some newspapers.

If the generative grammar were to expanded to allow these shapes by permitting a growth in the width of an item after its top has been inserted into

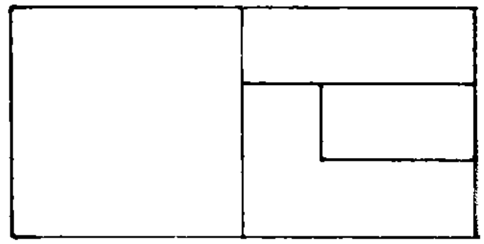


Figure 11-1. Example of Template with Item of Unusual Shape.

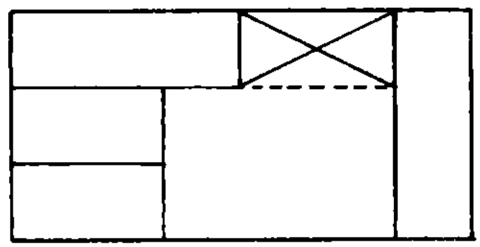


Figure 11-2. Example of Template with Item of Unusual Shape.

the template, this would have an undesirable effect on the partial layout equations. For instance, it would be impossible to determine what part of a partial template would be available to hold the incomplete items.

11.5 Variable-Width Columns

Most newspapers have a nominal six-column or eight-column structure. In many cases, though, the sizes of ads sold do not conform directly to the chosen structure. In such cases, various column sizes must be used, sometimes on the same page, as shown in Figure 11-3. Variable-width columns can also be forced by pictures of odd width, or may even be used voluntarily.

It is expected that achieving layouts with variable-width columns might prove difficult within the framework provided in this thesis. It is difficult to see how the definition of a widened template might be appropriately changed. It would also be necessary for the layout program to do some amount of

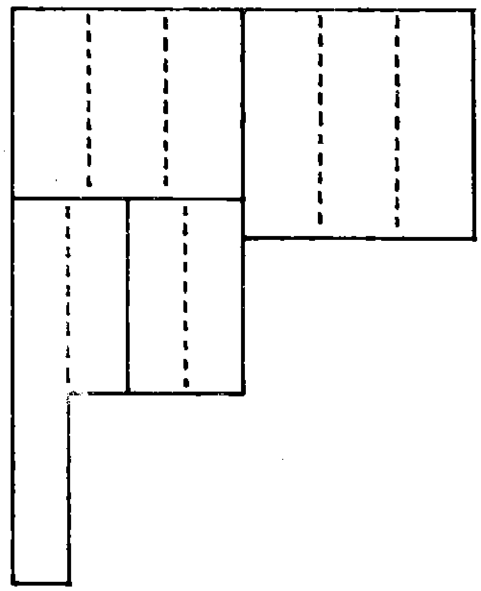


Figure 11-3. Variable-Width Columns.

planning ahead to anticipate transitions between the column widths. On the other hand, such layouts are often ugly, and are not strictly necessary, as evidenced by the New York Times, which entirely avoids them in its news sections. It might be more reasonable at least to decrease the need for such layouts.

11.6 Item Clustering

If related items are to appear on the same page, they should be adjacent to each other. One way to achieve this might be by means of an esthetic rule, as discussed in Chapter 7. Such a rule could not guarantee that related items would be clustered properly, but could only penalize those templates that failed to comply. It is expected that such an approach might prove very difficult to control.

Another approach might be to note that the overall shape of such a cluster is typically the same as a legal shape for a single item. Thus, the "top" of the cluster could be inserted into the template, the appropriate items inserted underneath it in following steps, and then the rest of the items inserted. This procedure assumes the use of asynchronous top-down layout, of course. More restrictive partial layout equations could be generated at intermediate points in this process, since it would be known that the items in the cluster could be inserted only under the cluster top.

11.7 Outside-In Layout

As was shown in Chapter 5, top-down layout is superior to bottom-up layout. A combination of the two might prove superior to either. Such a scheme can be called *outside-in layout*, because it inserts items both top-down and bottom-up, working both ends against the middle. The use of top-down layout alone, for example, is unlikely to produce layouts with a large item at the very bottom of the page, but such layouts are often used in actual newspapers.

The problem with such an approach would be to tune it properly. For example, we know that top-down layout alone is better than bottom-up layout, which suggests that relatively more items should be inserted from the top of the page than from the bottom. What proportion of items placed from the bottom of the page will result in the greatest improvement over basic top-down layout, if any, is not known.

11.8 Incremental Story Assignment

The number of dead ends increases dramatically toward the bottom of the layout tree. This is partially because the set of items remaining to be inserted into the template becomes much smaller, allowing less flexibility in move selection. It has been assumed in this thesis that a story-assignment phase provides the layout program with the exact set of items which are to appear on the given page. An alternative would be to provide a larger set of items than will fit, allowing the layout program some discretion in choosing which items to place. In practice, this will increase the number of possible moves, decreasing the probability of dead ends. Of course, the definition of a final position would have to be changed to be that of a position where enough of the items have been inserted, measured by comparing the sum of their areas to some threshold value. Note that, in this scheme, valid final positions might have descendants.

A basic version of this approach has been implemented and seems to work well. It requires that story assignment be executed *incrementally,* since it is not until the page has been finally laid out that it is known which items will be available for the remaining pages. This might require a number of subsidiary changes in the organization of story assignment.

This incremental story-assignment approach seems particularly applicable to pages containing stories that might be jumped to some other page. In the non-incremental version, it is necessary to pick for each such story some arbitrary fraction to be laid out on the given page, with the rest then being assigned to following pages. With incremental story assignment, it is possible to allow more of a choice for the layout program. This means that the layout equations can be very loose in stating the minimum and maximum amounts of each story that will appear on the page, increasing the ease of layout. Unfortunately, this seems to mean that virtually any generated template will be valid, possibly leading to ugly random layouts. On the other hand, since pages containing such stories, such as the first page, would tend to be laid out by hand anyway, this ability might not be necessary. This topic should be considered further.

11.9 Planning

The layout algorithm presented in this thesis makes no real effort to plan ahead when constructing a template, due primarily to the use of the layout tree as its conceptualization of the layout game. A similar lack of planning has been observed in chess-playing programs. An opposite approach would be an algorithm which began by making an overall *plan* for how it would go about producing the template, modifying the plan as it encountered problems. Although this extreme is probably unrealizable given the current understanding of the problem, perhaps certain aspects of it could be incorporated into the layout algorithm.

11.10 Criticism

Some of the layouts produced by the layout program will be unacceptable to the layout person. In certain cases it might be possible for the layout person to determine the reason for the unacceptability of the layout. It would be desirable if such reasons could be communicated to the layout program, and if the program could then produce a similar layout without the offending feature.

11.11 Human Interface

Any production implementation of the layout program will have to pay particular attention to the human interface to be provided for the layout person. The current view of the interface is as follows.

We assume that the layout person is seated at a display console. For each page, the layout person is presented with a list of the news items to be placed on that page, and can override the choice manually if desired. (In general, it is desirable for the layout person to be able to override any system decision; on the other hand, the number of points where it is necessary for the layout person to make decisions for the layout program should be kept as small as possible.)

For each of the items, it will be possible to specify certain constraints, such as a fixed headline or set of headlines, fixed format or set of formats, and so forth. If these constraints are not explicitly stated, default values will be assumed, such as a set of possible headline widths and heights based on the size of the story. As noted above, although a set of default formats can easily be provided for stories with one picture, this will probably be less reasonable for stories with more than one picture.

The layout person will then be allowed to pre-place some items in the template, if desired. This might be restricted to placing items either at the top of the template or at the bottom, as opposed to starting in the middle. If desired, the layout person can construct the entire template manually. If not, the layout program will generate one or more itself.

During the operation of the layout program, there is not much information that can reasonably be displayed. In breadth-(N_1,N_2) layout, the current state includes N_1 current positions and N_2 moves based on them. This is too much information to be absorbed by the layout person in the short amount of time during which it will be applicable. Instead, it is suggested that various indications of system operation be displayed, such as the amount of time spent on the page so far, the number of passes through the layout tree, the depth of the current ply, how many moves were examined at each ply, and so forth. The information provided should be sufficient to enable the layout person to decide how well the page layout is proceeding, and to vary the appropriate system parameters if necessary.

At any intermediate point, the layout person can halt the layout program and supply a template manually, or modify the input to the program, such as the set of items to be laid out, and try again.

The layout person will be shown each layout produced, and will be allowed to decide whether it should be accepted or whether more layouts should be generated. It will also be possible to return to a previous layout.

After a layout has been produced, it will be displayed. This will be a skeleton layout. It will still be necessary to determine a number of details, such as writing remaining headlines, deciding on headline typefaces, drawing lines around items, and so forth. It will also be possible to modify the numerical component of the layout at this stage, if desired.

11.12 Summary

There are a number of remaining research problems associated with the approach to computerized news layout presented in this thesis. Any potential implementor should consider these before finalizing the system design.

Chapter 12 - Conclusions

The conclusion to be presented here is that an approach to the problem of news layout based on the incremental generation of templates is feasible. As shown in Chapter 9, the algorithms developed are capable of laying out news pages in actual newspapers, and of producing layouts of quality comparable to layouts produced by hand. Although much developmental work remains to be done before the results can be applied operationally, the basic approach has been demonstrated to be sound, and it is believed that there are no significant obstacles to its ultimate use in a newspaper environment. The ability to lay out many newspaper pages by computer should provide an important dimunition in the amount of time and effort required for a newspaper office to produce an edition.

Appendix I - Human Layout Methods

1. Background

One approach to the construction of a computerized news-layout system might be to discover how humans solve the problem. In the past, when layout persons have been asked to explain their "algorithms", they have been unable to do so. This appendix presents a series of experiments which examine the performance of a layout person when solving layout problems.

2. The Experiments

In these experiments, an experienced layout person from *The Tech*, a student newspaper at M.I.T., was presented with a set of layout problems based on layouts taken from *Tech Talk*, an M.I.T. administrative newspaper. Both newspapers are tabloid-size, five columns wide and 16 inches tall. The layout person was asked to verbalize her thought processes as she laid out the pages.

2.1 The First Experiment

In this experiment, the news items shown in Figure I-1 were to be laid out. This is Page 8 of the February 9, 1977 issue of Tech Talk. The original Tech Talk layout was provided only to allow the layout person to acquire a feel for the relative sizes of the news items; numerical size measurements were also provided. The following is an edited transcript of the layout person's commentary.

"One thing I could try would be putting.... I could either put the [Dancing] *picture in the upper-right corner with the headline running above it, or having the headline just a little.... For the time being, I'll leave the headline the same*

Katie Earnshaw Mixes MIT Duties with Dancing Future

By KATHARINE S.C. JONES
Staff Writer

Katherine Earnshaw is a professional dancer enrolled in a master's degree program at Connecticut College and an administrative assistant in the MIT Department of Nuclear Engineering.

How does she manage to do so much? She attributes her success at combining the demands of a dancer's career with a full-time job to the flexibility of MIT and the support of her husband and her supervisor. She is married to William Earnshaw, a doctoral degree candidate in biology at MIT.

Ms. Earnshaw began working at MIT in 1973 as a clerk in nuclear engineering headquarters. Three evenings a week she went to modern dance classes. When she learned of an opportunity to study with dancer choreographer Dorothy Hershkowitz in the fall of 1975 she thought she would have to quit her job and work part time elsewhere. Her supervisor, John Cochrane, administrative officer in the Department of Nuclear Engineering, suggested they rearrange her schedule so she would have time for daytime classes. Her schedule was adjusted and now she has a class four days a week with Ms. Hershkowitz plus one ballet class with the Boston Repertory Ballet.

Ms. Earnshaw estimates that she devotes nine to ten hours each week to dance, more when she's rehearsing for a public performance. She spends 35 hours a week working at MIT.

"I try to be sensitive to the pressures of my job," she said. "I'd cancel a dance class if something urgent came up." Although her schedule is hectic, "I'm much happier than I was two years ago," she said.

Her modern dance classes in the fall of 1976 were repertory classes, and resulted in the premiere of "Monday Morning Quarterback (An American Dance Masque 1976)," a new work by Ms. Hershkowitz performed by Dance Circle on January 6-8 at the First Congregational Church in Cambridge. The dance was well received and highly acclaimed in the Boston press. It is "the best work yet by this talented choreographer," wrote Laura Shapiro of the Boston *Globe*. Iris M. Fanger of the *Christian Science Monitor* called it a "provocative work which deserves to be seen again by a wider audience."

This request has been granted. "Monday Morning Quarterback" will be performed Friday through Sunday, February 11-13, at the Joy of Movement Center (23 Main St.) in Watertown. Tickets cost $3 and are available at the Joy of Movement Centers in Cambridge and Watertown.

In addition to performances in this work, Ms. Earnshaw is working on a new piece with Christy Nichols, dance teacher at Pine Manor Junior College, to be performed March 25 and 26.

She began her dance career rather late. Her first dance lesson was at Colby College where she earned the BA degree in sociology in 1972. Colby did not have a dance program, but she took all the courses offered in dance history, composition, and improvisation.

"Dancing was something I always wanted to do but it was always at the back of my mind," she said. "Having started so late, I'm now trying to make up for lost time."

Ms. Earnshaw began working toward the master's degree in dance at Connecticut College in the summer of 1975. The program requires two summers and one academic year of training and study. She has completed two six-week summer sessions, and hopes to return for a full academic year in the near future. For each of the two summers she studied at Connecticut, she took a leave of absence from her job, and, when necessary, a temporary employee was hired to handle day-to-day office work.

Her ability to combine study and performance in dance with a full-time job has led Ms. Earnshaw to believe that "there are ways of working within the system at MIT to accomplish your career goals, and it's important for people to know that they can work something out."

DANCING A CHEERLEADERS' ROUTINE in "Monday Morning Quarterback" are members of the Dance Circle ensemble. That's Katherine Earnshaw on the far right. photo by George Siman

16 Named in Engineering School

The following appointments or changes of appointments have been announced in the School of Engineering:

Department of Electrical Engineering and Computer Science

Dr. Cyril Leung appointed as assistant professor until July 1, 1979. Dr. Leung, who received the B.Sc. Eng. Honours degree from Imperial College, London, completed his doctorate at Stanford University, where he was the 1975 King George VI memorial fellow in 1976. His research interests are in the areas of communication, information and complexity theories.

Mr. Markus Zahn appointed visiting associate professor Sept. 1, 1977. Dr. Zahn, who as an instructor of electrical engineering at MIT from July 1969 to July 1970, is associate professor in the Department of Electrical Engineering at the University of Florida, Gainsville.

Department of Chemical Engineering:

Dr. William M. Deen, appointed assistant professor in the department and in the Harvard-MIT Program in Health Sciences and Technology until July 1, 1979. Dr. Deen, who received the BS degree in chemical engineering from Columbia University in 1969, and the MS and PhD degrees in chemical engineering from Stanford University in 1971 and 1973, was adjunct assistant professor in the departments of medicine and physiology at the University of California, San Francisco, from January 1975 until June 1976.

Dr. John P. Longwell, change of appointment to professor in the Department of Chemical Engineering from visiting professor beginning Jan. 1, 1977. Dr. Longwell, formerly senior scientific adviser in corporate and government research at Exxon Research and Engineering Co., is a past president of the Combustion Institute and a member of the National Academy of Engineering. He received the BS degree in mechanical engineering from the University of California in 1940 and the ScD in chemical engineering from MIT in 1943.

Dr. Frederick A. Putnam, appointed assistant professor in the Department of Chemical Engineering until July 1, 1979. Dr. Putnam received the BA in physics, cum laude, from Dartmouth College in 1969, the MS in chemical engineering from Case Western Reserve University in 1975 and the PhD in chemical engineering from Carnegie-Mellon University in 1976. His research areas of interest are interfacial phenomena, physical adsorptions of gases, adhesion and adhesion, solid surface characterization, and thermodynamics and statistical mechanics.

Dr. Preetinder S. Virk appointed associate professor in the Department of Chemical Engineering until July 1, 1979. Dr. Virk, who received the Doctor of Science degree from MIT in 1966, was an assistant professor of chemical engineering at MIT from 1966-1971 and an associate professor from 1971-72. From 1972 until he returned to MIT he was a professor of chemical engineering at the Indian Institute of Technology in Madras, India. He is a consultant for the Process Industries Group at Stone & Webster Engineering Corp.

Department of Civil Engineering.

Dr. Oral Buyukozturk, appointed associate professor in civil engineering until July 1, 1979. Dr. Buyukozturk received the MS in civil engineering in 1963 from Technical University of Istanbul, Turkey, the MS in structural engineering from Cornell University in 1969 and the PhD in structural engineering from Cornell in 1970. Dr. Buyukozturk formerly was adjunct associate professor of engineering at Brown University.

Department of Mechanical Engineering

Dr. Adam C. Bell, change of appointment to associate professor from visiting professor until July 1, 1979. Dr. Bell, who received the BEng degree in 1960 from Nova Scotia Technical College, holds the SM, ME and ScD degrees from MIT. He became visiting associate professor in mechancial engineering at MIT in September 1975 after being on the engineering and applied sciences faculty at the State University of New York at Buffalo from September 1968—September 1975.

Dr. Peter W. Huber, change of appointment to assistant professor in mechanical engineering from lecturer until July 1, 1978. Dr. Huber, who received the SB and SM degrees in 1974 and the PhD in 1976, all from MIT, has been a research assistant and part-time teaching assistant at the Institute since 1972.

Dr. Alician V. Quinlan, appointment as Arthur D. Little Assistant Professor of Environmental Sciences and Engineering. Dr. Quinlan, who also is a Rockefeller Foundation Fellow in Environmental Affairs and an assistant professor of mechanical engineering, received the SB in life sciences in 1968 from MIT, the MS in physical oceanography in 1970 from the Institute of Marine Science, University of Alaska, and the PhD in environmental engineering in 1975 from MIT. Her main fields of interest are environmental engineering, element-cycle dynamics in aquatic and terrestrial ecosystems and reaction transport dynamics in living systems.

Dr. Derek Rowell, change of appointment to assistant professor in mechanical engineering from Sponsored Research Staff and lecturer in mechanical engineering until July 1, 1979. Dr. Rowell received the BE degree in 1966 and the PhD degree in 1970, both from the University of Canterbury, Christchurch, New Zealand. He is director of the Sensory Aids Evaluation and Development Center at MIT.

Dr. Hendrik G. Stassen appointment as visiting associate professor in mechanical engineering until Aug. 1, 1977. Dr. Stassen, associate professor in control engineering at Delft University of Technology, The Netherlands, since 1968, specializes in control engineering, man-machine systems and rehabilitation of the handicapped. He received the BSc degree in 1960, the MSc degree in 1964 and the Dr. of Technology degree in 1968, all from the Delft University of Technology.

Dr. Nak-Ho-Sung, change of appointment to assistant professor of mechanical engineering from research associate and lecturer until July 1, 1978. Dr. Sung received the BS in chemical engineering from Seoul National University, Korea, in 1964, the MS in physical chemistry from the University of Chicago in 1967 and the ScD in mechanical engineering from MIT in 1972. He joined MIT in 1967 as a research assistant in the Fibers and Polymers Division of the mechanical engineering department.

Department of Nuclear Engineering

Dr. Mujid S. Kazimi, appointment as assistant professor in the Department of Nuclear Engineering until July 1, 1979. Dr. Kazimi received the B.S. in nuclear engineering from the University of Alexandria in 1969, the MS in nuclear engineering in 1971 from MIT and the PhD in nuclear engineering in 1973, also from MIT. From May 1974 until coming to MIT he was associate nuclear engineer, Department of Applied Science, Brookhaven National Laboratory. His main fields of interest are nuclear reactor safety analysis, nuclear reactor design and optimization heat transfer and fluid mechanics in two-phase media and scientific advancement and technology transfer in developing countries.

Dr. Puthenveetil K. John appointed as visiting associate professor part-time in the Department of Nuclear Engineering, until July 1, 1977. An associate professor of physics at the University of Western Ontario since 1972, Dr. John received the BSc degree from Kerala University, India, in 1961, the MSc from Madras University in 1965 and the PhD from Ohio State University in 1963.

Department of Ocean Enineering:

Cmmdr. Clark Graham, change of appointment to adjunct professor from associate professor of ocean engineering until Sept. 1, 1979. Dr. Graham received the BS degree from the U.S. Naval Academy in 1964, the SM in mechanical engineering from MIT in 1967 an the PhD in mechanical engineering in 1969, also from MIT. His principal fields of interest are high-performance marine vehicles, marine engineering, marine power systems and heat transfer.

Toxic Chemicals Symposium Planned For February 15

A symposium on the analysis and assessment of toxic chemicals in the environment will be held at MIT Feb. 15.

The chairman of the symposium—which will begin at 9am in Kresge Auditorium—is Professor Ronald A. Hites of the MIT Department of Chemical Engineering. Professor Hites has been involved in major research projects on the application of analytical chemical techniques to the solution of environmental and energy-related problems.

"Concern about toxic chemicals in the environment has never been higher," Dr. Hites said.

"With increasing frequency, industrial, governmental and academic laboratories are being called upon to identify and quantitate potentially toxic chemicals in air, water and products and to assess the toxic effects, if any, of these chemicals.

"The objective of this symposium is to review current research techniques for the analysis of chemicals in the environment and for the determination of the toxic effects of such compounds.

"Most of the emphasis will be on organic compounds in water and their measurement by gas chromatographic mass spectrometry. The use of cellular toxicology and industrial epidemiology to assess toxic effects of chemicals also will be reviewed."

Professor Hites, in an article published in a recent issue of *Analytical Chemistry*, said the methods currently employed by most industries to identify chemicals in their wastewater do not detect many potentially toxic materials.

Reporting on a research project that examined the wastewater at two tire manufacturing plants, Professor Hites said in the article that the use of gas chromatographic mass spectrometry detected potentially toxic chemicals several miles from where one plant's wastewater entered a river.

The symposium is sponsored by the National Science Foundation's Division of Advanced Environmental Research and Technology.

Some 125 people signed up for the new MIT-Cambridge chapter of the American Association of Retired Persons at its organizational meeting last Thursday in Rm 9-150. Ballots for electing officers, the board of directors and the nominating committee will be mailed to all who signed up within the next week or two. Those eligible to belong (age 55 or older) who did not sign up at the meeting may do so by calling x3-7914.

Page 6, Tech Talk, February 9, 1977

Figure 1-1. News Items for First Experiment.

as they, and.... Well, all right, I'll assume half an inch [for the headline]. *Try putting the headline across 5 columns with the picture 3 1/4 inches deep, three columns. Well, just so it's even, I'll make it* [the picture] *a little bit bigger. That's 3 1/3, with two lines underneath, so that's about a third of an inch. And the story is 15 1/2. So I can run it in these two columns* [see Figure I-2].

"And I can either run it [the text].... *I could run it under the picture, so that it's even, and then have another.... I could try a five-column headline* [for the Engineering story] *since the story's so long. All right, I'll try that* [see Figure I-3].

"That's a little more than half an inch [for the Engineering headline]. *I think I'll try putting the AARP picture down here for balance, I guess.... The lower*

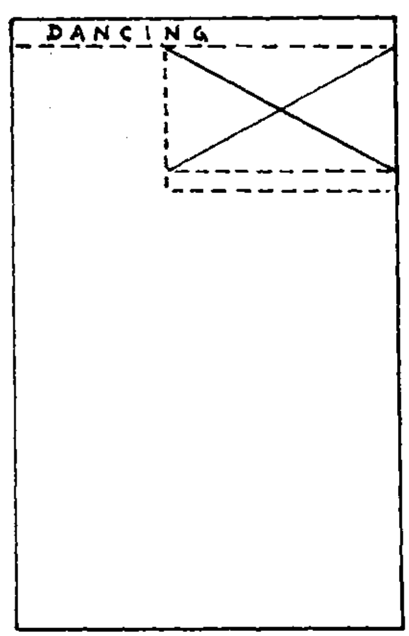


Figure I-2. Partial Layout for First Experiment.

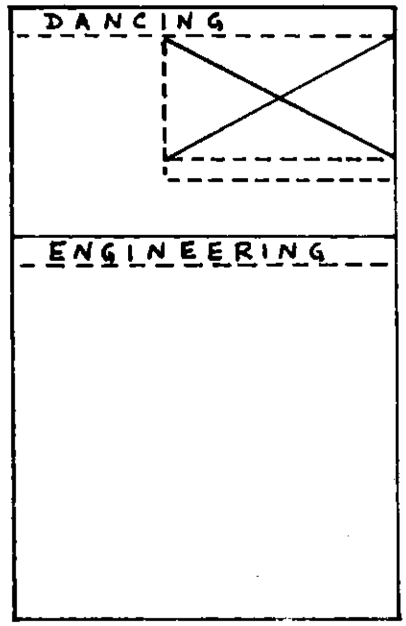


Figure I-3. Partial Layout for First Experiment.

left-hand corner.... So that's 2 3/4 inches high, plus about an inch or so of cutline [see Figure I-4].

"And if I run this headline across 5 columns.... All right, so that means if I put the Symposium story down here, everything should work out [see Figure I-5].

"That's one possibility. I'm not saying it's the best layout, but at least it gets the headlines apart from each other."

The elapsed time for this experiment was 4 minutes 50 seconds.

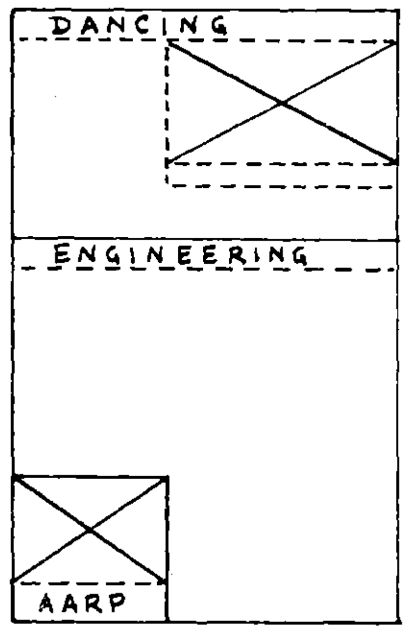


Figure I-4. Partial Layout for First Experiment.

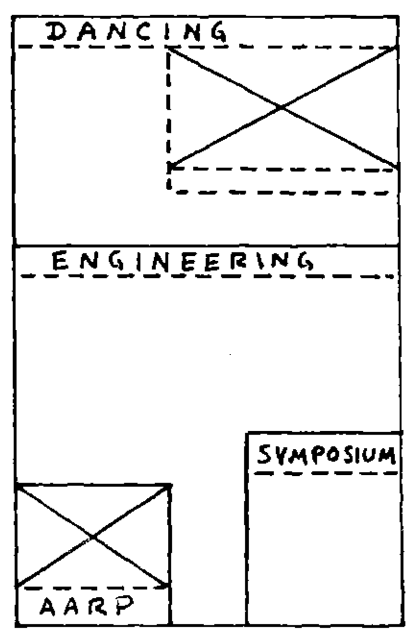


Figure I-5. Final Layout for First Experiment.

2.2 The Second Experiment

In this experiment, the layout person was to lay out the news items shown in Figure I-6. This is Page 5 of the February 9, 1977 issue of Tech Talk.

"*I think I'll try to get rid of this big blotch here* [the Energy jump] *for one thing, since the.... I could try to put the Energy jump lower down, and put some stories that are starting here on the top. So, let's see. Well, the longest one is Media, so I guess I'll try putting that at the top, moving over 3 columns, I guess. If I spread it over 5 columns.... Well, I think I'll try keeping the picture in the upper right corner: that's the Edgerton picture that doesn't go with anything* [see Figure I-7].

"*So that's 3 inches of picture and about 2 inches of cutline, and run the Media story in the upper left* [see Figure I-8].

"*I think I'll try to run a story in 3 columns underneath the picture, and have the copy of Media in the first two columns* [see Figure I-9].

"*Okay, three short stories and one long one are left. I could put a story in here* [under Edgerton], *say.... That's a little too big for any of these stories.*

"*I'm thinking of maybe putting the.... That picture could get a little bigger. Let's see. That would come down half an inch, and then I would have about a 5-inch story* [under Edgerton]. *Booklet.... It's probably a stupid place to put it, since Booklet can't be very important. Oh well, nothing looks terribly important. All right, I'll try putting ...* [see Figure I-10].

"*Now, I could either have a five-column headline* [for the Energy jump], *or ... or I could run it in 3 columns and put the graphic in* [to the right of the headline]. *I don't really like this. It's working out so I can either.... I could put the chart on one side here* [on the left], *and the headline here* [on the right],

Energy Savings Reported

Continued from page 1

affect interior environmental conditions across campus creating an integrated communications network with a central control.

Traditional HVAC heating, ventilation and air conditioning systems have control thermostats and can provide only two settings day night or heat cool. A computer controlled system of this type called "super-clock" functional at MIT for several years, did a creditable job of energy conservation, Mr. Shepherd said, but it has been set aside in favor of the more advanced methods of EMS.

A centrally controlled, intelligent system is the best way to optimize the use of energy, Mr. Shepherd said, and the facilities management system is one of the most advanced building control systems we are aware of.

The new system is a dynamic one. Response models built into the central software enable it to sense and react to environmental changes. Savings are achieved through more efficient use of HVAC machinery, minimizing their energy requirements and through electrical load shedding. The latter is a technique of limiting costly peak electrical load demands by temporarily turning off non-essential equipment during periods of high energy use.

"With only twenty-five buildings now connected to the system," Mr. Shepherd said, "we are already saving an average of 1,000 kilowatts of electricity per hour. We expect the system to pay for itself within two years," he said.

The Institute is currently paying $14.80 per barrel of fuel, a $2.70 increase over last year's price. Since 1970, when dramatic price increases in fuel costs began, the cost of No. 6 fuel oil, which supplies most of MIT's heating energy, has risen from $1.87 per barrel to the current rate, the highest MIT has ever paid.

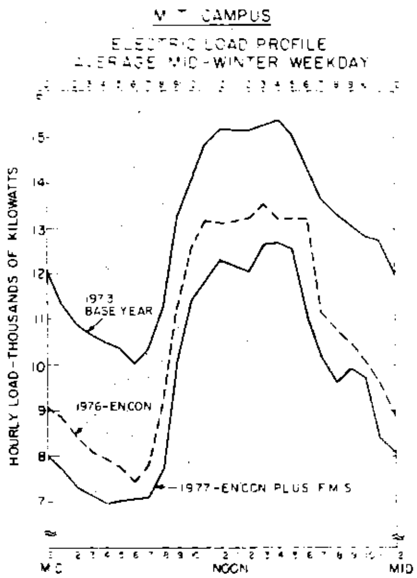
"A 10 to 15 percent inflation rate for the cost of fuel has been built into the budget," Mr. Shepherd said, "but a severe winter such as the one we are now experiencing can increase costs by as much as $150,000 over the anticipated levels."

This winter has been about 15 per cent colder than normal and 26 per cent colder than last year, according to Professor August L. Hesselschwerdt, Jr., consultant to Physical Plant. He expects this trend to continue through the end of the winter.

A normal winter is based on a 30-year average of temperatures computed in degree days, a form of accumulated temperature for specifying nominal heating load in winter. A degree day is a one degree (F) temperature difference between inside and outside based on a reference temperature of 65°F (19°C) inside. The number of degree days in a 24-hour period is computed by subtracting the mean temperature for that period from 65°F. For example, for a 30-day month in which the outside temperature averages 20°F for 10 days and 35°F for 20 days, there are 1050 degree days. Degree days = 10(65-20) + 20(65-35) = 1050.

By Jan. 28 of this year the total number of degree days was 3092. This is 641 degree days more than the total recorded during this time period for last year and 407 more than the 30-year average.

"Even with this extreme cold," Mr. Shepherd said, "the voluntary participation of the community in our energy conservation program continues to be outstanding. Of the $2.5 million annual cost avoidance attributed to the ENCON program, 20 percent of the savings are directly attributed to community action. The Department of Physical Plant commends these outstanding efforts to conserve resources and urges continued support," he said.



MONITOR MONITOR—MIT's Harold E. "Doc" Edgerton, right, famed for his development of the strobe light, underwater photography and sonar, checks equipment that will be used to make photos of the Civil War ironclad Monitor, which sank off North Carolina's Cape Hatteras during a gale in 1862. The location of the famed warship was confirmed by a 1974 expedition that used sonar and photographic equipment designed by Edgerton. The new equipment, on which Dr. Edgerton and Bill MacRoberts, project technician in the MIT Strobe Lab, are working, includes a television camera, a 35mm camera and a strobe light. Notice the TV picture of Professor Edgerton's hand on the TV monitor on the table to the left. Professor Edgerton says the equipment will be used to make photographs from the side of the Monitor. Earlier photos were all of overhead views. The new equipment can be positioned by activating small water pumps mounted on the frame.

Neustadt to Close Election Forum

Dr. Richard E. Neustadt, professor of government at Harvard University, will speak on "The Presidential Transition," at the last program of the Agenda '76 discussion series on Wednesday, Feb. 16, at 5 pm in the Millikan Room E53-482.

Best known for his books, *Presidential Power* (1960) and *Alliance Politics* (1970), Dr. Neustadt recently served as a consultant to the Carter-Mondale transition group.

Dr. Neustadt has been closely associated with the workings of the federal government for the past thirty years. A member of the White House staff from 1950-53, he became consultant to the Senate Subcommittee on National Security Staffing and Operations in 1959. In 1960 he served as special consultant to President-elect Kennedy and became a consultant to the President in 1961, continuing in this position under Presidents Kennedy and Johnson through 1966. Dr. Neustadt has also served as consultant to the Bureau of the Budget, State and Defense Departments and the Atomic Energy Commission.

From 1965-75, Dr. Neustadt was associate dean of the John Fitzgerald Kennedy School of Government at Harvard and served as founding director of the School's Institute of Politics from 1965-71. He received the AB degree from the University of California at Berkeley in 1939, and the MA, 1941, and PhD, 1951, degrees from Harvard University.

The discussion series is sponsored by the MIT Department of Political Science to promote an exchange on the critical issues facing the US. Sessions are open to the community and include a question and answer period and group discussion.

Booklet Includes MIT Women

Thoughts of five present or former MIT women are included in "Space for Women: Perspectives on Careers in Science," a new booklet just published by the Harvard-Smithsonian Center for Astrophysics in Cambridge.

MIT was represented by Tanya Atwater, assistant professor of marine geology in the Department of Earth and Planetary Sciences; Alice d'Entremont, '60, an electrical engineer at the Harvard College Observatory; Catherine A.H. Mink, '74, a scientific programmer at the Smithsonian Astrophysical Observatory; Judith Walzer, director of women's education at Radcliffe College and former humanities instructor at MIT, and Marlene Williamson, a former graduate student in systems analysis who is now a geophysicist at the Smithsonian Astrophysical Observatory.

Text for the booklet was drawn from a symposium addressing the satisfactions and difficulties encountered by women in traditional male careers, held in celebration of the International Women's Year.

Copies of the booklet are available free of charge from the Publications Office, Center for Astrophysics, 60 Garden St., Cambridge, Mass. 02138, or by calling 495-7461.

Sobieszek to Discuss Photo Images

Robert A. Sobieszek will give a lecture, "The Composite Image: Photo Montage 1855-1935," at 8pm on Wednesday, February 9, in Jewett Arts Center at Wellesley College.

The lecture, jointly sponsored by the Wellesley College Art Department and the MIT Creative Photography Laboratory, is open to the public free of charge.

Mr. Sobieszek is associate curator and director of the Research Center of the International Museum of Photography at George Eastman House in Rochester, NY. He joined the Eastman House staff in 1968 to assist in organizing its photographic collections.

He has been responsible for several exhibitions at the International Museum of Photography and has assisted in organizing exhibitions at The Philadelphia Museum of Art, The Museum of Modern Art, and the Oakland Museum. Recently he directed the exhibit, "Albert Sands Southworth and Josiah Johnson Hawes: American Daguerreotypists," funded by a grant from the National Endowment for the Arts and designed as the museum's contribution to the nation's Bicentennial celebration.

Mr. Sobieszek is assistant professor of art history at the University of Rochester and on the editorial board of *Image*, a quarterly journal. He has contributed to several books and portfolios, and his articles have appeared in such professional journals as *Album*, *Aperture*, *Art in America* and *Contemporary Photographer*.

MIT Experts Assess Media Influence on 1976 Election

Two members of the MIT Department of Political Science teaching staff—Professor Walter Dean Burnham and Senior Lecturer Edwin Diamond—are among nine media experts whose assessments of news media influence on the 1976 presidential election appear as a question-and-answer panel in the current issue of the newspaper trade journal, *Editor and Publisher*.

Other panelists include three from Syracuse University and one each from Duke University, the University of Michigan, California State University at Long Beach and Indiana University. The three questions posed by *E&P* and the Burnham and Diamond answers were:

1. Do you believe the television debates influenced the voters' decision-making process this year, and if so, how? Have you found any evidence to show that the debates had more influence on one specific group of voters than on others?

Burnham: They certainly were influential. They probably marginally contributed to Carter demonstrating himself as a credible alternative to the incumbent in a close election. They helped Carter stop his slide downhill. There was a greater influence in the Vice Presidential debates, though, where it was clear that Dole was far outclassed. It's hard to evaluate if one group was influenced more than any other.

Diamond: The debates were a reinforcement mechanism, not a conversion mechanism. They didn't change the way people voted but reassured them about the candidates. Since I don't think the debates changed many minds, this makes them less than they've been trumpeted. But in '76 small swings (of voters) were important. Nothing I've heard or read indicates that one group was more swayed by them than others.

2. Newspapers, as opposed to television, have been characterized as the "agenda setting" medium during political campaigns, i.e., their coverage of issues directly determines what issues voters consider important. Do you agree with this theory?

Burnham: In a way yes, and in a way no. Clearly newspapers tend to be read by people who are already politically motivated. They pick up their cues in the in-depth coverage of the *New York Times*, *Washington Post*. These people are a national elite, and they have clout. As far as mass coverage is concerned, it's clear that the electronic media don't cover politics in depth like print. It has to do with the nature of television itself; it doesn't lend itself readily to the longer view.

Diamond: I'd like to suggest that as television and newspapers become less and less serious about more and more things, the public tends to discount what they are atwitter about. If the press ran days and days of close economic analysis, if it hit us with heavy, well-researched stuff, then it could make us think. But when papers day after day go on about Carter's *Playboy* interview, then the public doesn't take them seriously. Newspaper editors are trying to put out papers that look like television—young, "groovy," wide space, picture oriented—and they're not taken seriously. If they're not taken seriously, then how can they set the agenda? They're only setting an agenda for amusement, apathy, alienation, and feelings of inefficacy.

3. Do you believe the media in the swing electoral states such as California, New York and Texas, have a greater responsibility in setting the agenda than the media in states with a smaller electoral vote?

Burnham: I don't think it's the electoral vote as such. After all, Ford won in California but it didn't help him get elected. I think things are rapidly changing. I see the nationalization of political effects that makes the small (electoral) state argument less relevant. In 1976 you had a large number of states with close (voting) outcomes; similarly situated voters—upper class Republicans in the suburbs, black Democrats—voted the same way wherever they were found in the US. It's a different ball game than in the New Deal. The media by their very existence contribute to this situation. Television structures the voter's perception, parties are decaying as cue-givers. People are looking at the personal image of the candidates, and television is exceedingly important in the process. The timing that things appear at 6pm on the evening news is an important part of the candidates' strategy. Newspapers are auxiliaries in the background. Their influence is greatest among the specialized elite, the better educated, more affluent class that has more leisure, those who have clout. The accumulation of their social advantages gives them greater personal individual influence. But all respectable news media have an obligation to deal with the issues of the day no matter where they are located, even if their resources limit them.

Diamond: I don't think it's the press' job to look over its shoulder to see how many electoral votes there are. The press does its best for losing causes. If it looks at electoral votes it won't be into journalism but political advocacy. It's different from journalism, and I think it would be bad journalism.

Tech Talk, February 9, 1977, Page 3

Figure I-6. News Items for Second Experiment.

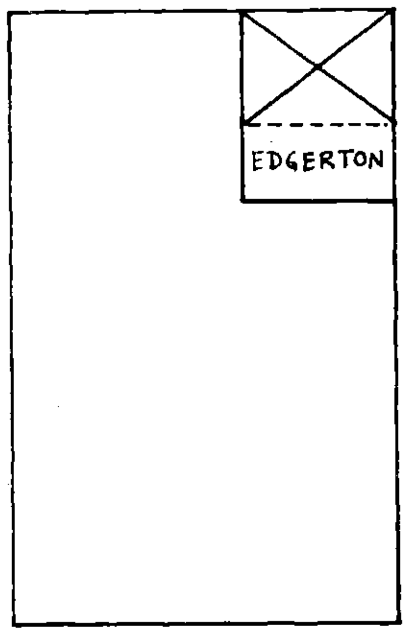


Figure I-7. Partial Layout for Second Experiment.

which would mean two three-column headlines right on top of each other, or I could put the graphic over here [on the right], *which would mean both pictures on the same side of the page. Well, maybe I'll try.... I'll try a five-column headline, and I'll try putting.... I'll try putting the graphic in the lower left corner* [see Figure I-11].

"And that leaves two stories down here, and I can just run.... Well, both of them could be fairly small-point headlines, I guess. In fact, they can still be 24, slightly less than 1/2 inch, say, and I guess I'll put the longer one on top, that's Neudstadt, and that's ... [see Figure I-12].

"Down here, that's Sobieszek ... [see Figure I-13].

"I sort of changed their one-column thing. We [at The Tech] *don't run too*

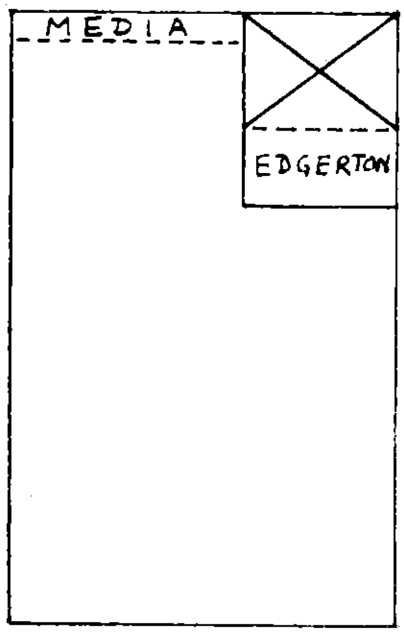


Figure I-8. Partial Layout for Second Experiment.

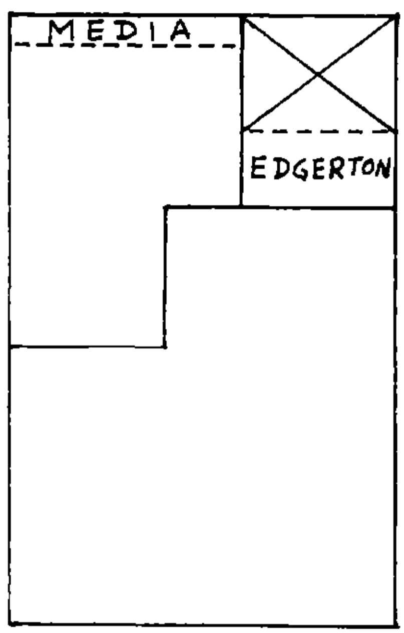


Figure I-9. Partial Layout for Second Experiment.

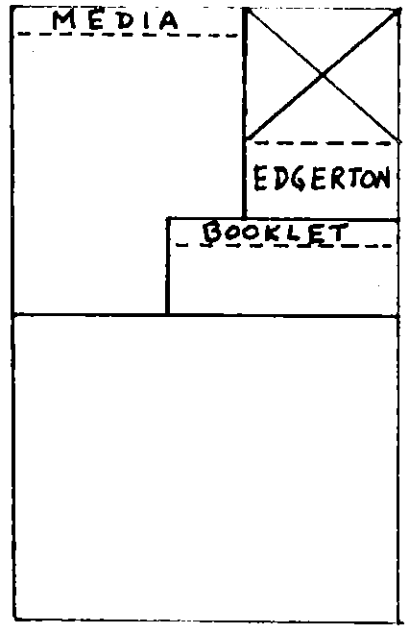


Figure I-10. Partial Layout for Second Experiment.

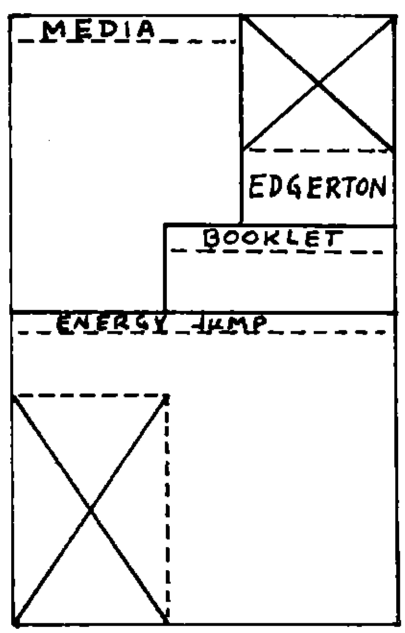


Figure I-11. Partial Layout for Second Experiment.

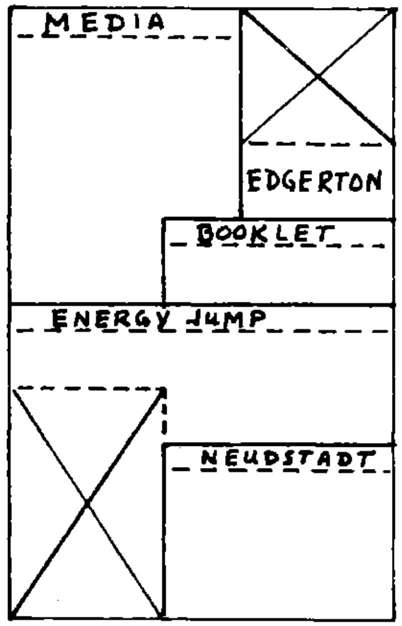


Figure I-12. Partial Layout for Second Experiment.

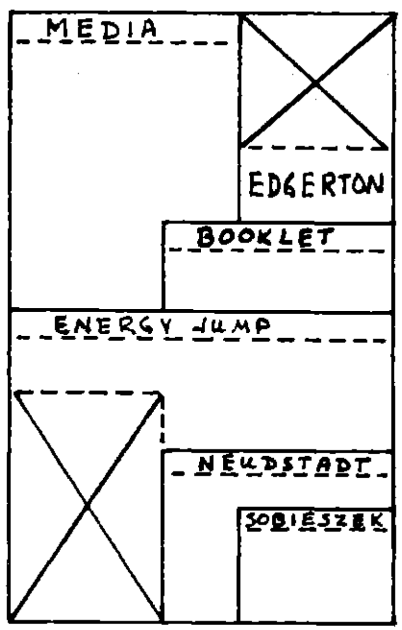


Figure I-13. Final Layout for Second Experiment.

many one-column headlines, except on very tiny jumps."

The elapsed time for this experiment was 8 minutes 5 seconds.

2.3 The Third Experiment

In this experiment, the layout person was to lay out the same news items as in the first experiment, was asked to find a better-looking layout than the first one produced.

"Well, I guess with 31 inches of copy, there's not going to be much way to break it up, very much. All right, I could try.... Well, working from the bottom, I could put the AARP picture, sort of stick it in either the 2nd and 3rd or 3rd and 4th columns and run Engineering around it. I want to keep Engineering in 5 columns, I think, because it'll spread it out more. So, okay, AARP is.... I'll try putting it here in the 2nd and 3rd columns [see Figure I-14].

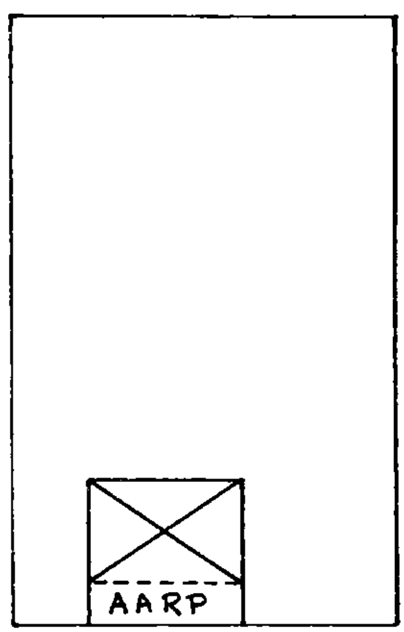


Figure I-14. Partial Layout for Third Experiment.

"And just run Engineering around it [see Figure I-15].

"And headline is about 1/2 inch. That takes up about half the page. That still looks moderately ugly. All right, so, Symposium I could put.... Symposium takes up a total of about 5 inches over 2 columns. Let's see. I could keep the Dancing picture in the upper right corner and put the Symposium story in 3 columns down here, and run copy underneath the picture, which is okay, or.... I really can't put the Symposium story at the top, because then I'd end up with something else underneath it, which would be ugly. So, I think I'll do that. All right, Symposium is ... [see Figure I-16].

"And, see what happens with Dancing. I could either have the Dancing picture in the upper right-hand corner with no headline above it, and have the headline, say, 2 columns by 2 lines over here, or I could have the headline all the way across. I think it would look better all the way across, with the

---

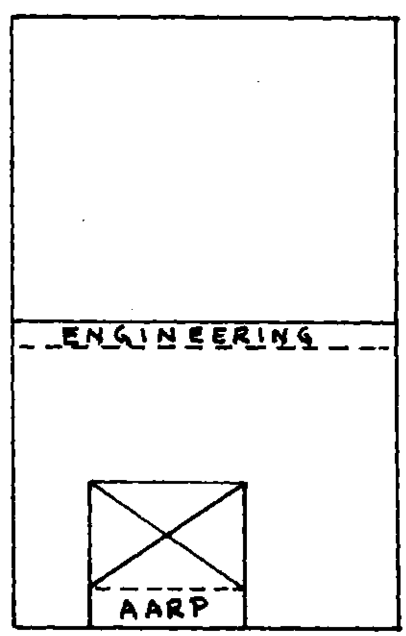


Figure I-15. Partial Layout for Third Experiment.

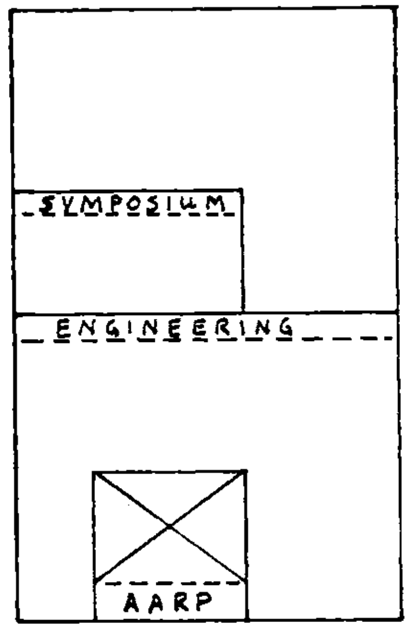


Figure I-16. Partial Layout for Third Experiment.

picture underneath it [see Figure I-17].

"Well, the only problem with this is that there's not very much copy in-between the.... Once you put in the cutline, there's not very much copy in-between these two stories, but it's almost an inch. I don't know, let's see. The top of the page looks too crowded.

"I could have tried putting Symposium down here at the bottom of the page, below the Engineering story, instead. Then I would have had two pictures somehow on the top half of the page.

"If I run the [Dancing] *headline in 2 columns here, that would have moved the picture up a little bit, which would have separated it more from this block over here.*

"It's really hard to make 31 inches of copy look nice."

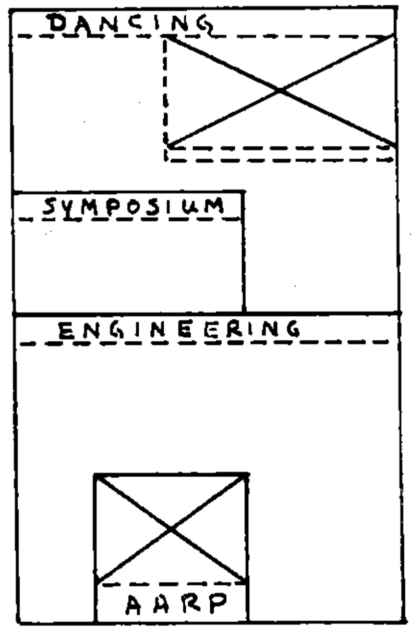


Figure I-17. Final Layout for Third Experiment.

The elapsed time for this experiment was 7 minutes 45 seconds.

3. Conclusions

It would seem that no significant layout rules can be extracted from the performance of this layout person. For example, she works both top-down and bottom-up, depending on the layout problem. As compared with the layout procedure presented in this thesis, the layout person's method seems to be based on her ability to plan ahead, to reason abstractly, and to understand the consequences of her actions in terms of their effects on the possible placements of the remaining items. Instead of working solely with local criteria, she is able to understand the global impact of her decisions.

It should be noted that this layout person does not depend solely on an innate reasoning ability, but also seems to use a number of learned rules, for their esthetic effect. These rules would seem to be primarily of an esthetic nature, and are goal-oriented. For example, she makes an effort to achieve picture balance on the page. It should also be noted that the layouts she produces can be identified as being more in the style of The Tech than the style of Tech Talk (e.g., there are more five-column headlines in The Tech than in Tech Talk, while Tech Talk uses more one-column headlines than The Tech).

Appendix II - Theoretical Considerations

1. Computational Complexity

In a 1972 paper [Karp 72], Karp first identified the set of computationally intractible problems now called the *NP-complete* problems. The best algorithms known for the solution of these combinatorial problems require an amount of time that grows exponentially with the size of their input. This means that there exists some limit on the size of the input, usually a small limit, beyond which any such algorithm becomes impractical because of its astronomical running times. Any NP-complete problem can be reduced to any other in polynomial time, which is less than exponential time. Therefore, if any of the NP-complete problems could be solved by some algorithm in less than exponential time, then all of the NP-complete problems could, by first transforming them to that problem and then solving them using the fast algorithm. This consideration, combined with the large class of problems known to be NP-complete, strongly suggests that it will never be possible to develop an algorithm to solve any NP-complete problem in less than exponential time.

One of the problems shown NP-complete by Karp is the *partition problem*. In this problem, a set of positive integers is given, and the problem is to determine whether the integers can be divided into two subsets whose sums are equal. The best approach known for solving this problem relies basically on generating and testing all possible partitions of the original set into two subsets.

We present the following theorems:

Theorem 1. The news layout problem is NP-complete.[1]

Proof. (This problem is obviously a member of the class of NP problems.)[2] Consider an arbitrary instance of the partition problem; we transform it into an instance of the news layout problem as follows. Each integer in the original set becomes a picture, of height equal to that integer, and of width 1. The newshole has height equal to half the sum of the set of integers, and width 2. Obviously, any layout achieved for the transformed problem represents a solution of the original partition problem. Since this transformation shown can be performed in polynomial time (in particular, it can be done in linear time), this shows that the news layout problem is NP-complete.

Theorem 2. The problem of news layout with a known abstract template is NP-complete.[3]

Proof. (This problem is obviously a member of the class of NP problems.) Consider an instance of the partition problem where there are n integers in the set. We transform the problem as in the previous proof. We also note that there are only n-1 distinct abstract templates which could possibly be applicable to the layout problem, assuming asynchronous layout; these are the abstract templates with two columns, having i items in the first column and n-i items in the second, as i ranges from 1 to n-1. Since n-1 is a polynomial function of the size of the input, we see that the problem of news layout with a known abstract template is NP-complete, since solving it n-1 times would suffice for solving the partition problem.

---

1. Actually, what is being proven NP-complete here is the problem of determining whether any layout exists for a given layout problem; this is necessary because of the "yes-no" nature of the class of NP-complete problems. Obviously, the problem of finding some layout if layouts exist can be expected to be no easier than this problem.
2. Membership in the class of NP problems, also defined by Karp, is a precondition to membership in the class of NP-complete problems.
3. That is, the problem of determining whether a given abstract template is valid for a given layout problem is NP-complete.

These results show that the news layout problem can be considered as computationally intractible. The best known algorithm for solving the problem depends on an exhaustive search of all possible concrete templates. This algorithm is applicable up to about three news items, beyond which the amount of time required is excessive.

One approach which is often used when dealing with computationally intractible problems is to solve some different problem, related to the original problem but computationally simpler. The approach taken in this thesis is simply to try to find some layouts based on the given layout problem. If layouts are found, then that shows that layouts exist; if none are found, then either none exist or the algorithm was unlucky. Also, instead of trying to find the best of all possible layouts, the approach will be only to try to find good layouts. For the mix of layout problems commonly found, this approach, as incorporated into the layout algorithm presented in this thesis, seems adequate, in that it will typically find some layouts in a reasonable amount of time.

2. Number of Templates

The number of possible templates is very large. To count their number, a program has been written which generates all distinct legal synchronous abstract templates. The templates generated are all *reduced* templates, in that no column is identical with an adjacent column, nor any row with an adjacent row. Table II-I shows the number of possible templates for up to five items, broken down by the number of items and by the number of columns in the reduced template. As can be seen, the number of abstract templates grows quite rapidly with the number of items. The number of concrete templates grows even more rapidly. There are n! possible ways that an abstract template with n items can be marked. The number of ways an abstract template with i columns can be widened into a newshole with j columns is C(j-1,i-1). The number of ways an abstract template can be fitted into a

Table II-I. Number of Templates.

	items = 1	2	3	4	5
columns = 1	1	1	1	1	1
2		3	12	42	141
3		1	29	280	2185
4			24	716	12163
5			7	874	33430
6				519	50978
7				121	44142
8					20382
9					3907
sum =	1	5	73	2553	167329

newshole depends on the shape of the template items, as well as on numerical factors. It can easily be seen that exhaustive searching of all possible concrete templates is generally impractical.

Figures II-1 through II-9 show the set of possible abstract templates for up to three items, broken down by the number of template columns. It can be seen that certain of these templates look noticably worse than others. The best general rule that has been developed to explain this esthetic difference is the convexity rule presented in Section 7.2; most of the worst-looking templates violate the convexity rule. This rule, when incorporated into the

Figure II-1. All Possible One-Column Abstract Templates for One Item.

Figure II-2. All Possible One-Column Abstract Templates for Two Items.

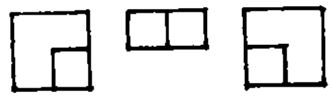


Figure II-3. All Possible Two-Column Abstract Templates for Two Items.

Figure II-4. All Possible Three-Column Abstract Templates for Two Items.

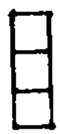


Figure II-5. All Possible One-Column Abstract Templates for Three Items.

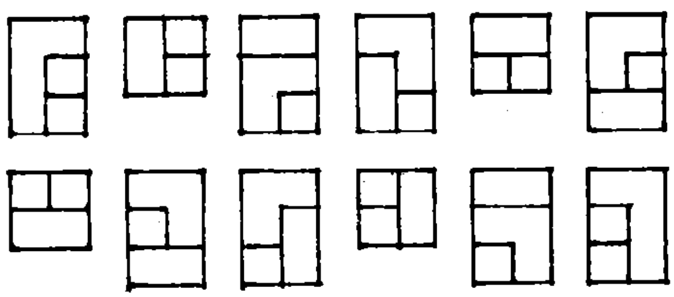


Figure II-6. All Possible Two-Column Abstract Templates for Three Items.

layout program as an esthetic rule, helps the program to avoid the generation of such templates.

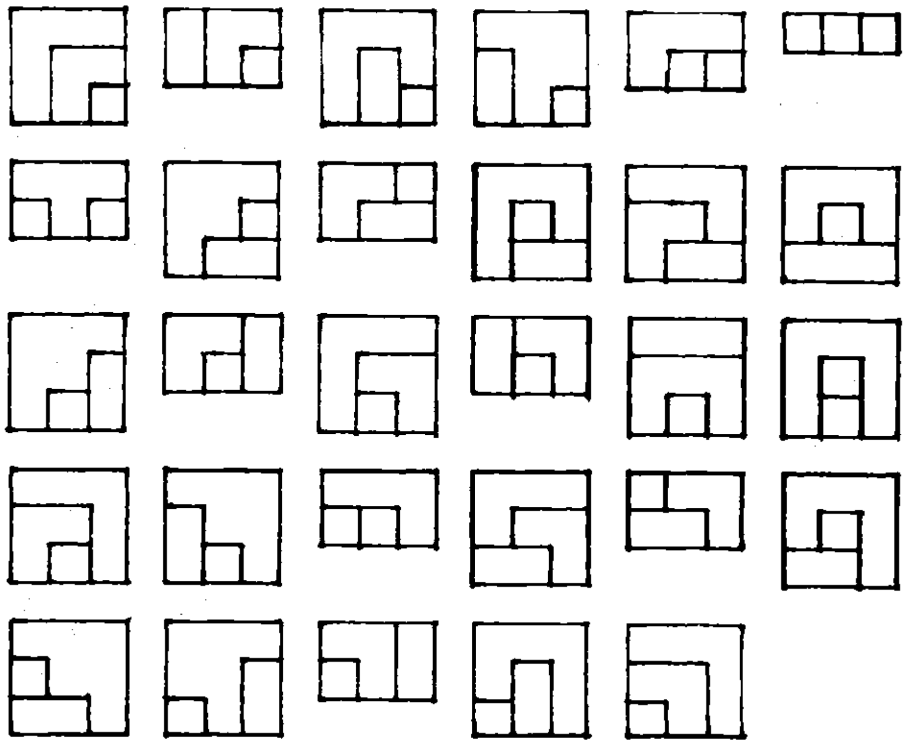


Figure II-7. All Possible Three-Column Abstract Templates for Three Items.

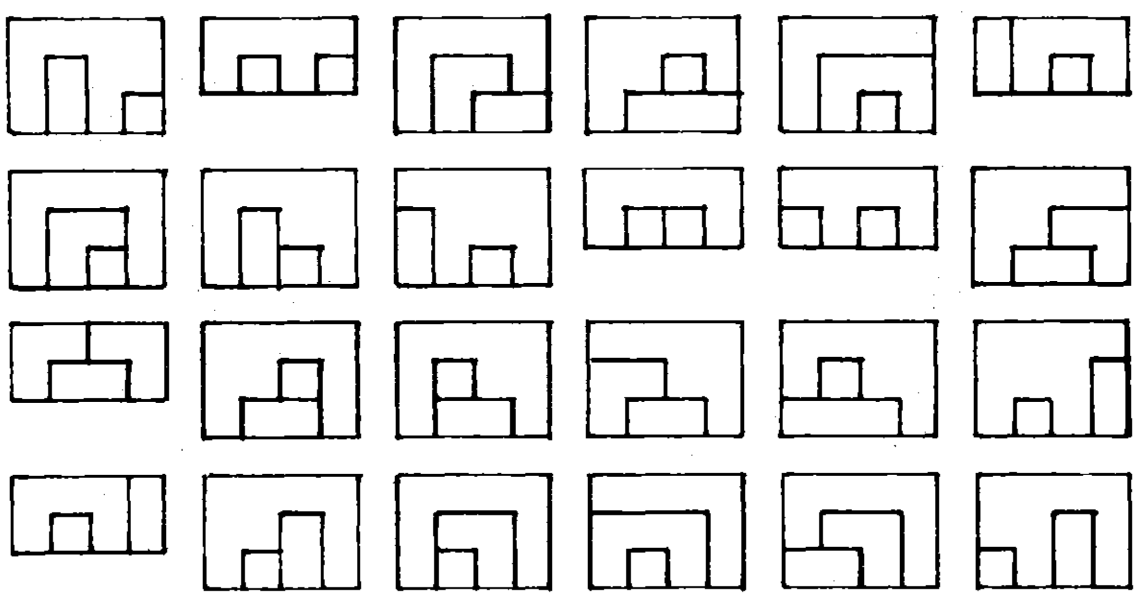


Figure II-8. All Possible Four-Column Abstract Templates for Three Items.

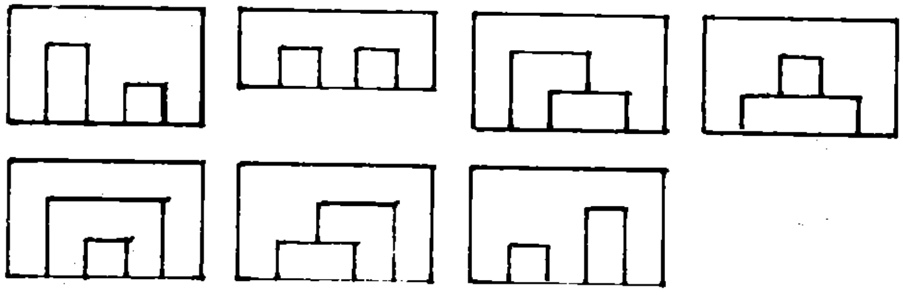


Figure II-9. All Possible Five-Column Abstract Templates for Three Items.

3. Template Satisfiability

3.1 General Observations

Some templates are more easily satisfied than others. That is, the probability that a given set of news items will satisfy the template is greater for some templates than for others. We note, for example, that a template such as the one shown in Figure II-10 is applicable for any pair of stories,[4] while a template such as the one shown in Figure II-11 is applicable in fewer

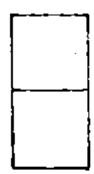


Figure II-10. Universally Applicable Template.

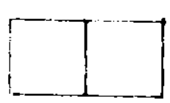


Figure II-11. Less Applicable Template.

4. This section considers just the areas of stories, and ignores their internal structure. It also ignores pictures. All templates considered in this section are abstract templates.

cases, because column boundaries are discrete, leading to a limited number of possible positions for the boundary between the two stories.

Figure II-12 shows the set of item sizes which will satisfy the template in Figure II-11, based on a six-column page, assuming that the sum of the item sizes must be at least 90% of the area of the rectangular newshole. Assuming a uniform distribution of item sizes, there is a 26.3% probability that a random pair of items obeying the overall size restriction will satisfy this template.

We can expect that the template shown in Figure II-13 will have a higher probability of satisying a given pair of items, since the item in the corner is not restricted to occupy all of the area in its columns. Figure II-14 shows the set of item sizes which satisfy this template. This template has a 87.7% probability of being satisfied with an arbitrary marking. Similarly, the template in Figure II-15 has a 70.2% probability of being satisfied with an arbitrary marking.

3.2 Residue

It can be seen from these graphs that the probability of satisfaction of a template grows less as the sum of the item sizes approaches the size of the newshole, forcing the amount of residue to zero. For the template in Figure II-11, the probability of satisfaction drops to zero as the limit, while the probability of satisfaction for the template in Figure II-13 drops to 83.3%. Conversely, the probability of satisfaction can be seen to grow monotonically as the proportion of residue grows.

This predicts that layout problems with a low proportion of residue will be harder to lay out, since fewer templates will be satisfied by them, while layout problems with a high proportion of residue will be easier to lay out.

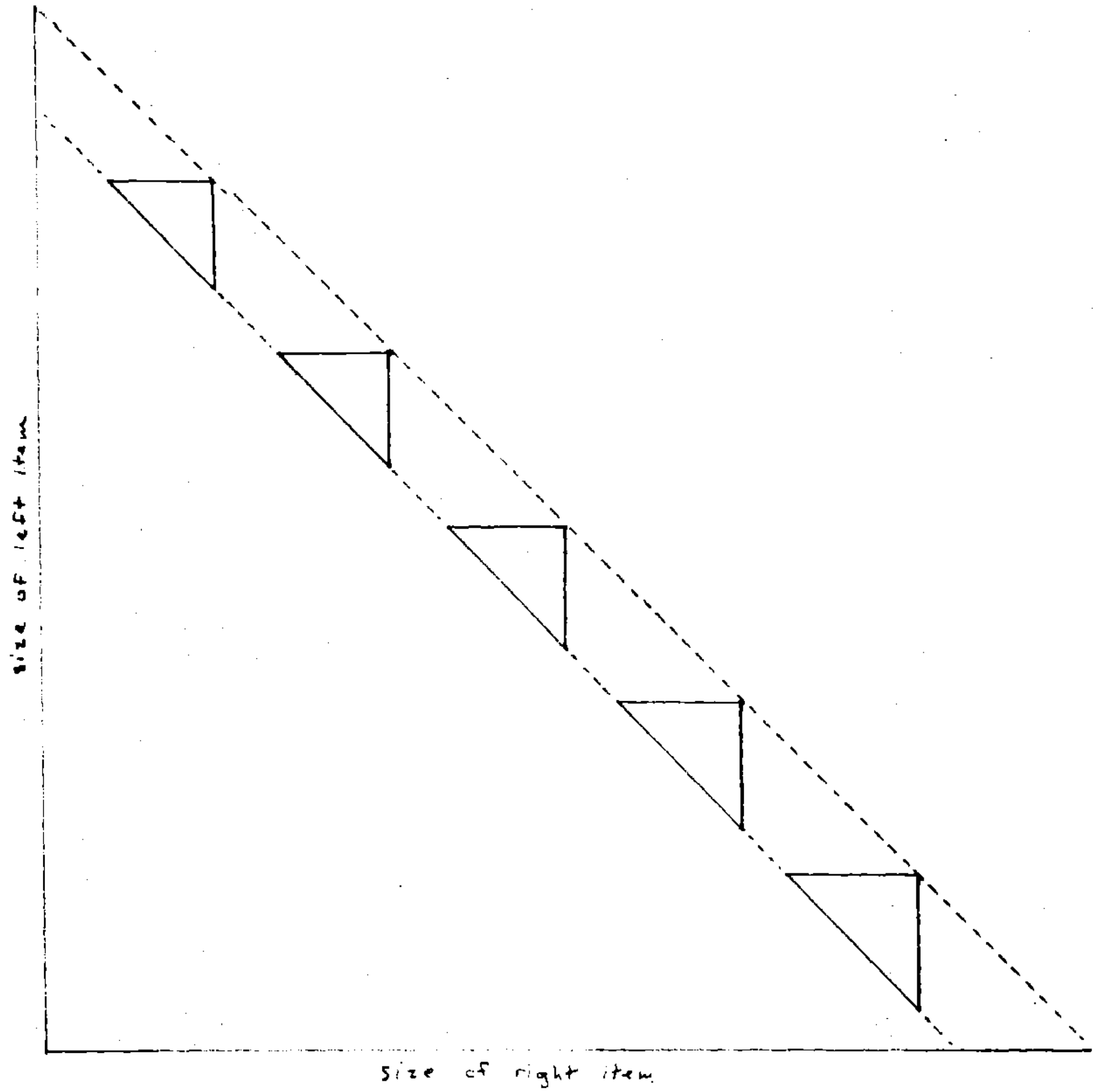


Figure II-12. Item Sizes Satisfying Template.

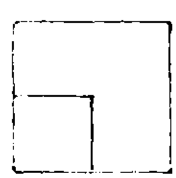


Figure II-13. More Applicable Template.

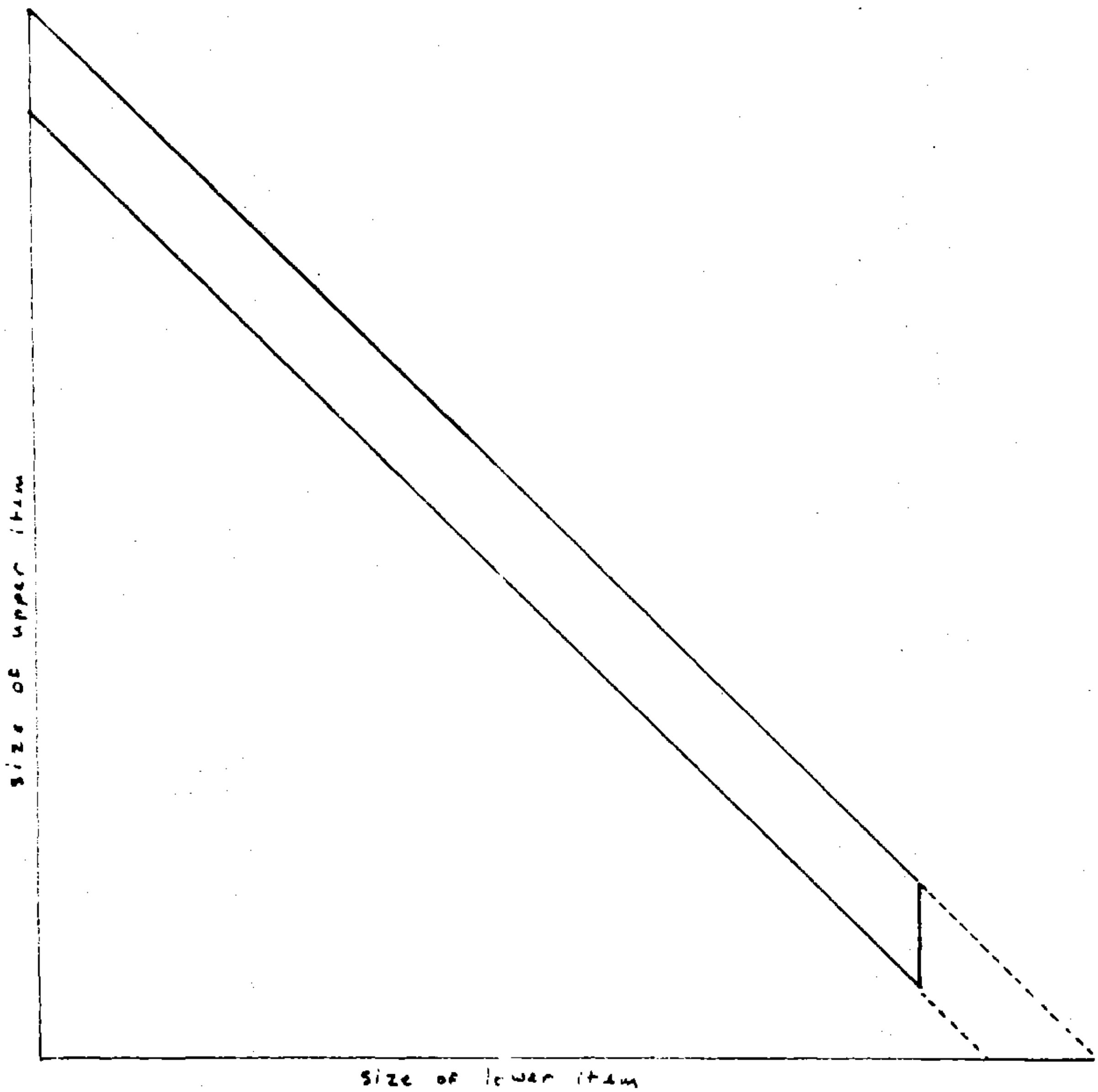


Figure II-14. Item Sizes Satisfying Template.

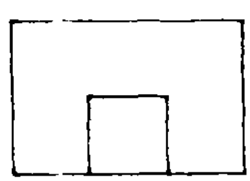


Figure II-15. Related Template.

3.3 Columns

It can also be seen from these graphs that the probability of satisfaction of an abstract template grows greater with the number of columns on the page. If the template in Figure II-11 were to be laid out in a newshole with eight columns, the probability of success would climb to 35.1%, an increase proportional in this case to the increase in the number of columns.

A limiting case would exist if there were an infinite number of columns. This would have the space in the newshole arbitrarily divisible in the horizontal direction as well as the vertical. It is conjectured that it might be possible to lay out any legal unwidened template in such a case.

3.4 Restrictions

It can be seen that placing restrictions on acceptable layouts decreases the number of layouts possible. Consider, for example, the template in Figure II-11. Assuming the same conditions as before, if it is additionally required that the maximum leading factor for either of the stories be 10%, this decreases the probability of satisfaction to 8.5%, as shown in Figure II-16.

In general, reducing the proportion of valid templates can be expected to make layouts harder to find.

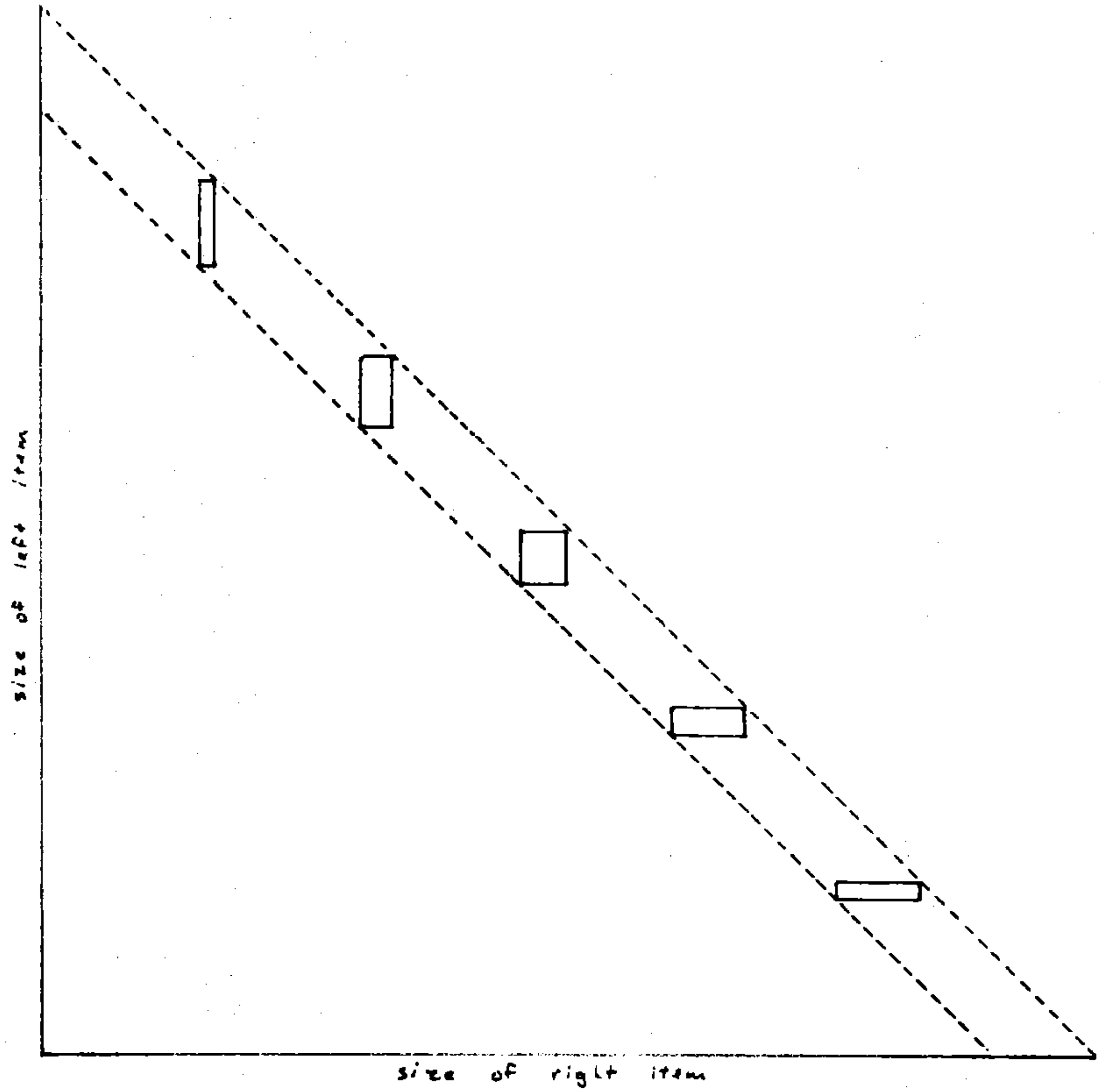


Figure II-16. Item Sizes Satisfying Template with Leading Restriction.

Appendix III - Comparison with Longtin's Work

1. Background

In his 1972 thesis [Longtin 72], Longtin presented an approach to news layout in which layouts were to be built incrementally, as illustrated in Figure III-1. The insertion of an item could entail the modification of the placements of the previously inserted items, as necessary.

Implementations of Longtin's algorithm [Polansky 74] showed this approach to be inadequate. Impasses in layout occurred very often, as illustrated in Figure III-2. Although the implementation included a number of heuristics to

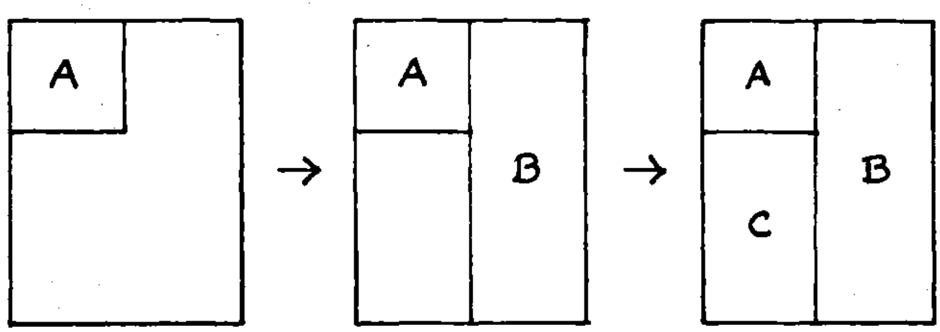


Figure III-1. Example of Successful Layout with Longtin's Algorithm.

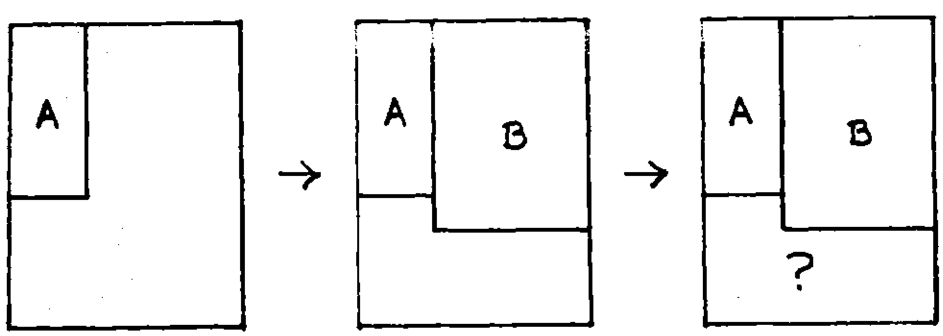


Figure III-2. Example of Unsuccessful Layout with Longtin's Algorithm.

modify the partial layouts to sidestep such situations, none were found that were strong enough to allow the layout of more than about three items. Beyond that point, no layout could typically be found because of the nearly inevitable occurrence of dead ends.

2. Analysis

Longtin's approach and the approach taken in this thesis are similar, in that they both involve the incremental placement of the items to be laid out. The difference lies in the fact that Longtin's approach was to build *layouts* incrementally, while the approach in this thesis is to build *templates* incrementally.

Implementations of Longtin's algorithm were attempted both for bottom-up layout and for top-down layout. As was shown in Section 5.2.3, template generation in the bottom-up case can be considered as being essentially the same as layout generation, since the incremental solution of the layout equations possible in bottom-up layout can be immediately applied to the partial templates to produce partial layouts, which themselves grow incrementally. Therefore, Longtin's approach and the approach in this thesis are very similar in the case of bottom-up layout. As was shown in Section 5.4, however, bottom-up layout is inefficient for all but the smallest layout problems, because of the high probability of dead ends. Longtin's algorithm, upon encountering a dead end, would simply halt without producing a layout. In this way, it resembles a breadth-1 search which is restricted to operate in only one pass. As shown in Section 6.3, such a search has only a small probability of success.

In the case of top-down layout, the methods are more dissimilar. Longtin's method, since it produces layouts incrementally, must specify the exact shape and dimensions of an item as soon as it is inserted into the partial layout.

Although these may be changed somewhat later, the inclusion of numerical dimensions can overly restrict top-down layout, as shown in Section 5.3.3. Since the approach presented in this thesis is to generate templates incrementally, where neither the shapes of items nor their actual dimensions are determined until the information for doing so is available, it is more successful in avoiding dead ends.

Longtin's algorithm does not seem to include the concept of a grammar whereby legal item shapes can be distinguished from illegal ones. The implementation of his approach tend to produce layouts with extremely bad item shapes, especially in the top-down case.

As was noted, Longtin's algorithm restricted itself to finding a layout in one pass, taking the perceived best move at each step. The approach in this thesis allows for multiple passes through the layout tree, picking moves probabilistically; this vastly increases the potential for successful layout.

Appendix IV - Comparison with Kan's Work

1. Background

In his 1976 thesis [Kan 76], Kan presented a different approach to news layout, based on his concept of a *template* as an abstraction of a layout. The problem of news layout was reduced to first finding an appropriate template, and then producing a layout based on that template. Templates were to come from a *template library,* a large stored collection of abstract templates. For a given layout problem, the template library would be searched for templates with the same configuration of stories, independent pictures, and dependent pictures, and layouts would be attempted for those templates matching the layout problem. The template library would be sufficiently large for there to be a high probability, for an arbitrary layout problem, of producing a successful layout from at least one template in the library.

2. An Experiment

The size needed for a template library is crucial in determining how practical such an approach would be. Kan estimated that about 500 templates would be needed, but admitted that this number was uncertain. To better determine the size necessary for a template library, an experiment was designed in which a small prototype template library with only a few dozen abstract templates was to be implemented. Although the performance of this small library would be expected to be less than the desired level, it was believed that it would be possible, based on the performance level thereby achieved, to estimate some lower bound on the template library size needed to satisfy 95% of all layout problems, where 95% was chosen as a reasonable performance level for a template library. The difference between the number of templates needed for 95% performance and the number needed, for example, for 90% performance can be expected to be less than a third.

In the implementation of this experiment, the templates[1] in the library were all based on layouts appearing on Page 3 of the daily edition of the Boston Globe. This page was chosen because the layouts on the page are stylistically consistent over time, making it an ideal candidate for use with a template library. Additionally, Page 3 never contains jumps or ads, which could cause experimental complications. Templates from 23 pages were used to form the template library.

To test this template library, 23 layout problems based on the same 23 pages were used. Each layout problem was tried with each template, except, of course, for the template that was used for its layout in the Globe, since such a combination would be guaranteed to succeed. Of the 506 combinations possible, 14 were found to have the same story-picture configuration in both the layout problem and the template.[2]

In each layout problem for which a layout was attempted, the height of the newshole was artificially set to make the space available be 1.05 times the sum of the areas of the news items, to allow for experimental consistency. For the 14 problem-template matches, no layouts were achieved. This suggests an overall success rate of less than one success per 506 problem-templates, which would require a template library of more than 1500 templates in order to have a 95% success rate for layout problems.

---

1. The template representation scheme used in this thesis and that used by Kan are quite dissimilar, because of their different intended uses. However, it is believed that the scheme used in this implementation (basically supplying abstract templates asynchronous in both the horizontal and vertical directions) can, if anything, surpass Kan's in its descriptive power.
2. These matches were based on three equivalence classes: one of three pages each having two stories with no pictures, two stories with one picture, and one independent picture; one of three pages each having four stories with no pictures, one story with one picture, and one independent picture; and one of two pages each having six stories with no pictures, one story with one picture, and two independent pictures.

The experiment was rerun with the space available equal to 1.10 times the sum of the areas of the news items, with the expectation that this might produce more layouts. In this case, there were layouts achieved for three of the problem-template matches, suggesting a template library size of about 500, as estimated by Kan.

3. Analysis

Obviously, these figures have a high degree of uncertainty, which could be remedied by an increase in the experimental template library size. It was decided, though, that a template library size sufficient to raise the number of problem-template matches from the 14 achieved in this experiment to a more significant number would be sufficiently great to represent a major undertaking. It seems that it might be necessary for the experimental template library to become fairly near the size needed for 95% performance for a meaningful estimate of that size to be produced.

In the absence of hard quantative results, though, it is possible to argue quantitatively against a template library approach. Consider a layout problem with 7 stories and no pictures. There would be 7!, or 5040, possible ways to mark each appropriate abstract template from the template library in the process of turning it into a concrete template;[3] this is a large number of concrete templates to test for validity. There are methods by which such tests can be done more efficiently than on a simple one-by-one basis; these methods reduce the number of individual tests somewhat, but the number of tests can still be expected to grow exponentially with the number of items.

Such testing would have to be done, on the average, for many of the

3. Although Kan did not use the same distinction between abstract templates and concrete templates as is used in this thesis, he did use a process of marking during layout using an abstract template.

templates for seven stories and no pictures before a layout would be achieved. It seems likely that this number of templates would have to be greater than the equivalent number for six stories and no pictures. This suggests that the amount of time required might grow excessively great for complex layout problems.

Moreover, there would have to be some set of templates stored for most story-picture configurations possible. Since the number of configurations is quite large, this suggests that the size of a template library might be exceptionally great. On the other hand, of course, some configurations are more probable than others. The most probable of all are those with a small number of news items, such as are found on most inside pages. A template library which was biased toward the simpler layout problems might be able to achieve a high proportion of layouts, averaged over all of the pages in the newspaper. The above arguments, though, cast doubt on its ability to handle complex layout problems very rapidly or in very many cases.

It is believed that the layout algorithm given in this thesis will perform better than a template-library scheme. The approach in this thesis does not entail storing a large set of templates, since all possible templates can be mechanically generated. While Kan was forced to use an exhaustive search procedure to search through the set of appropriate abstract templates and to determine whether a given abstract template could produce a layout, more effective search procedures are possible in the framework given in this thesis.

References

Bracken, Diane, "Headline Specifications Determination for Computer-Aided News Layout", S.B. Thesis, Department of Electrical Engineering and Computer Science, M.I.T., Cambridge, Mass., 1975.

Elkin, R. M., "An Evaluation of Computer-Assisted Display-Ads Layout in an Operating Newspaper Environment", Electronic Systems Laboratory Report ESL-R-553, M.I.T., Cambridge, Mass., June 1974.

Kan, Hsin-Kuo, "Computer-Assisted Display-Ads Layout for Newspapers", Electronic Systems Laboratory Report ESL-R-501, M.I.T., Cambridge, Mass., June 1973.

---, "A Computerized Template-Driven News-Layout System for Newspapers", Ph.D. Thesis, Department of Electrical Engineering and Computer Science, M.I.T., Cambridge, Mass., 1976.

Karp, R. M., "Reducibility Among Combinatorial Problems", in *Complexity of Computer Computations*, Plenum Press, N.Y., 1972.

Longtin, Brent, "A Computer Assisted News Layout System", S.M. and E.E. Thesis, Department of Electrical Engineering, M.I.T., Cambridge, Mass., 1972.

Newborn, Monroe, *Computer Chess*, Academic Press, N.Y., 1975.

Newspaper System Development Group, "Procurement Specification for a Composition and Makeup Terminal System", 1973.

Polansky, Robert, "Documentation of News-Layout Program", Electronic Systems Laboratory Memorandum NEWSI-20, Cambridge, Mass., May 1974.

Rabin, Michael, "Probabilistic Algorithms", in *Proceedings of the Symposium on New Directions and Recent Results in Algorithms and Complexity*, Academic Press, N.Y., 1976.

Reintjes, J. Francis, et al, *Computer-Assisted Layout of Newspapers*, Electronic Systems Laboratory Report ESL-R-778, M.I.T., Cambridge, Mass., 1977.

Shannon, Claude, "Programming a Digital Computer for Playing Chess", *Philosophy*, vol. 41, pp. 356-375, 1950.