73

Fishpaper: Automatic Personalized Newspaper Layout

by

Benjamin Durant Schoon

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Bachelor of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 18, 1994

Author ....[illegible]
Electrical Engineering and Computer Science
Date Submitted

Certified by ....
Pascal Chesnais
Research Specialist, Media Laboratory
Thesis Supervisor

Accepted by ....
Leonard A. Gould
Chairman, Department Committee on Undergraduate Theses

Automatic Personalized Newspaper Layout

by

Benjamin Durant Schoon

Submitted to the Department of Electrical Engineering and Computer Science on May 20, 1994, in partial fulfillment of the requirements for the degree of Bachelor of Science.

Abstract

Flexible newspaper style templates are used to generate personalized PostScript newspaper layout. The program `fishpaper` was designed to print personal newspapers for users of MIT's Athena-wide Freshman Fishwrap news project. `fishpaper` relies on Fishwrap for news content filtering and organization. PostScript utilities and news objects in were used from the *Newskit* library. Style templates allow `fishpaper` to print multi-column page layouts which resemble modern newspapers. When expected types of news items (articles, graphics, or schedules) are not available, contingency plans are used to print alternative types of content .

Thesis Supervisor: Pascal Chesnais
Title: Research Specialist

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The role of newspaper layout is quickly changing. With the advent of a national information infrastructure, news providers will be able to rapidly deliver more news to the home or office than is currently possible with printing press technology. Anticipating this drastic change in the newspaper industry, some researchers are concerned with the presentation of electronic news on computer monitors. However, printed newspapers are often more convenient to carry and read than portable computers. Until portable news reading devices become pervasive, "hardcopy," i.e. paper-printed, newspapers will be the primary form of transportable, text-based news.

In The Future of Paper, Jaakko Pöyry asserts that the continued use of paper as a printing medium will be justified by "... our need for touching ... the material traces of our symbolic creations, and for having them displayed in 'frozen' form" even in an era of increased use of electronic dissemination of textual and graphical information. However, to some, the ability to dispose of a paper newspaper guiltlessly is of more importance than the tactile interaction with its texture and intuitive feel (manually turning back a page).

Problems of newspaper layout are addressed by several groups working within the "News In the Future Consortium" at the MIT Media Laboratory. The Information and Entertainment Section at the Media Lab focuses its efforts on the personalization of news. A goal of this group is to provide a personal newspaper layout tailored to the reader's choice of format and presentation.

In the current model of bringing news to the reader, a major newspaper firm, (1) collects and filters news, (2) creates a single layout, and (3) delivers the same newspaper to many people. The only personalization occurs when special regional sections are printed for specific communities or suburbs of larger cities[2]. Advertisement inserts are included for many newspapers by the paperboy at the local distribution level. The new model, which our group is investigating, follows the same process according to the *individual* tastes and preferences of each reader.

The Freshman Fishwrap project[7][1] was created to provide a personalized news service using the NCSA's Mosaic program on Athena, MIT's computing environment. Transmission of articles can occur in many ways. One utility allows students to electronically mail articles to one another. The program `fishpaper` was written to print personal newspapers on the public laser printers in all the Athena clusters. `Fishpaper` uses personally selected articles from Fishwrap to print personal newspapers.

Mosaic outputs both text and graphics in a single-column, linear fashion (top to bottom) on a page. The `fishpaper` program generates a newspaper layout which is intended to resemble traditionally printed newspaper (see Appendix D.4). `fishpaper` was designed to accommodate the student on the run. During the January IAP[2] period at MIT many activities and events are offered. Daily schedules, intended to show which events are available and at what time, can be printed by `fishpaper`. When the day is over, the paper can simply be recycled.

The purpose of any automated layout system is to cover pages with text and graphics according to pre-specified stylistic guidelines[3]. The described approach to automatic layout is first to pre-define how each page should look, and second to allow each article growth or shrinkage on each page so it will fit in the allotted space. This is a heuristic

1. Fishwrap is taken from a secondary function of newspapers, particularly in England where fish and chips are often served in folded newspapers. This reference is somewhat esoteric and eludes most of the freshman asked informally.
2. IAP is the Independent Activities Period at MIT. The month of January is reserved for classes and activities offered by students, faculty and MIT community members.
3. The Standardized General Markup Language (SGML) is one type of page description common in the newspaper industry.

approach which incrementally adjusts each page. That is, changes on a page affect only that page and following pages. No attempt is made to perform global optimizations over an entire newspaper (i.e., affecting previous pages *as well as* following pages such as in the Tex formatting language by Donald Knuth).

The dual purpose of this thesis is to consider both the programming issues of automatic newspaper layout and the newspaper issues addressing the aesthetics of layout and presentation. This thesis documents the rudimentary progress made in each of these directions. A discussion of Newspaper issues appears in the Results, Conclusions and Recommendations Chapters. Descriptions and examples of program file formats can be found in Appendix A. Appendix A.2 includes guidelines for creating style templates. Programming issues are covered in Methodology, Conclusions, Recommendations and Appendix B (Guide to Using Layout Frames). A description of the *how* flexible templates work appears in Appendix C.

A visual history of PostScript formatted newspapers in my group is presented in Appendix D. As a result of the many experiments with automatic layout found in Appendix D, the Recommendations Chapter provides a very good approach for modeling and solving human-unassisted newspaper layout.

Chapter 2

Methodology

2.1 Fishpaper Overview

This Chapter addresses many of the programming issues concerning printed newspaper generation. The program used to create a printable PostScript file is called `fishpaper`, since it is used to generate a hardcopy news*paper* for the *Fish*wrap project. Two sets of software tools were used for writing `fishpaper`: Fishwrap's personalization of news content and Newskit's news item abstractions and PostScript formatting capabilities.

This chapter is divided into sections covering the news source, the layout information, and the newspaper object which uses the source and the layout information for rendering a PostScript file. Articles are printed by rendering visual elements such as headlines, article paragraphs and/or related graphics. Left over article text is saved and printed on later pages.

The "Garden" refers a subgroup of the Information and Entertainment Section at the Media Lab. The programs in Figure 2.1 are all written by members of this subgroup.

The next page displays two illustrations of `fishpaper`'s placement in relation to Glue[4] and Fishwrap. Figure 2.1 is an illustration of the modularity of news utilities. Glue provides a simplified interface to Fishwrap. The details of user modelling, knowledge representation for articles, and News Service are all hidden from front end applications. Fishwrap has a single channel of interaction with all these activities through Glue.

The `fishpaper` program is designated by HardCopyFishwrap in the Front End Application cloud in Figure 2.1. The pipeline in Figure 2.2 shows the data or communication at each intermediate step in the layout process.

The Garden's model for gathering and filtering news is shown in the diagram below:

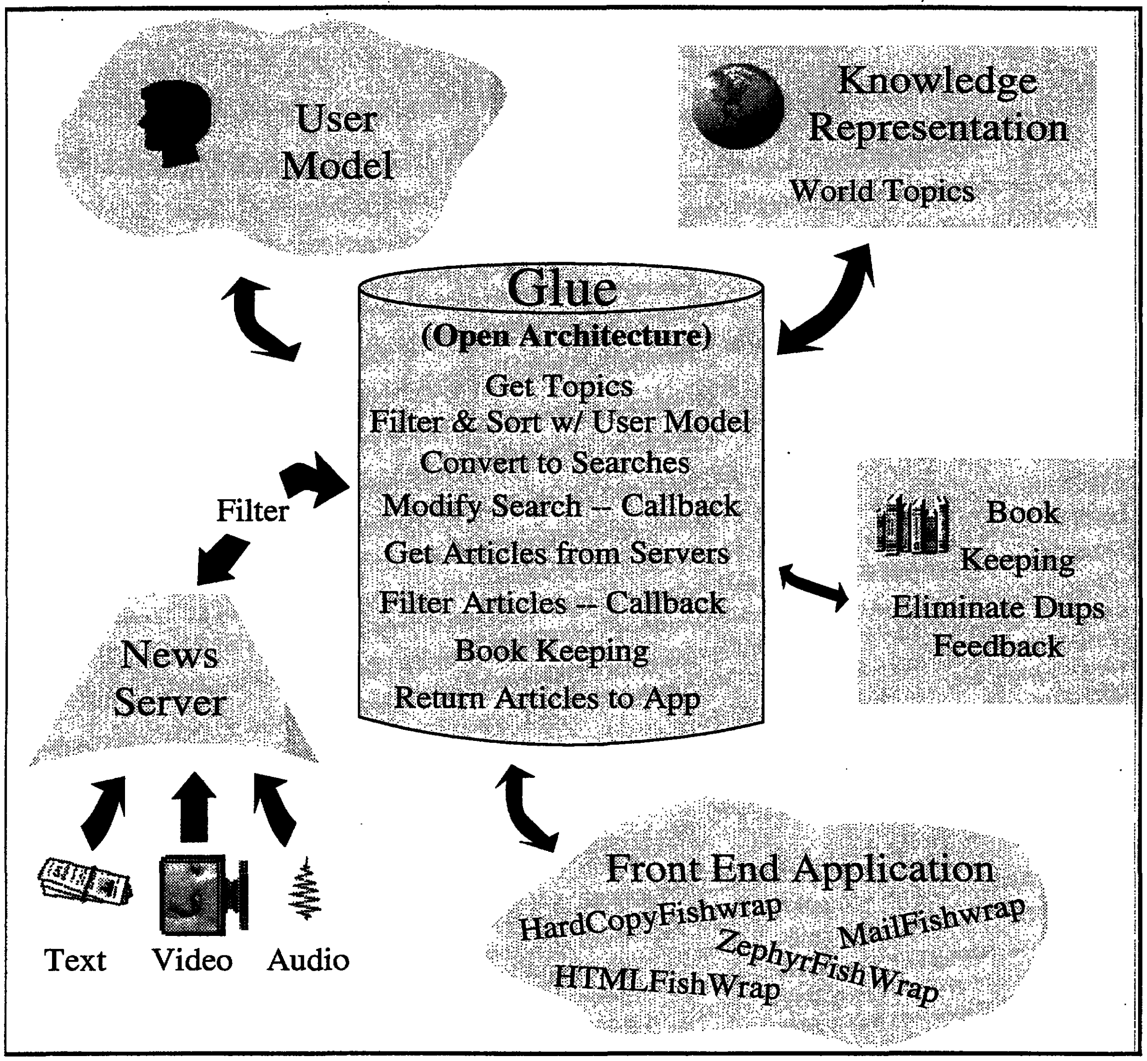


Figure 2.1: Glue, an interface to news related software tools in the Garden

This model is used for the Freshman Fishwrap Project. The pipeline in Figure 2.2 is an illustration of the process of printing a PostScript newspaper using, Glue, Fishwrap and `fishpaper`.

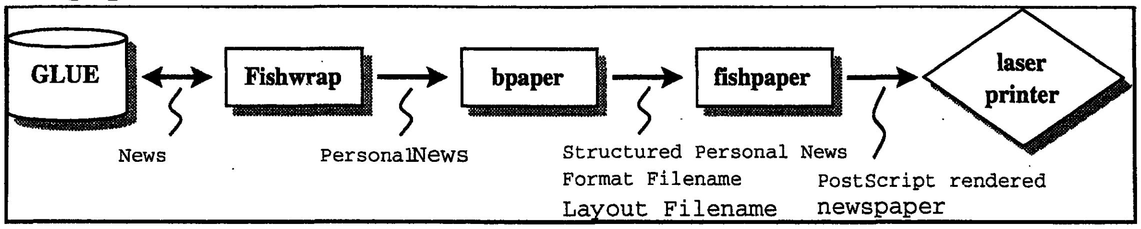


Figure 2.2: Pipeline to Generate Personalized PostScript Newspapers

The *Newskit*[5] libraries for PostScript font and news abstractions were used as a base platform for writing `fishpaper`. The basic model of news item parsing, creation, and output functionality was adopted from *Newskit*. New subclasses of existing news items were created for newspapers, articles and schedules.

2.2 General Programming Concerns

Object oriented programs often follow the same approach: (1) An object is created with initialization information, (2) methods of the object are invoked, which often call other object methods, and (3) this invocation either alters or produces some data. The very simple idea of creating an object and instructing it to do something is used repeatedly throughout the execution of `fishpaper`.

Following the pattern described above, `fishpaper` simply creates an object of type `layout_ps_newspaper_fishwrap` and instructs it to render a postscript file. Of course, the object must be initialized with enough information to perform newspaper layout. The following sections are a description of what information is used for the initialization as well as the details of the layout process.

Which objects perform which tasks is of critical importance for the design if the program is to be modified in the future. The class organization should be flexible enough to replace objects with others of similar functionality. This model of programming allows tasks to execute by calling subtasks. This is of general importance because (1) each task should appear simple, (2) inheritance allows reuse of code for similar objects, and (3) subtasks can be replaced with different implementations which either produce the same supertask more efficiently, or to perform an entirely new supertask. Future improvements are considered in the Conclusions Chapter and are described in the Recommendations.

2.3 News Source

`fishpaper` is supplied with the name of a dtype[6] file[1] to as an argument. Dtypes unify many common data types which can be used on a diverse set of computer architectures. This file contains: the news articles and the file names of any graphics to include. It also contains the layout file and format file which are to be used to layout the newspaper. The file is not guaranteed to be in an acceptable format[2], so the input is first read into a dtype and then reformatted.

The reformatted dtype is then parsed as a `news_dtype_source` and is used to create a `news_bpaper_source`. The news source supplies the content of the newspaper to be generated.

2.4 Layout Paper

Besides the news source, the `layout_ps_newspaper_fishwrap`, requires a layout manager for performing layout. All relevant information for layout is encapsulated in the layout manager. Page margins, inter-column distances and masthead[3] heights are among the modifiable properties for determining layout. These variables are set explicitly in the `fishpaper` program source code.

The `user_hardcopy.dtype` file which was read as input should contain the name of the file specifying a newspaper template[4]. Templates contain page descriptions for a newspaper. The `layout_nonterminal_bpaper_ps_fishwrap` performs the page layout for a newspaper.

2.5 Newspaper

`Fishpaper` creates a `layout_ps_newspaper_fishwrap`. The news source of the

1. The file format for this input file is described in Appendix A.1
2. See Appendix A for file format specifications.
3. Mastheads are the banners at the top of the sections of a newspaper.
4. Flexible templates for newspaper layout are discussed at length in Appendix B.

`layout_ps_newspaper_fishwrap` is set to the `news_bpaper_source` and its layout manager is set to the `layout_nonterminal_bpaper_ps_fishwrap`. The data of the source is parsed immediately, but the layout file name is set without parsing the template.

When the news source is set, a newspaper section is created for every section in the input file. The data in the dtype file can contain either articles, images, or schedules. These fall into one of six topics per section. Table 2.1 shows the six topic categories[1].

Topics in each section
Home Town News
Special Topic News
AM activities On Campus
PM activities On Campus
AM activities Off Campus
PM activities Off Campus

Table 2.1: Topic Categories for `fishpaper`

The categories are used for assigning locations on the newspaper pages, so that the newspaper template can place related articles and graphics together (see Figure "Front Page Template for Fishpaper" on page 15). These categories were chosen for printing personal newspapers during the Independent Activities Period.

Schedules contain information about the time, the event name, the location and description of an event. Events are categorized as being AM or PM, and On or Off Campus. The design of the template was chosen so that a reader could fold the page into quar-

1. The types of allowable information (articles, graphics or schedules) is determined by which C++ classes exist in *Newskit*. For this project, schedules were added to the existing types of news items in *Newskit*.

ters with the schedule of events visible. A discussion of the layout process can be found later in this subsection.

Each type of data (articles, photos, or schedules) specified in the dtype file is used to create a `news_item` of the appropriate type and in the appropriate topic. For example, an article dtype is parsed into a layout_terminal_cart and is placed in a topic category (i.e. "Home Town News"). All of the `news_items` in the "Home Town News" topic can be retrieved.

When the layout manager is set, the layout file name is recorded and a flag is set in the manager that indicates that the file needs to be parsed when recomputed.

2.6 Rendering the PostScript file

The method `render_postscript()` is invoked for `layout_ps_newspa-per_fishwrap` and the output is directed to standard out[1]. The rest of this subsection describes the actual process of newspaper layout.

Most of the work is done as each section of the layout is (re)computed. When the section is built, two objects, `article_set_states` and `schedule_set_states`, are constructed. These objects maintain state for each related grouping of articles. This state keeps track of (1) which articles have not been printed, (2) which articles have not finished printing, and (3) which articles have been completely printed.

Each page of the section is laid out in order[2] until all of the article sets are "done." An article set is "done" if all of its articles have been completely printed. Every page has six areas (topic frames) reserved for rendering each topic. Figure2.3 shows the template for the first page.

1. The PostScript output is directed to `cout` defined in the iostream library for the C++ libraries for the UNIX file system.
2. The order of the articles is determined by the input file. Usually, the most important articles are placed first according to some metric of interest used the program that generates the input file.

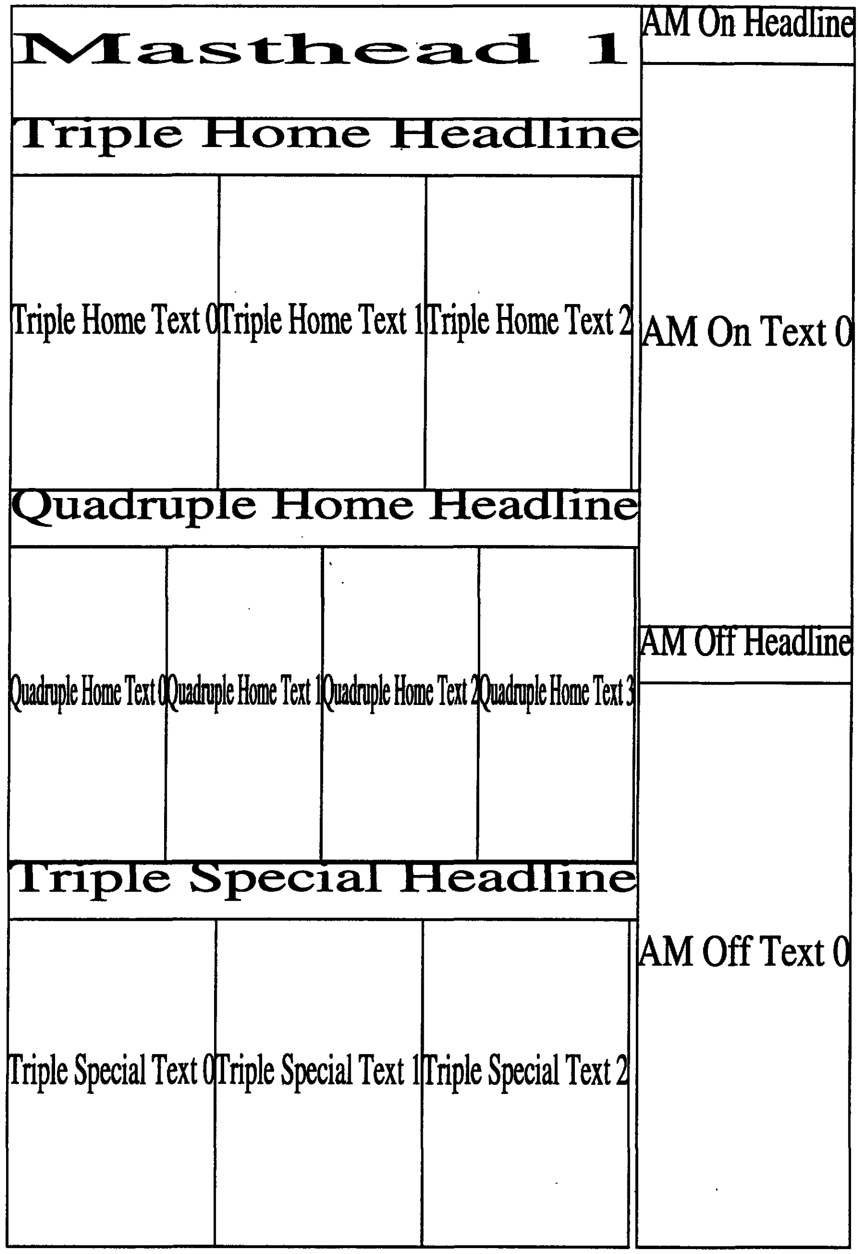


Figure 2.3: Front Page Template for Fishpaper

Each topic frame contains one or more subframes for the articles and schedules. These subframes are numbered in the template file so that they can be filled sequentially, that is body 0, 1, 2, etc. They are also designated as being frames for the either the headline, the author, the dateline, or the nth body frame.

Schedules are laid out one at a time. Multiple schedules are placed in the same frame if enough room is available. No frame resizing occurs for schedules. If all the schedules have been used, articles are rendered in their places.

Presently, **visual elements** and **contingencies** are the two primary concepts for laying out the individual news items (articles and schedules). A **visual element** can be thought of as a (rectangular) boundary which doesn't overlap any other **visual element.** For example, each paragraph in an article is treated as a **visual element.** Another example is the map of Georgia graphic in Appendix Figure D.3. The Recommendations Chapter provides an examination of representing articles as a compressible fluid instead of visual elements.

A **contingency** is a useful notion which is applicable to more than one situation. **Contingencies** are heuristics for handling uncommon situations. If a frame is designated as a graphic frame and no graphic is available, a **contingency** might use a three column article frame in its place. The Recommendations Chapter contains a description of how a frame might split into two frames in order to fill blank page space.

When one set of articles has been exhausted (i.e. no more articles exist for a particular topic) the topic frame might be left distractingly unfilled for every remaining page. To remedy this situation, the article frame is replaced with an article from the other set. A decision is made to determine whether to use a new or an unfinished article based on the page number and availability of unfinished articles. If a new article is started, its headline frame is filled and it is added to the unfinished set of articles for that topic. The headline frame size depends on the length and font size of the text for the headline. When the height of the frame is changed the column frames for that article are automatically shortened or lengthened[1]. Then the article columns are filled with text.

1. The process of dynamic updating is described in Appendix B.

This project includes an extension of the normal notion of an article in *Newskit*. A new article object contains the old object as well as the information necessary to maintain[1] the following:

1. data about continuations from one page to another,
2. where each paragraph is located,
3. paragraph split information,
4. article, headline and body frames, and
5. whether the article has been entirely rendered.

When the article frames are filled with text, an attempt is made to balance the text columns. If any columns of an article frame are left unfilled, the amount of leftover space is subtracted from the area of the article frame. All the paragraphs and continuations placed in this article frame are discarded and the columns are filled again. With the correct amount of area subtracted, columns of text should more nearly fill the new height.

When all of an article's text has been properly placed, the article is removed from the set of unfinished articles. Graphics are not included in the current version of `fishpaper`.

The layout process sets the initial information (rectangles for layout) and rendering writes the PostScript output to those rectangles. After all of the pages have been laid out, the header to the PostScript file is written to the output stream. Next, each page is rendered. Each article is checked to determine if it has any paragraphs on the current page. If so, an article method is invoked to output the PostScript code correctly placing the paragraph for that page. This process is repeated until each page is completely rendered.

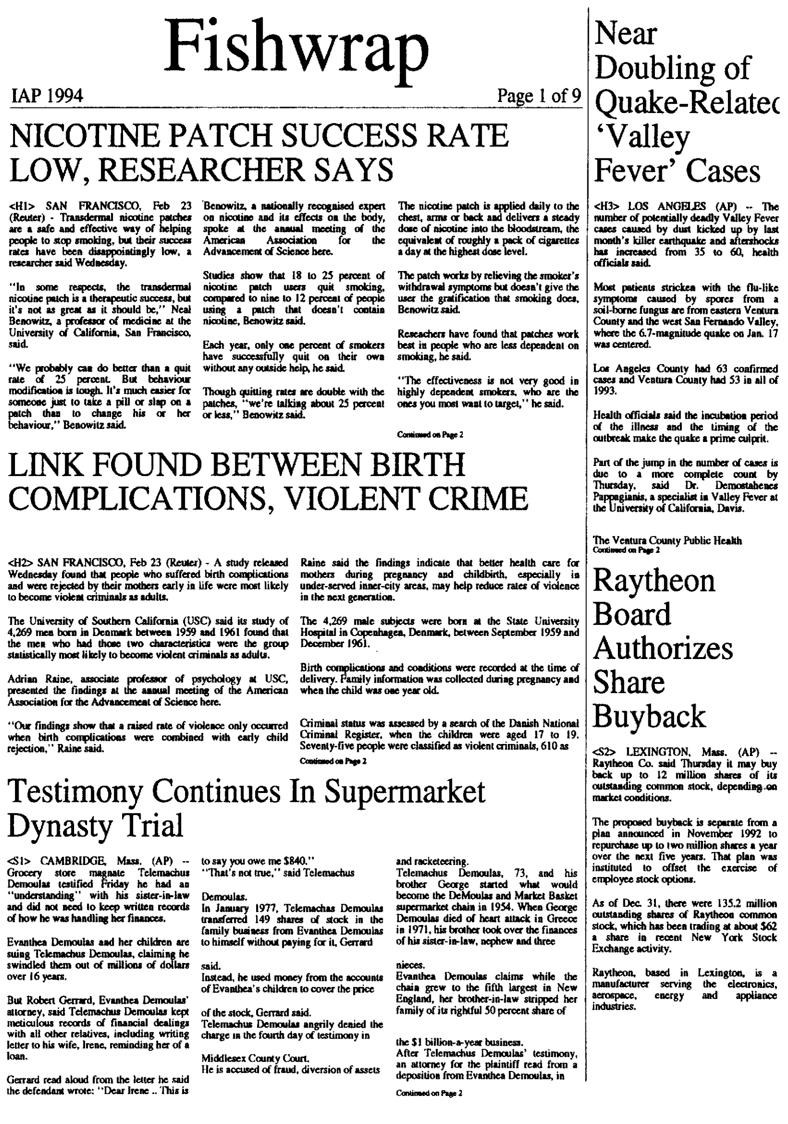
1. There is other state saved as well, like the horizontal distance between columns.

Chapter 3

Results

3.1 The Printed Newspapers

Figure 3.1 shows the front page of a newspaper printed by `fishpaper`. `fishpaper` was not completed in time for IAP 1994, so no schedules were available. `fishpaper` used a contingency to fill the right hand side with articles in the place of the absent schedules.



Fishwrap

IAP 1994 Page 1 of 9

NICOTINE PATCH SUCCESS RATE LOW, RESEARCHER SAYS

<H1> SAN FRANCISCO, Feb 23 (Reuter) - Transdermal nicotine patches are a safe and effective way of helping people to stop smoking, but their success rates have been disappointingly low, a researcher said Wednesday.

"In some respects, the transdermal nicotine patch is a therapeutic success, but it's not as great as it should be," Neal Benowitz, a professor of medicine at the University of California, San Francisco, said.

"We probably can do better than a quit rate of 25 percent. But behaviour modification is tough. It's much easier for someone just to take a pill or slap on a patch than to change his or her behaviour," Benowitz said.

Benowitz, a nationally recognised expert on nicotine and its effects on the body, spoke at the annual meeting of the American Association for the Advancement of Science here.

Studies show that 18 to 25 percent of nicotine patch users quit smoking, compared to nine to 12 percent of people using a patch that doesn't contain nicotine, Benowitz said.

Each year, only one percent of smokers have successfully quit on their own without any outside help, he said.

Though quitting rates are double with the patches, "we're talking about 25 percent or less," Benowitz said.

The nicotine patch is applied daily to the chest, arms or back and delivers a steady dose of nicotine into the bloodstream, the equivalent of roughly a pack of cigarettes a day at the highest dose level.

The patch works by relieving the smoker's withdrawal symptoms but doesn't give the user the gratification that smoking does, Benowitz said.

Reseachers have found that patches work best in people who are less dependent on smoking, he said.

"The effectiveness is not very good in highly dependent smokers, who are the ones you most want to target," he said.

Continued on Page 2

LINK FOUND BETWEEN BIRTH COMPLICATIONS, VIOLENT CRIME

<H2> SAN FRANCISCO, Feb 23 (Reuter) - A study released Wednesday found that people who suffered birth complications and were rejected by their mothers early in life were most likely to become violent criminals as adults.

The University of Southern California (USC) said its study of 4,269 men born in Denmark between 1959 and 1961 found that the men who had those two characteristics were the group statistically most likely to become violent criminals as adults.

Adrian Raine, associate professor of psychology at USC, presented the findings at the annual meeting of the American Association for the Advancement of Science here.

"Our findings show that a raised rate of violence only occurred when birth complications were combined with early child rejection," Raine said.

Raine said the findings indicate that better health care for mothers during pregnancy and childbirth, especially in under-served inner-city areas, may help reduce rates of violence in the next generation.

The 4,269 male subjects were born at the State University Hospital in Copenhagen, Denmark, between September 1959 and December 1961.

Birth complications and conditions were recorded at the time of delivery. Family information was collected during pregnancy and when the child was one year old.

Criminal status was assessed by a search of the Danish National Criminal Register, when the children were aged 17 to 19. Seventy-five people were classified as violent criminals, 610 as

Continued on Page 2

Testimony Continues In Supermarket Dynasty Trial

<S1> CAMBRIDGE, Mass. (AP) -- Grocery store magnate Telemachus Demoulas testified Friday he had an "understanding" with his sister-in-law and did not need to keep written records of how he was handling her finances.

Evanthea Demoulas and her children are suing Telemachus Demoulas, claiming he swindled them out of millions of dollars over 16 years.

But Robert Gerrard, Evanthea Demoulas' attorney, said Telemachus Demoulas kept meticulous records of financial dealings with all other relatives, including writing letter to his wife, Irene, reminding her of a loan.

Gerrard read aloud from the letter he said the defendant wrote: "Dear Irene .. This is to say you owe me $840."

"That's not true," said Telemachus

Demoulas.

In January 1977, Telemachas Demoulas transferred 149 shares of stock in the family business from Evanthea Demoulas to himself without paying for it, Gerrard

said.

Instead, he used money from the accounts of Evanthea's children to cover the price

of the stock, Gerrard said.

Telemachus Demoulas angrily denied the charge in the fourth day of testimony in

Middlesex County Court.

He is accused of fraud, diversion of assets and racketeering.

Telemachus Demoulas, 73, and his brother George started what would become the DeMoulas and Market Basket supermarket chain in 1954. When George Demoulas died of heart attack in Greece in 1971, his brother took over the finances of his sister-in-law, nephew and three

nieces.

Evanthea Demoulas claims while the chain grew to the fifth largest in New England, her brother-in-law stripped her family of its rightful 50 percent share of

the $1 billion-a-year business.

After Telemachus Demoulas' testimony, an attorney for the plaintiff read from a deposition from Evanthea Demoulas, in

Continued on Page 2

Near Doubling of Quake-Relatec 'Valley Fever' Cases

<H3> LOS ANGELES (AP) -- The number of potentially deadly Valley Fever cases caused by dust kicked up by last month's killer earthquake and aftershocks has increased from 35 to 60, health officials said.

Most patients stricken with the flu-like symptoms caused by spores from a soil-borne fungus are from eastern Ventura County and the west San Fernando Valley, where the 6.7-magnitude quake on Jan. 17 was centered.

Los Angeles County had 63 confirmed cases and Ventura County had 53 in all of 1993.

Health officials said the incubation period of the illness and the timing of the outbreak make the quake a prime culprit.

Part of the jump in the number of cases is due to a more complete count by Thursday, said Dr. Demostahenes Pappagianis, a specialist in Valley Fever at the University of California, Davis.

The Ventura County Public Health

Continued on Page 2

Raytheon Board Authorizes Share Buyback

<S2> LEXINGTON, Mass. (AP) -- Raytheon Co. said Thursday it may buy back up to 12 million shares of its outstanding common stock, depending on market conditions.

The proposed buyback is separate from a plan announced in November 1992 to repurchase up to two million shares a year over the next five years. That plan was instituted to offset the exercise of employee stock options.

As of Dec. 31, there were 135.2 million outstanding shares of Raytheon common stock, which has been trading at about $62 a share in recent New York Stock Exchange activity.

Raytheon, based in Lexington, is a manufacturer serving the electronics, aerospace, energy and appliance industries.

Figure 3.1: Printed Newspaper

Chapter 4

Conclusions

4.1 Limitations

The results indicate that flexible templates can be used for page-by-page stylistic variation. `Fishpaper`'s model of newspaper layout attempts to address the issue of aesthetics by offering the reader various choices. Ideally, many methods for creating and choosing among new layouts and formats would exist. However, this experiment has not been extended to allow more than one style of newspaper layout. The old joke is that the model T comes in any color as long as the color is black. Likewise, there is only one template that is in use. The creation of alternate templates is not too difficult. Guidelines for writing newspaper style templates can be found in Appendix B.

`fishpaper` only begins to solve some of the problems of whitespace reduction, however the Recommendations Chapter contains a description of the necessary steps for modifying the ideas used in Fishwrap to reduce blank regions.

For the case of `fishpaper`, the template file which was used to generate Figure 3.1 can only be changed in only certain ways. The number of columns can be altered for any article frame and the number of articles per topic frame can be changed as well. One inflexibility of `fishpaper` is the requirement of exactly six topics, using very specific names[1].

Arbitrary templates cannot currently be used in `fishpaper` because of one particular problem. The vertical line in the Figures of Chapter 3 were hard-coded into the `lay-out_nonterminal_bsection_ps_fishwrap` class. That is, the line is not dependent on the template file, but is instead drawn directly on the page by the class's

1. See Table 2.1 for the topic names.

render_postscript() method. Graphical enhancements, like rules, need to be based on template information, rather than rigidly encoded in the source code.

4.2 Contributions

Although this phase of layout is incomplete, progress has been made beyond previous attempts at automatic layout. The advancement of personalized PostScript newspapers in the Garden[1] can be found in Appendix D.

Layout for fishpaper contained several contributions to the endeavor to automatically generate personalized newspapers. The newspaper page in Epitaxy section D.5 appears to be more similar to a modern newspaper[2] than previous PostScript papers. One improvement over the image in D.5 is that fishpaper introduces balanced text columns.

Two designations for template files were discovered to be quite useful. One "article_frame" which is used for frames that contain the headline frame and the body frames. The body_frame designation specifies the frame which contains the list of body frames (the frames for each column are designated ("body" 0), ("body" 1), etc.).

There is something important to note about the "article_frame" designation. Every frame designated with "article_frame" must have its height set as one of its Y dimension values. This is a simple way to make sure that the height can be updated dynamically. The *layoutframe*[3] library used by fishpaper can currently only reset variables which are already set. For example, a frame's height cannot be reset if its top and its bottom are set, since there is no way to decide which to reset, top or bottom. This

1. The "Garden" is short for the "Terminal Garden" on the third floor of the MIT Media Lab. This room is so named because of its many computer terminals.
2. Compare image in D.5 with image in D.4.
3. *layoutframe* is the official name of frame template library.

problem can be solved by favoring one or the other, but that is a matter for future consideration.

Another contribution `fishpaper` of is not as blatantly visible as balanced columns. The special treatment of the layout of articles based on topic grouping is not found in any of the pspaper programs before `fishpaper`. Thoughtful newspaper layout places related articles on a single page. Content recognition is very important for this aspect of layout. During the Presidential Elections, Newspapers often create special layouts with candidates and related articles separated on a page for comparison. Visually offsetting these related articles with borderlines could be done automatically as well (see figure in Appendix D.2).

Notable improvements on the programming side of the problem concern the organization of groups of articles and the individual states of articles. The `article_set_-state` class maintains information about whether articles are finished or unfinished. This abstraction is important for choosing which articles to layout on a given page. Articles which have already been laid out, no longer need to be considered.

The bookkeeping involved with partially laid out articles was more difficult before the creation of the `article_state` abstraction. The task of the programmer is simplified if unsuccessful attempts to layout an article can easily be discarded. In previous implementations, these details were collected by the section's layout methods. Moving these details to the article is more appropriate since this information pertains directly to the article.

The `article_state` keeps track of layout information in temporary storage, the `layout_result_state`. This buffered information can either be used or discarded at a later time. For example, buffering layout state is useful when an article does not evenly fill frame columns. The layout state can be used to compute a better approximation for the height of the columns. The state can then be cleared and the frame laid out again, this time with more nearly even heights.

Chapter 5

Recommendations

5.1 The Overall Goal

This Chapter is divided into two parts. The first describes a method for modeling and solving the problem of automatic newspaper layout. This initial part is an outline of the direction and goal of the author's cumulative research concerning personalized hardcopy newspapers. The second half of this chapter contains suggestions for modifying and extending existing software for achieving this goal.

Automatic layout of a newspaper has traditionally been a difficult task to perform. It involves both the whitespace problem -- the subtle positioning of all visual elements on a page without wasting space -- as well as the problem of maintaining high quality aesthetics and readability.

Part 1: MODELING THE PROBLEM

Solving a problem well is often a matter of phrasing the question such that the answer is obvious.

Proposed Changes to *Newskit*

We can model the problem of newspaper layout as a multi-source/multi-sink maximum flow problem in graph theory. Each node in the graph represents a content frame[1] on a page. Frame dependencies form links from frame to frame (where some frames appear on later pages). It is natural to think of fluid flow, in the graph model, as the text which must "flow" from frame to frame (page to page). Flow capacities from frame to frame are the heights (areas) of text frames on a page. These capacities represent how much text can

1. The term "content frame" refers to a frame designated to contain text or graphics.

be held at a node. The multiple sources are the beginnings of each article and the multiple sinks represent the last frames used for each article.

The problem is slightly modified from the canonical form. The graph consists of a series of trees. Each tree is a page with the page frame as its root. Links will exist from tree (page) to tree (page) exactly when articles continue from tree (page) to tree (page).

Some constraints are looser, some are tighter. A looser constraint is that new nodes (frames) can be added to a page (see part 2 of this chapter). An obvious tighter constraint is that the sum of all the frame areas on a given page must be constant, since there is a fixed amount of layout real-estate per page.

Flow might not be continuous. The quantum of text might be a sentence or a paragraph (i.e. don't split paragraphs from column to column or from page to page). Multiple fonts affect the maximum amount of text in a frame as well.

Chapter 27 of Introduction to Algorithms[3], contains a discussion of preflow-push algorithms.

Part 2: MODIFYING EXISTING PROGRAMS

Frame dependencies (Appendix A.2 & B) form topologically sorted dags[1]. Frames can depend on other frames appearing in later pages with the `layout_article_state` class, which should be modified to keep track of all the frames in which an article appears. Currently, a page is read in and then discarded when the next page is read. The ideal way is to parse a new style page for every actual page needed. This way, when an article changes size on one page, its effects can be sent to all the other pages that need to update frame sizes to handle different amounts of text.

1. "Topologically Sorted" means that a frame only depends on frames named before itself (in the template file). A dag is a "directed acyclic graph" meaning that dependencies are one way and that there are no cyclic dependencies.

There are two ways to alleviate the rigid constraints of layout. The Results chapter contained pages of layout with large, blank areas of white space. The proposed method of avoiding this extra space is to precompute the total amount of text area needed for a section.

A useful concept is "scalable text," text that can be reduced in length by removing information of lowest importance first[1]. Some articles contain optional passages which are marked for omission[2]. Once it is known that the articles for a section will fit exactly, white space can be filled with new article frames. One possible improvement is to have a subclass of the frame class which is a "VSplitter." Subclasses can be designed to "behave" differently in a given situation. When a "VSplitter" frame does not have enough text to fill itself, it will vertically split into frames for two articles.

Special frame subclasses, dynamically resizable templates, and the maximum flow problem model can hopefully be used for solving problems of automatic newspaper layout.

5.2 Improvements for layout frames, *Newskit* and `fishpaper`

The visual editor for layout frames and an improved template layout language

A graphical program for creating frames and assigning designations (e.g. article order) is highly recommended if many templates are going to be generated and stored for reference. A program which would allow slight modification of predefined styles could store the differences as personalization data. For example, one reader might like the narrow column style of Wall Street Journal, but prefers a different font and no rules[3]. The difference file would just contain the *differences* between the personal file and the generic tem-

1. The concept of "scalable text" may one day include *increasing* size which is extended by searching for related information (a footnote type expansion).
2. Knight Ridder Tribune, for example, provides electronic articles which are segmented such that optional passages are distinct.
3. Rules are the lines dividing articles (columns).

plate. A visual interface is ideal for layout and would improve the speed of creating templates. The layout process would still be automatic, but the personalization aspect of layout would be improved by a greater diversity of styles. See Appendix B for a detailed account of the existing method for creating templates.

The new *dtypes++* library has interpretive language characteristics, like an evaluator for SCHEME[1] like expressions. The expressive nature of this language would allow arbitrary relationships of frames to one another. Currently, the only functional dependencies between frames are `offset` and `percent`. For example, using the new library, the `width` of frame A might be dependent on the *square* of the `width` of frame B, by using the *sqr* function. The evaluator is complete with lambda expressions and scoping.

Recommendations for layout with *Newskit*

Although *Newskit* contains the abstraction for creating new news items like schedules, perhaps there is a less time intensive way to describe the layout of these new objects. A language for placing text and graphics and the use of a visual frame editor would make the creation and layout of new news items more intuitive. For example, a new weather item might be created with a 14 point centered Roman title, a graphic, and an italic caption at the bottom. In this example, the contents of the *Newskit* object would contain a list of three things which could be laid out according to a certain template frame file. Placing objects in this manner is a better approach since the list's elements are laid out in order, independently of the list length. No specific information about the list elements is needed. Newly designed objects with their own individual layout methods can be added to the list.

Extensions to the format file might include border styles and other graphical flourishes.

1. The SCHEME programming language is similar to lisp.

Recommendations for Fishpaper

SHORT CUTS WHICH SHOULD BE GENERALIZED

The most obvious deficiency in using `fishpaper` as a model of generalized newspaper layout is the dependence on six particular topic categories. A benefit of this experiment, however, is that an approach for generalized layout may result from considering the use of arbitrarily many topics.

Most of the functionality of an arbitrary version would be the same, however, the layout would be blind to the particular topic categories. Any list of topics could be used. The layout routine would chose the next topic frame and match it (numerically or by string comparison with the topic title) to the correct topic, in order to extract the next article.

EXTENSIONS

More variety of content in the presented articles should be included. Both *Newskit* and the Freshman Fishwrap provide the inclusion of graphics (journalistic photos). Previous experiments with graphics and frame templates have been successful[1] in PostScript generation of newspapers. The inclusion of graphics in `fishpaper` is first among the proposed extensions.

Other forms of article content include author and dateline information[2]. The inclusion of these is straightforward: create specific, designated frames for this information and update the appropriate layout routines for articles. In addition, printing the topic category with an article would convey a sense of overall placement in a section. This placement could be important for justifying to the reader why a particular article was included (e.g. the article is in the "Movies Topic").

1. Summer 1993 UROP for Pascal Chesnais and the MIT Media Lab concerning automated newspaper layout with flexible templates. Aspect ratios of photos can be maintained by frame dimensions which depend on the dimensions of the same frame (e.g. 3 to 4, width depends on height). Appendix D does not contain an example of this.
2. See Appendix D.2.

CONSOLIDATING

Some of the code in fishpaper should be combined so that articles and schedules are treated as uniform layout items. Grouping in this way is necessary for allowing arbitrary news items to print according to individual methods of the particular item.

Appendix A

Example Files

A.1 User_hardcopy.dtype -- The Input File

The `user_hardcopy.dtype`[1] file is generally many pages in length. For clarity, only a single article is attached.

The important aspects of its structure are listed below. This example is the beginning of a file titled "bdschoon_bpaper.dtype." The *`user`* part of `user_hardcopy.dtype` is the actual user name of the reader. The example is a simplified illustration of the structure. Table 5.1 shows the necessary fields for a `user_hardcopy.dtype` file. Any extra fields are ignored by `fishpaper`.

Required Field of Newspaper	Description of Field Contents
"name"	Title of the newspaper
"type"	type of news item which should be created when this file is parsed
"format-file"	filename for the format file
"layout-file"	filename for the layout file (newspaper style template)
"sections"	contains a list of the sections of the newspaper.

Table A.1: Dtype Fields for newspaper

Each section comprises a list of topics[2], each topic contains a list of articles. Table 5.2 lists the necessary fields of an article. Notice that the article in the `user_hardcopy.d-type` file contains many extra fields. These fields are ignored by `fishpaper`.

1. See FOO for a detailed description of dtypes.
2. Table 2.1 lists the valid topics for `fishpaper`.

Required Field of an Article	Description of Field Contents
"headline"	headline of the article
"type"	type should be ("item" "text" "article")
"body"	body (the text) of the article

Table A.2: Dtype Fields for article

A.2 Template.style.dtype -- The Layout File

The attached template file was used to generate Figure 3.1. The *style* part of the `tmplate.style.dtype` filename is replaceable by a style name. For example there is a file entitled "template.fishwrap4.dtype" for the style fishwrap4.

A.3 Format_file.dtype -- The Format File

The format file `format.fish1.dtype` was used to create the newspaper in Figure 3.1. Notice that the font information for schedules is ignored in the newspapers which don't have schedules in them. The *fish1* part of the file title indicates the style of the format file.

Appendix B

Guide To Using Layout Frames

B.1

Compiling

To compile the `fishpaper` program, type

```
% cd /source/$SYS/garden/news/pspaper
% gmake
```

To rebuild from scratch, type

```
% cd /source/$SYS/garden/news/pspaper
% gmake reset
% gmake depend
% gmake
```

Viewing Templates

To view a template, use the program `template_view` in

/source/$SYS/garden/news/pspaper/install/bin

See the various version descriptions below to find template files.

Writing Template Files

Three versions of template file specification exist. Version 2 is the current version. Since version 2 and version 3 have few modifications with respect to the version 1, most of the specification will be given in describing version 1, with only the appropriate modifications appearing in the descriptions version 2 and version 3.

B.2 Version 1

Examples can be found in

/source/MASTER/garden/news/pspaper/resources/templates/version1

There are two parts to every template file, an INFO part and a DATA part. The INFO part consists of name/value pairs, such as `("name" "Demo Template 2")` below:

```
(("INFO"
  (("name" "Demo Template 2")
   ("template" "true")
   ("Note" "This is a layout dtype for a newspaper"
           "Avoid using the special words below")
   ("Special" ("page" "pagewidth" "columnwidth" "parent"))
   ("Hierarchy" ("Sections" "Pages" "Frames" "Attributes"))))
```

The DATA part consists of a list of sections. Each section begins with a section name:

```
(("Section 1"
```

and is followed by a list of it's pages. Similarly, a page begins with a page name and contains a list of frames. A frame description begins with a frame name as well:

```
(("Page 1"                              ◄──── page title
 (("Demo Template 2 Masthead 1"         ◄──── frame title
```

Regard the description of the frame "Demo Template 2 Masthead 1":

```
(("Demo Template 2 Masthead 1"
  (("parent" "page")
   ("designator" "masthead")
   ("top" ("top" "parent"))
   ("left" ("left" "parent"))
   ("width" ("width" "parent"))
   ("height" 36)))
```

Every frame can contain some optional information described below.

Assigning a Parent Page

Any frame name used before the current frame may be used as a parent. If a graphical template builder were written, in which frames can be removed and inserted, the frames must be topologically sorted[1].

1. Topologically sorted means that frames only depend on previous frames.

There are two special frame names to avoid: "`page`" and "`parent`." The implicit frame "`page`" has the dimensions of the page passed in to `template_view`. The purpose of the "`parent`" frame is described below.

Designating Frames for Special Purposes[1]

Frame "`Demo Template 2 Masthead 1`" has the designator:

("designator" "masthead")

Some other valid designators are "`pagehead`" and "`headline`."The problem of frame order is solved by numerically designating the topic, article and body of the frame. For example:

("designator" ("topic" 0 ("article" 3 ("body" 0))))

specifies that this frame should be filled with the 0th topic's 3rd article's 0th body text.

Other Attributes ("empty" and "graphic")[2]

There are two attributes "`empty`" and "`graphics`" which can either be paired with "true" or "false." If a frame specification contains `("empty" "true")`, `templat-e_view` will not draw the frame in the generated PostScript file. These attributes can be used or ignored depending on the program using the frame.

Geometric Attributes

A frame should be specified with exactly four geometric constraints[3]. In the X (horizontal) dimension, one may specify any two of

`left, right, width, xcenter.`

Similarly, two should be chosen in the Y (vertical) dimension:

`bottom, top, height, ycenter.`

1. This type of designation is specific to version 1.
2. These are also specific to version 1.
3. cf. Appendix C

NUMERIC CONSTRAINTS

The value of an attribute can be numeric (only integer numbers can be used with the current version of dtypes). Default units are 72nd's of an inch.

("height" 36)

SIMPLE DEPENDENCE

An attribute value can depend on the value of another frame. The special string "`parent`" refers to the specified parent of the frame. Three of the attributes of the frame example above depend on the "`parent`" frame, which in this case is the "`page`" frame.

FUNCTIONS OF OTHER ATTRIBUTES

Frame attributes can depend on functions of other frame attributes. Here are three examples:

("left" ("offset" ("inch" 1) ("xcenter" "parent")))

("top" ("offset" ("ptsize" -36) ("top" "parent")))

("width" ("percent" (40) ("width" "parent")))

In the first line, the "`left`" of the frame is set to 1 inch plus the "`xcenter`" of the "`parent`" frame. In the second line, the "`top`" of the frame is set to the 36 "`ptsize`" (point sizes) below the "`top`" of its "`parent`." In the third line, the "`width`" of the frame is set to forty percent of the width of the its "`parent`" frame.

Techniques for writing template files

The most useful way to describe a page using templates is to imagine dividing the page into two frames. Then just keep splitting the subframes in two until the template looks right. In the case of `template.globe.dtype`, the page is first divided into the masthead at the top and the rest on the bottom. Then the rest is split into a left and a right.

This process is continued until the final result. In this process of recursively subdividing the page, each frame in entirely contained within its parent.

B.3 Version 2

Examples can be found in

/source/MASTER/garden/news/pspaper/resources/templates/version2/

"Designators" are replaced by "Designations"

There is one main difference in version 2. The "`designator`" is called "`designation`" and is structured differently. Previously a frame which looked like this:

```
("Left 0 Graphic Contingency Headline"
 (("parent" "Left 0 Graphic Contingency")
  ("empty" "true")
  ("designator" ("topic" 0 ("article" 2 ("headline"))))
  ("top" ("top" "parent"))
  ("height" 36)
  ("left" ("left" "parent"))
  ("width" ("width" "parent"))))
```

will be transformed into:

```
("Left 0 Graphic Contingency Headline"
 (("parent" "Left 0 Graphic Contingency")
  ("designation" (("topic" 0)
                  ("empty" "true")
                  ("article" 2)
                  ("headline" "true")))
  ("top" ("top" "parent"))
  ("height" 36)
  ("left" ("left" "parent"))
  ("width" ("width" "parent"))))
```

All the information except the parent and the geometry information is placed in the designation. The parent information might be incorporated into the designation in a future version.

Converting Version 2 files

The current version of the *dtype++* library uses a dtype file format which makes reading and editing `template.style.dtype` files more difficult than before. Fortunately there is a Perl script which can be used to add the appropriate symbols to create usable version 2 files.

There is a link to the program in the `/source/MASTER/garden/news/pspaper/resources/templates/version2` directory. Here is an example of what to type to create a new dtype file called "`new.dtype`" from a version 2 template file "`template2.fishwrap.dtype`":

```
convert_template.prl < template2.fishwrap.dtype > new.dtype
```

The new file will contain symbols like "." and "#".

B.4 Version 3 -- Not Fully Implemented

Examples can be found in

/source/MASTER/garden/news/pspaper/resources/templates/version3/

Exploiting Hierarchy

In this version, the major difference is that templates are separate. Instead of listing all the sections in the newspaper directly, a newspaper template might look like this:

```
(("INFO"
   (("name" "Newspaper Demo")
    ("template" "true")
    ("newspaper" "true")
    ("version" 3.0)))
 ("DATA"
   ("sections"
    (("Section 1"
       ("file" "section.demo.dtype"))
     ("Section 2"
       ("file" "section.demo2.dtype"))))))
```

Here the section files are read in later. This scheme has the advantage that frames can easily by substituted. This is the first step toward creating subclasses of frames which have different functionality depending on different situations (all this is to be defined by a layout program which would use frames). Version 3 is a good stage at which to redesign the use/separation of the `INFO` part and the `DATA` part.

Appendix C

Flexible Templates For Newspaper Layout

C.1 Page Description

STRUCTURE

To define the problem of layout, the *structure* of the layout representation must be described. Newspaper structures require sections, pages, and subpage elements. Each of these can be specified by a textual description called a **template**[1].

Templates convey commonalities between similar objects. For example, any page composed of nine frames could be specified by the same template[2].

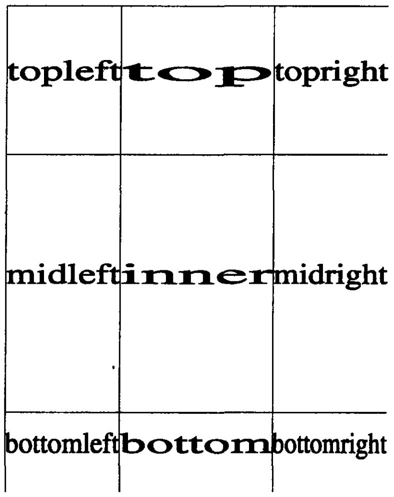


Figure C.1: Nine Frames inside a (hidden) tenth frame.

Similarly, **frames** are templates for rectangular subpage elements[3]. One frame might describe a three column text article, while another specifies a graphic with a caption.

1. See Appendix C.1 (file template.demo2.dtype)
2. These nine frames may depend on each other in many ways. To be more specific, any page template with nine frames, *which depend on each other in the same way*, can be specified by the same template. See "FUNCTIONALITY -- Three Difficulties" for a description of dependencies.
3. See Appendix C.2 (Three Column Text Article)

Because these template descriptions are independent of content and because they can change dynamically, we need only to specify relative placement and shape. Templates are structured hierarchically; the following is an ordered list, (each structure can contain the following one): *newspaper*, *section*, *page*, *frame*. Frame structures can contain other frame structures, as well. Templates resemble trees more than they resemble lists, (the newspaper is the root which has several sections -- the branches[1]; the leaves are frames).

FUNCTIONALITY -- Three Difficulties

Consider the *functionality* of the layout representation. The specification of a frame must allow for shrinkage or growth. When describing a frame, one might want to say that its `top` should be attached to the `bottom` of a second frame. Then if that second frame grows taller, the first frame can move down and/or contract if necessary. There is thus a **constraint** imposed on the first frame because its location and/or shape is dependent upon another frame.

Frame specifications depend on an expressive set of **attribute**s which describe a frame, such as `width`, `left`, or `center`. Three main difficulties must be resolved with this scheme: the constraint problem, the many-possible-descriptions problem and the dependency-representation problem.

THE CONSTRAINT PROBLEM

The first problem arises when there isn't enough information about a frame's geometry. The program must determine whether a description is **under-constrained**. For example, if all we know is that the frame's `left`[2] is 50, we cannot determine its `right`. Conversely, perhaps too many requirements are set on the frame and it is **over-constrained**. In this case, the over-specified frame might have: `top` = 200, `bottom` = 100,

1. See Appendix C.1 (file template.demo2.dtype).
2. Assume generic units. Unless specified otherwise, units are 72nd's of an inch and the origin of the axes on a page is at the bottom, left hand corner.

height = 5. It would be much too burdensome for the person writing the frame template to have to follow a complicated set of attribute ordering rules. A better approach would employ a computer to catch errors of under and over-constrained descriptions.

THE MANY-POSSIBLE-DESCRIPTIONS PROBLEM

The second problem is that there are several ways to describe the same dimensions. This case is equivalent to not knowing how much information to store about a frame. For example, a frame with left = 100, width = 50, bottom = 100, ycenter = 125 will have the same location and size as a frame specified by xcenter = 125, right = 150, top = 150, height = 50. If the attribute values were only numbers, it would suffice to limit attribute descriptions to some canonical set, such as left, top, right, and bottom. However, it is important to note, that these specifications may not describe the same frame. Consider the case when each frame's attributes depend on other frames rather than on specific numbers. The first frame's width might actually be set to be 50% of the width of a containing frame with the same center. If the containing frame's width is doubled, the inner frame's width should also be doubled. In this example it is much more useful to represent the width as actually a width and not a left and a right.

THE DEPENDENCY-REPRESENTATION PROBLEM

The third problem deals with recording dependencies between frames when we know the important information concerning the dimensions. If frame A depends on frame B's width, how should the dependency be recorded if B is defined not by a width, but by a right and an xcenter. Fortunately, there is an efficient way to solve all these problems at once by choosing the right representation.

The "Right" Representation

The solution involves first simplifying the problem into a smaller one, solving that, and then generalizing the smaller problem to the original problem. Consider one dimension at a time. In the horizontal directions, we have `left`, `right`, `width` and `xcenter` attributes. A critical observation is that any two of these *exactly* specifies our left and right boundaries. Therefore, if we have any fewer than two in a specification, the frame is under-constrained. Similarly with more than two, the system is over-constrained. There are no deeply nested case decisions to make so the *CONSTRAINT* problem is solved.

To solve the *MANY-POSSIBLE-DESCRIPTIONS* problem, only two attributes need to be stored, that is, it is unnecessary to store all four attributes[1] in this dimension. The other attributes can be determined when needed, by computing them. In fact, the width is the positive difference of the two end points (`left` and `right`) and the `xcenter` is the average of the two end points. This solution can be generalized to the vertical direction using height as the positive difference and `ycenter` as the average, or even to the third dimension, imagining boxes with depths and `zcenters`. The same code can even be used for each dimension. The DEPENDENCY-REPRESENTATION problem is solved, as well, since a dependent frame's attribute values can be computed at anytime. All that is needed is to find the independent frame and request its attribute value.

Maintaining the Representation Invariant

So far, the design for frame specification has been presented. It is another matter to design a program which maintains the dependencies of one frame on another and updates changes in an efficient manner. To perform this task, frames must maintain:

1. pointers to two other frames per dimension (in the case of dependencies)
2. a list of dependents (if the frame is destroyed, its dependents should no longer depend on it)

1. The four attributes in the X (horizontal) dimension) are `left`, `right`, `width`, and `xcenter`.

When a frame's `height` is changed, that frame's attributes must be changed. This leads to two difficulties. The first problem occurs if one of the frame's `height` depends on another frame. This problem can be avoided by breaking the dependency, although this may not always yield the desired result.

The second problem occurs if the frame has no vertical dimension attributes are defined by `height`. There are two attributes to change, but there is no way to know which one to alter (consider that the `top` and `bottom` are set when the `height` is changed). This problem is also circumvented in an undesirable manner: article frames must be specified by their heights and one other vertical dimension attribute. This is a convention for now, and is acceptable for newspaper templates[1].

After updating a frame's `height`, all of its dependents might need to be changed. If the `bottom` changes then the `ycenter` will also change. These attributes (`height`, `bottom`, and `ycenter` in this case) are checked against the attributes of the frame's dependents. If any of the dependents match, then those dependents are updated and the process continues with the dependents' dependents. This program exploits the following efficiency. When a frame's `height` changes, then its `left` or `width` will not change. The dependency update procedure only needs to check changes of attributes in a single dimension. The same code can be used for any dimension.

1. The Recommendations section suggests a way of specifying a "nailed" so that, the `top` would not change if nailed and the height were modified.

Data Dependency Diagram for C++ Layout Frames

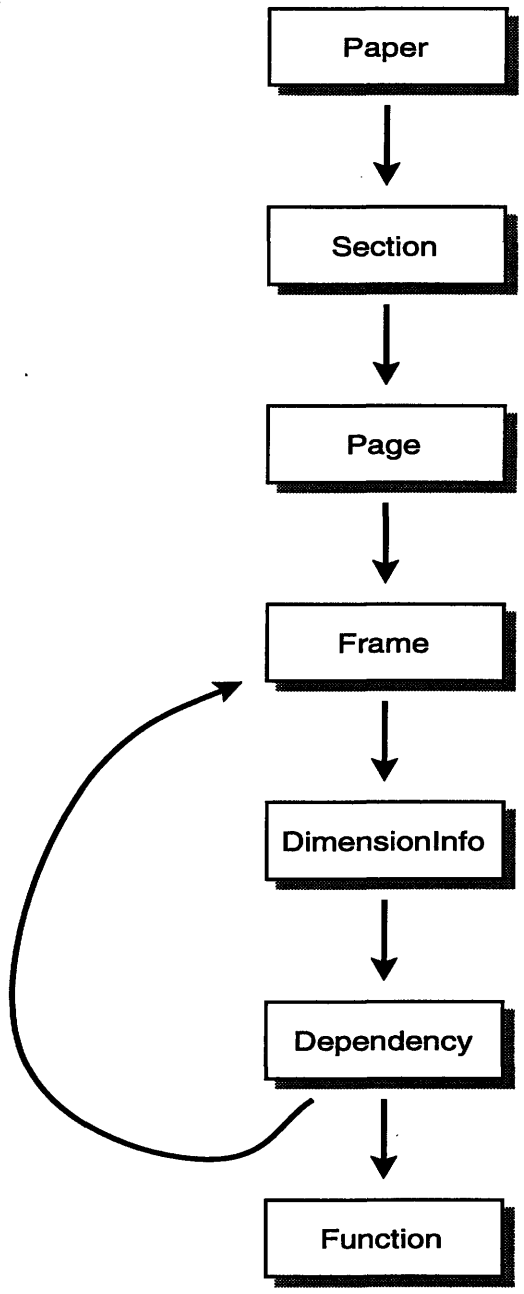


Figure C.2: Data Dependency Diagram

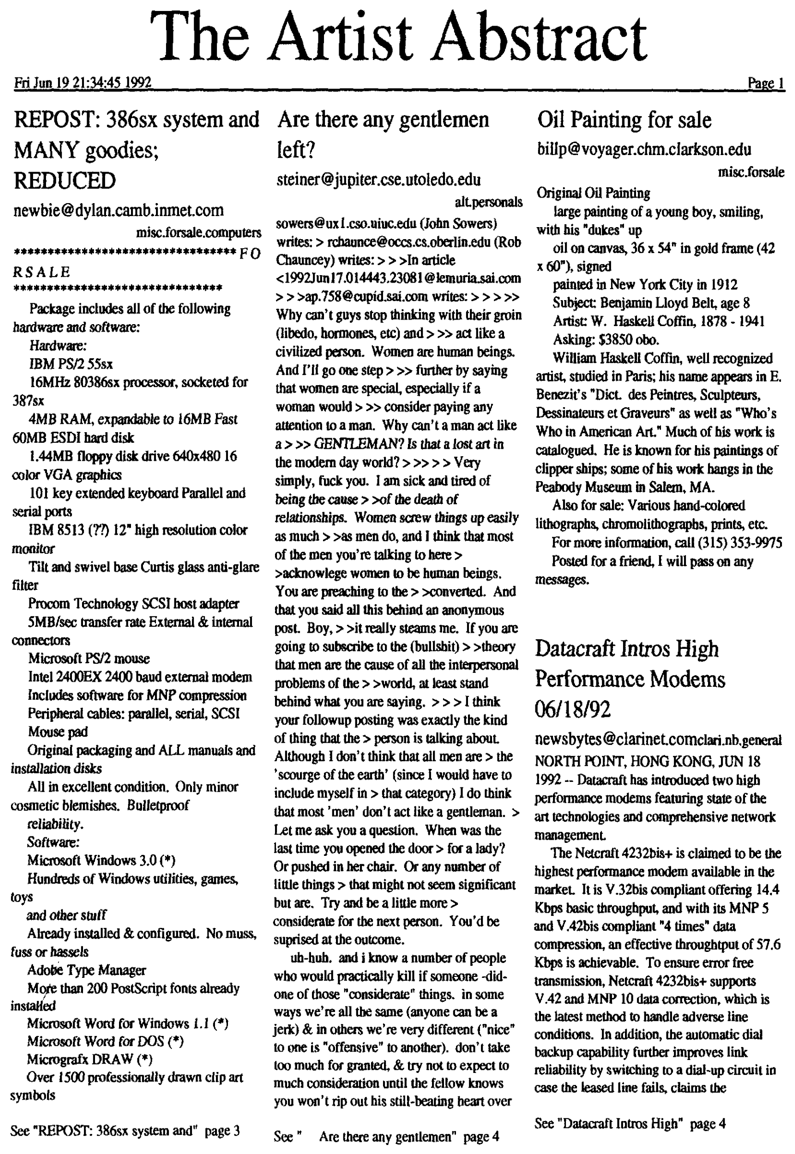
Appendix D

The History of PostScript Newspapers in The Garden

D.1 Pspaper 1992 -- The Artist Abstract

Below is page 1 of "The Artist Abstract" which was generated for the artist community model. Notice the three column format and continuations at the bottoms of each column. The fonts can be customized for each individual reader from a format file.

Figure D.1: The Artist Abstract -- 6/19/92



The Artist Abstract

Fri Jun 19 21:34:45 1992 Page 1

REPOST: 386sx system and MANY goodies; REDUCED

newbie@dylan.camb.inmet.com

misc.forsale.computers

********************************* F O R S A L E *******************************

Package includes all of the following hardware and software:

Hardware:

IBM PS/2 55sx

16MHz 80386sx processor, socketed for 387sx

4MB RAM, expandable to 16MB Fast 60MB ESDI hard disk

1.44MB floppy disk drive 640x480 16 color VGA graphics

101 key extended keyboard Parallel and serial ports

IBM 8513 (??) 12" high resolution color monitor

Tilt and swivel base Curtis glass anti-glare filter

Procom Technology SCSI host adapter

5MB/sec transfer rate External & internal connectors

Microsoft PS/2 mouse

Intel 2400EX 2400 baud external modem

Includes software for MNP compression

Peripheral cables: parallel, serial, SCSI

Mouse pad

Original packaging and ALL manuals and installation disks

All in excellent condition. Only minor cosmetic blemishes. Bulletproof

reliability.

Software:

Microsoft Windows 3.0 (*)

Hundreds of Windows utilities, games, toys

and other stuff

Already installed & configured. No muss, fuss or hassels

Adobe Type Manager

More than 200 PostScript fonts already installed

Microsoft Word for Windows 1.1 (*)

Microsoft Word for DOS (*)

Micrografx DRAW (*)

Over 1500 professionally drawn clip art symbols

See "REPOST: 386sx system and" page 3

Are there any gentlemen left?

steiner@jupiter.cse.utoledo.edu

alt.personals

sowers@ux1.cso.uiuc.edu (John Sowers) writes: > rchaunce@occs.cs.oberlin.edu (Rob Chauncey) writes: > > >In article <1992Jun17.014443.23081@lemuria.sai.com > > >ap.758@cupid.sai.com writes: > > > >> Why can't guys stop thinking with their groin (libedo, hormones, etc) and > >> act like a civilized person. Women are human beings. And I'll go one step > >> further by saying that women are special, especially if a woman would > >> consider paying any attention to a man. Why can't a man act like a > >> *GENTLEMAN? Is that a lost art in* the modern day world? > >> > > Very simply, fuck you. I am sick and tired of *being the cause > >of the death of relationships. Women screw things up easily* as much > >as men do, and I think that most of the men you're talking to here > >acknowlege women to be human beings. You are preaching to the > >converted. And that you said all this behind an anonymous post. Boy, > >it really steams me. If you are going to subscribe to the (bullshit) > >theory that men are the cause of all the interpersonal problems of the > >world, at least stand behind what you are saying. > > > I think your followup posting was exactly the kind of thing that the > person is talking about. Although I don't think that all men are > the 'scourge of the earth' (since I would have to include myself in > that category) I do think that most 'men' don't act like a gentleman. > Let me ask you a question. When was the last time you opened the door > for a lady? Or pushed in her chair. Or any number of little things > that might not seem significant but are. Try and be a little more > considerate for the next person. You'd be suprised at the outcome.

uh-huh. and i know a number of people who would practically kill if someone -did- one of those "considerate" things. in some ways we're all the same (anyone can be a jerk) & in others we're very different ("nice" to one is "offensive" to another). don't take too much for granted, & try not to expect to much consideration until the fellow knows you won't rip out his still-beating heart over

See " Are there any gentlemen" page 4

Oil Painting for sale

billp@voyager.chm.clarkson.edu

misc.forsale

Original Oil Painting

large painting of a young boy, smiling, with his "dukes" up

oil on canvas, 36 x 54" in gold frame (42 x 60"), signed

painted in New York City in 1912

Subject: Benjamin Lloyd Belt, age 8

Artist: W. Haskell Coffin, 1878 - 1941

Asking: $3850 obo.

William Haskell Coffin, well recognized artist, studied in Paris; his name appears in E. Benezit's "Dict. des Peintres, Sculpteurs, Dessinateurs et Graveurs" as well as "Who's Who in American Art." Much of his work is catalogued. He is known for his paintings of clipper ships; some of his work hangs in the Peabody Museum in Salem, MA.

Also for sale: Various hand-colored lithographs, chromolithographs, prints, etc.

For more information, call (315) 353-9975

Posted for a friend, I will pass on any messages.

Datacraft Intros High Performance Modems 06/18/92

newsbytes@clarinet.comclari.nb.general

NORTH POINT, HONG KONG, JUN 18 1992 -- Datacraft has introduced two high performance modems featuring state of the art technologies and comprehensive network management.

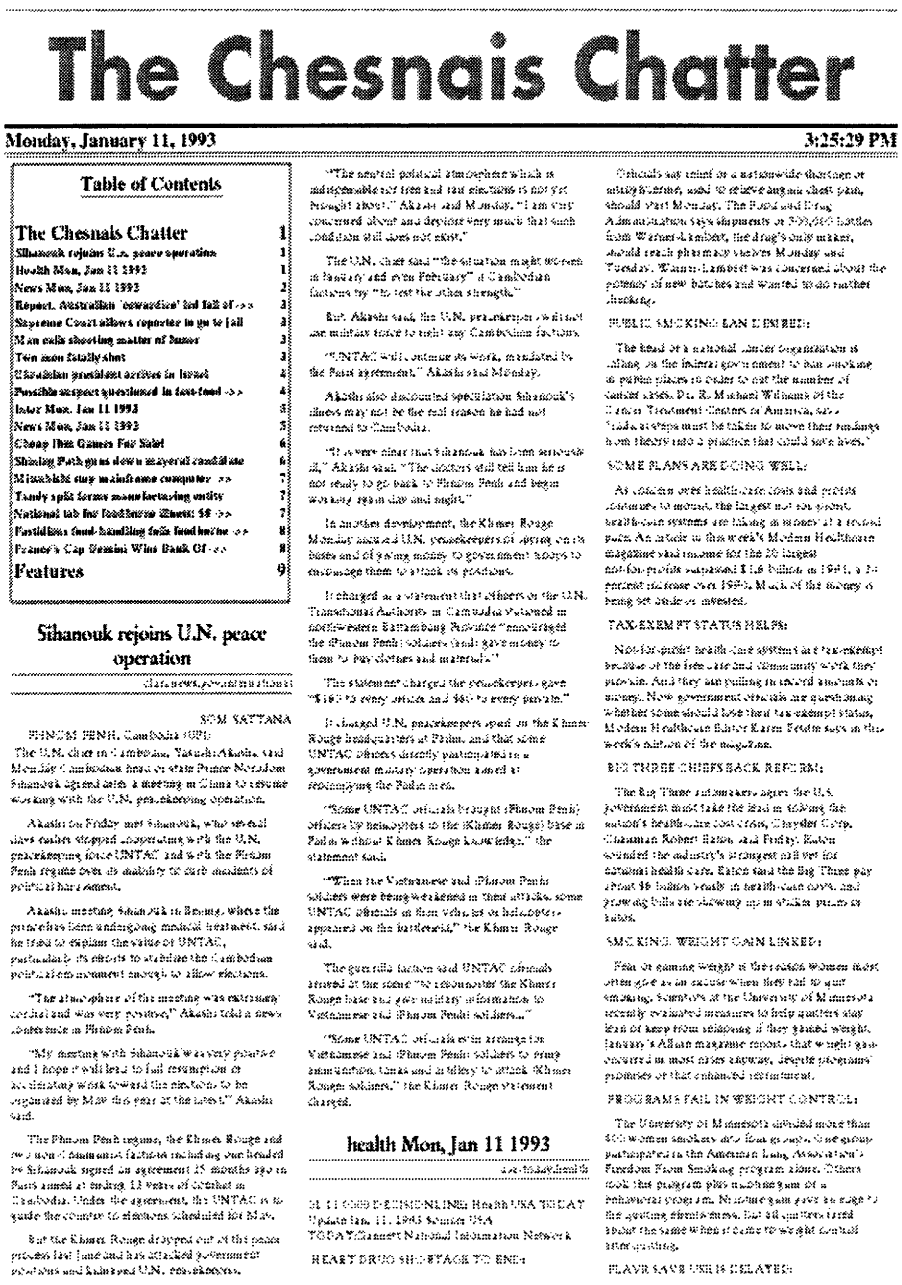
The Netcraft 4232bis+ is claimed to be the highest performance modem available in the market. It is V.32bis compliant offering 14.4 Kbps basic throughput, and with its MNP 5 and V.42bis compliant "4 times" data compression, an effective throughtput of 57.6 Kbps is achievable. To ensure error free transmission, Netcraft 4232bis+ supports V.42 and MNP 10 data correction, which is the latest method to handle adverse line conditions. In addition, the automatic dial backup capability further improves link reliability by switching to a dial-up circuit in case the leased line fails, claims the

See "Datacraft Intros High" page 4

D.2 Pspaper 1993 -- The Chesnais Chatter

This version of pspaper is a little more aesthetically pleasing. The source of the first article "clari.news.gov.international" is visually distinct by the horizontal rules. Source, Author and Dateline information were printed if available, each with it's own justification and font characteristics. Note the "Table of Contents" which is distinguished by a border

Figure D.2: The Chesnais Chatter 1993 - 1/11/93



The Chesnais Chatter

Monday, January 11, 1993 **3:25:29 PM**

Table of Contents

Sihanouk rejoins U.N. peace operation

clari.news.gov.international

SOM SATTANA

PHNOM PENH, Cambodia (UPI)

The U.N. chief in Cambodia, Yasushi Akashi, said Monday Cambodian head of state Prince Norodom Sihanouk agreed after a meeting in China to resume working with the U.N. peacekeeping operation.

Akashi on Friday met Sihanouk, who several days earlier stopped cooperating with the U.N. peacekeeping force UNTAC and with the Phnom Penh regime over its inability to curb incidents of political harassment.

Akashi, meeting Sihanouk in Beijing, where the prince has been undergoing medical treatment, said he tried to explain the value of UNTAC, particularly its efforts to stabilize the Cambodian political environment enough to allow elections.

"The atmosphere of the meeting was extremely cordial and was very positive," Akashi told a news conference in Phnom Penh.

"My meeting with Sihanouk was very positive and I hope it will lead to full resumption of accelerating work toward the elections to be organized by May this year at the latest," Akashi said.

The Phnom Penh regime, the Khmer Rouge and two non-Communist factions including one headed by Sihanouk signed an agreement 15 months ago in Paris aimed at ending 13 years of conflict in Cambodia. Under the agreement, the UNTAC is to guide the country to elections scheduled for May.

But the Khmer Rouge dropped out of the peace process last June and has attacked government positions and kidnapped U.N. peacekeepers.

"The neutral political atmosphere which is indispensable for free and fair elections is not yet brought about," Akashi said Monday. "I am very concerned about and deplore very much that such condition still does not exist."

The U.N. chief said "the situation might worsen in January and even February" if Cambodian factions try "to test the other strength."

But Akashi said the U.N. peacekeepers will not use military force to fight any Cambodian factions.

"UNTAC will continue its work, mandated by the Paris agreement," Akashi said Monday.

Akashi also discounted speculation Sihanouk's illness may not be the real reason he had not returned to Cambodia.

"It seems clear that Sihanouk has been seriously ill," Akashi said. "The doctors still tell him he is not ready to go back to Phnom Penh and begin working again day and night."

In another development, the Khmer Rouge Monday accused U.N. peacekeepers of spying on its bases and of giving money to government troops to encourage them to attack its positions.

It charged in a statement that officers of the U.N. Transitional Authority in Cambodia stationed in northwestern Battambang Province "encouraged the (Phnom Penh) soldiers (and) gave money to them to buy clothes and materials."

The statement charged the peacekeepers gave "$100 to every officer and $50 to every private."

It charged U.N. peacekeepers spied on the Khmer Rouge headquarters at Pailin, and that some UNTAC officers directly participated in a government military operation aimed at recapturing the Pailin area.

"Some UNTAC officials brought (Phnom Penh) officers by helicopters to the (Khmer Rouge) base in Pailin without Khmer Rouge knowledge," the statement said.

"When the Vietnamese and (Phnom Penh) soldiers were being weakened in their attacks, some UNTAC officials in their vehicles or helicopters appeared on the battlefield," the Khmer Rouge said.

The guerrilla faction said UNTAC officials arrived at the scene "to reconnoiter the Khmer Rouge base and give military information to Vietnamese and (Phnom Penh) soldiers..."

"Some UNTAC officials even arrange for Vietnamese and (Phnom Penh) soldiers to bring ammunition, tanks and artillery to attack Khmer Rouge soldiers," the Khmer Rouge statement charged.

health Mon, Jan 11 1993

usa.today.health

01 11 0000 DECLINE-NLINE: Health USA TODAY Update Jan. 11, 1993 Source: USA TODAY/Gannett National Information Network

HEART DRUG SHORTAGE TO END:

Officials say relief on a nationwide shortage of nitroglycerine, used to relieve angina chest pain, should start Monday. The Food and Drug Administration says shipments of 500,000 bottles from Warner-Lambert, the drug's only maker, should reach pharmacy shelves Monday and Tuesday. Warner-Lambert was concerned about the potency of new batches and wanted to do further checking.

PUBLIC SMOKING BAN URGED:

The head of a national cancer organization is calling on the federal government to ban smoking in public places in order to cut the number of cancer cases. Dr. R. Michael Williams of the Cancer Treatment Centers of America, says "radical steps must be taken to move these findings from theory into a practice that could save lives."

SOME PLANS ARE DOING WELL:

As concern over health-care costs and profits continues to mount, the largest not-for-profit health-care systems are taking in money at a record pace. An article in this week's Modern Healthcare magazine said income for the 20 largest not-for-profits surpassed $1.6 billion in 1991, a 24 percent increase over 1990. Much of the money is being set aside or invested.

TAX-EXEMPT STATUS HELPS:

Not-for-profit health-care systems are tax-exempt because of the free care and community work they provide. And they are pulling in record amounts of money. Now government officials are questioning whether some should lose their tax-exempt status, Modern Healthcare Editor Karen Fesche says in this week's edition of the magazine.

BIG THREE CHIEFS BACK REFORM:

The Big Three automakers agree the U.S. government must take the lead in solving the nation's health-care cost crisis, Chrysler Corp. Chairman Robert Eaton said Friday. Eaton sounded the industry's strongest call yet for national health care. Eaton said the Big Three pay about $6 billion yearly in health-care costs, and growing bills are showing up in sticker prices of autos.

SMOKING, WEIGHT GAIN LINKED:

Fear of gaining weight is the reason women most often give as an excuse when they fail to quit smoking. Scientists at the University of Minnesota recently evaluated measures to help quitters stay lean or keep from relapsing if they gained weight. January's Allure magazine reports that weight gain occurred in most cases anyway, despite programs' promises of that enhanced recruitment.

PROGRAMS FAIL IN WEIGHT CONTROL:

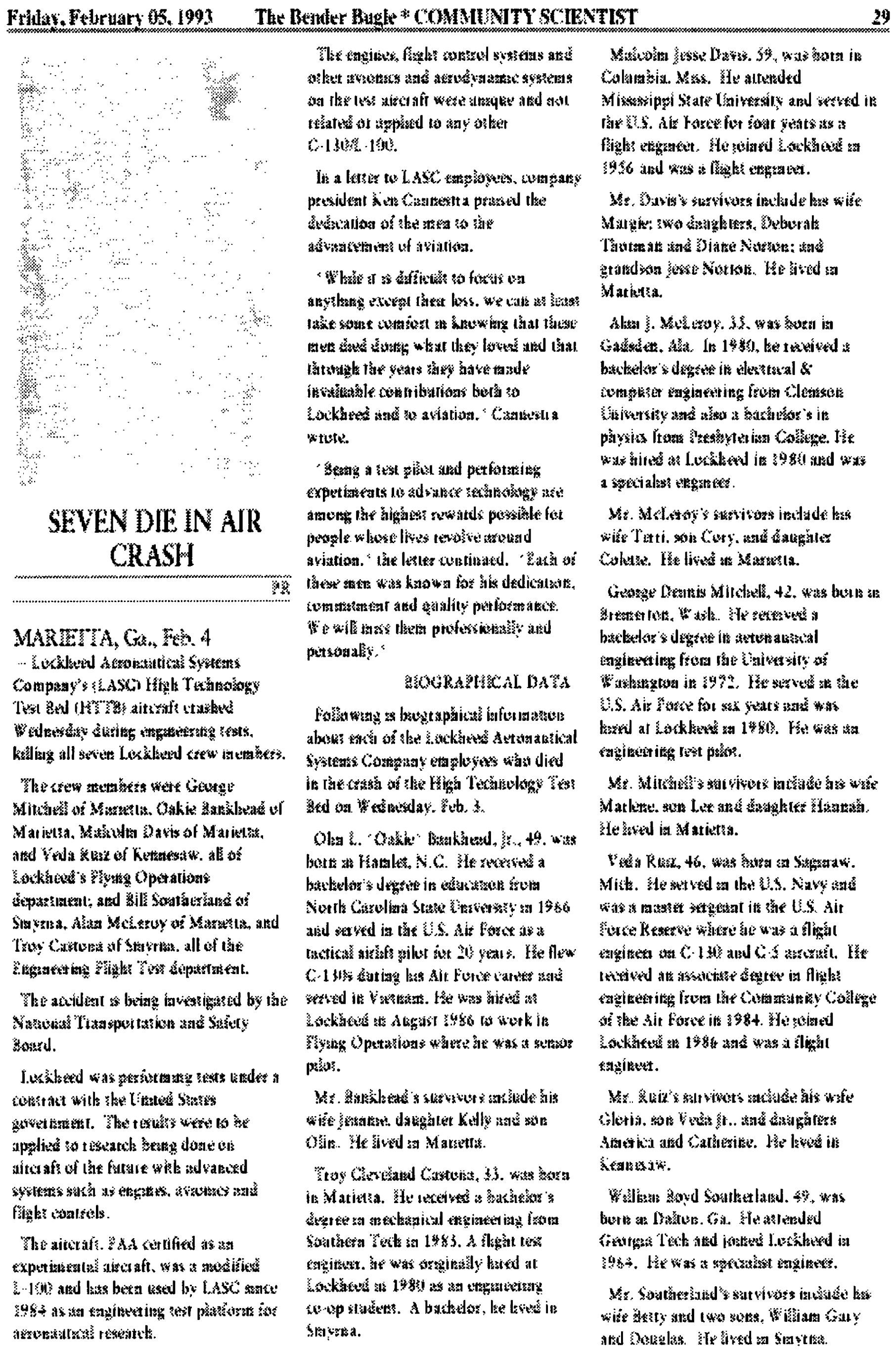
The University of Minnesota divided more than 400 women smokers into four groups. One group participated in the American Lung Association's Freedom From Smoking program alone. Others took that program plus nicotine gum or a behavioral program. Nicotine gum gave an edge to the quitting determination, but all quitters fared about the same when it came to weight control after quitting.

FLAVR SAVR USE IS DELAYED:

D.3 Pspaper 1993 -- The Bender Bugle

Still using the same layout program as the previous example in A.2, a map of Georgia has been included automatically since "Ga." appears in the dateline.

Figure D.3: The Bender Bugle -- 2/5/92



Friday, February 05, 1993 The Bender Bugle * COMMUNITY SCIENTIST 29

SEVEN DIE IN AIR CRASH

92

MARIETTA, Ga., Feb. 4

-- Lockheed Aeronautical Systems Company's (LASC) High Technology Test Bed (HTTB) aircraft crashed Wednesday during engineering tests, killing all seven Lockheed crew members.

The crew members were George Mitchell of Marietta, Oakie Bankhead of Marietta, Malcolm Davis of Marietta, and Veda Ruiz of Kennesaw, all of Lockheed's Flying Operations department; and Bill Southerland of Smyrna, Alan McLeroy of Marietta, and Troy Castona of Smyrna, all of the Engineering Flight Test department.

The accident is being investigated by the National Transportation and Safety Board.

Lockheed was performing tests under a contract with the United States government. The results were to be applied to research being done on aircraft of the future with advanced systems such as engines, avionics and flight controls.

The aircraft, FAA certified as an experimental aircraft, was a modified L-100 and has been used by LASC since 1984 as an engineering test platform for aeronautical research.

The engines, flight control systems and other avionics and aerodynamic systems on the test aircraft were unique and not related or applied to any other C-130/L-100.

In a letter to LASC employees, company president Ken Cannestra praised the dedication of the men to the advancement of aviation.

'While it is difficult to focus on anything except their loss, we can at least take some comfort in knowing that these men died doing what they loved and that through the years they have made invaluable contributions both to Lockheed and to aviation,' Cannestra wrote.

'Being a test pilot and performing experiments to advance technology are among the highest rewards possible for people whose lives revolve around aviation,' the letter continued. 'Each of these men was known for his dedication, commitment and quality performance. We will miss them professionally and personally.'

BIOGRAPHICAL DATA

Following is biographical information about each of the Lockheed Aeronautical Systems Company employees who died in the crash of the High Technology Test Bed on Wednesday, Feb. 3.

Olin L. 'Oakie' Bankhead, Jr., 49, was born in Hamlet, N.C. He received a bachelor's degree in education from North Carolina State University in 1966 and served in the U.S. Air Force as a tactical airlift pilot for 20 years. He flew C-130s during his Air Force career and served in Vietnam. He was hired at Lockheed in August 1986 to work in Flying Operations where he was a senior pilot.

Mr. Bankhead's survivors include his wife Jeanne, daughter Kelly and son Olin. He lived in Marietta.

Troy Cleveland Castona, 33, was born in Marietta. He received a bachelor's degree in mechanical engineering from Southern Tech in 1985. A flight test engineer, he was originally hired at Lockheed in 1980 as an engineering co-op student. A bachelor, he lived in Smyrna.

Malcolm Jesse Davis, 59, was born in Columbia, Miss. He attended Mississippi State University and served in the U.S. Air Force for four years as a flight engineer. He joined Lockheed in 1956 and was a flight engineer.

Mr. Davis's survivors include his wife Margie; two daughters, Deborah Thotman and Diane Norton; and grandson Jesse Norton. He lived in Marietta.

Alan J. McLeroy, 33, was born in Gadsden, Ala. In 1980, he received a bachelor's degree in electrical & computer engineering from Clemson University and also a bachelor's in physics from Presbyterian College. He was hired at Lockheed in 1980 and was a specialist engineer.

Mr. McLeroy's survivors include his wife Terri, son Cory, and daughter Colette. He lived in Marietta.

George Dennis Mitchell, 42, was born in Bremerton, Wash. He received a bachelor's degree in aeronautical engineering from the University of Washington in 1972. He served in the U.S. Air Force for six years and was hired at Lockheed in 1980. He was an engineering test pilot.

Mr. Mitchell's survivors include his wife Marlene, son Lee and daughter Hannah. He lived in Marietta.

Veda Ruiz, 46, was born in Saginaw, Mich. He served in the U.S. Navy and was a master sergeant in the U.S. Air Force Reserve where he was a flight engineer on C-130 and C-5 aircraft. He received an associate degree in flight engineering from the Community College of the Air Force in 1984. He joined Lockheed in 1986 and was a flight engineer.

Mr. Ruiz's survivors include his wife Gloria, son Veda Jr., and daughters America and Catherine. He lived in Kennesaw.

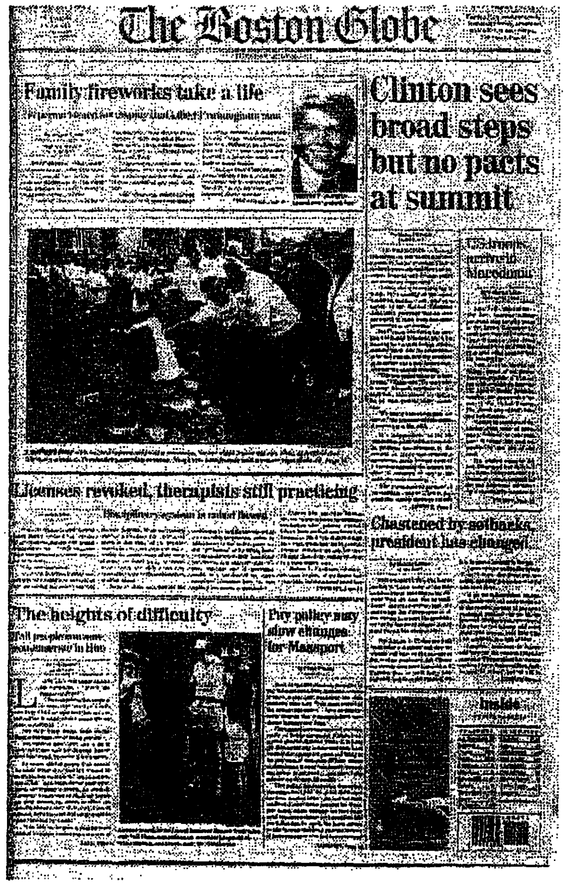
William Boyd Southerland, 49, was born in Dalton, Ga. He attended Georgia Tech and joined Lockheed in 1964. He was a specialist engineer.

Mr. Southerland's survivors include his wife Betty and two sons, William Gary and Douglas. He lived in Smyrna.

D.4 Pspaper 1993 -- Templates

In the summer of 1993, templates were used in an attempt to create newspaper layout expressive enough to mimic major modern newspapers. Here a scanned in copy of the front page of the Boston Globe, July 6, 1993.

Here is the template version of the same front page:



The Boston Globe

Family fireworks take a life

Clinton sees broad steps but no pacts at summit

US troops arrive in Macedonia

Licenses revoked, therapists still practicing

Chastened by setbacks, president has changed

The heights of difficulty

Pay policy may allow changes for Massport

Inside

Figure D.4: The Boston Globe -- 7/6/93

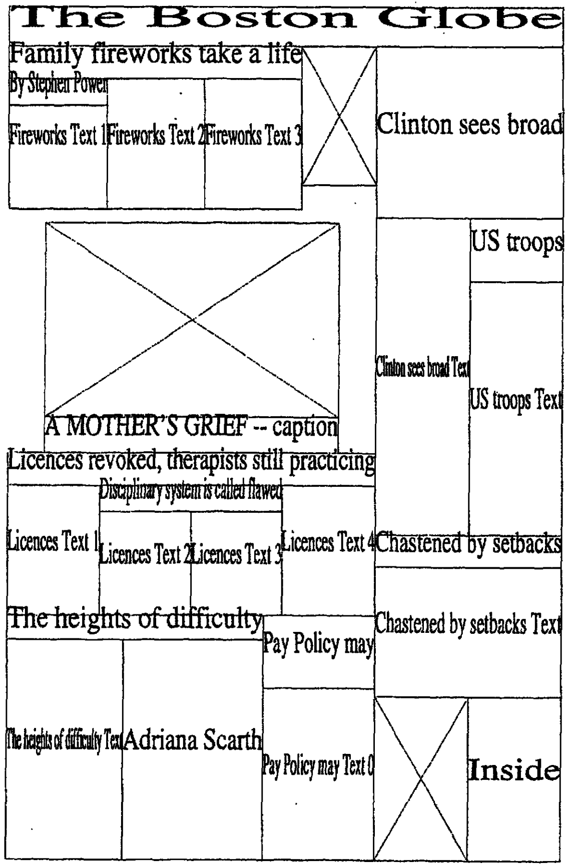
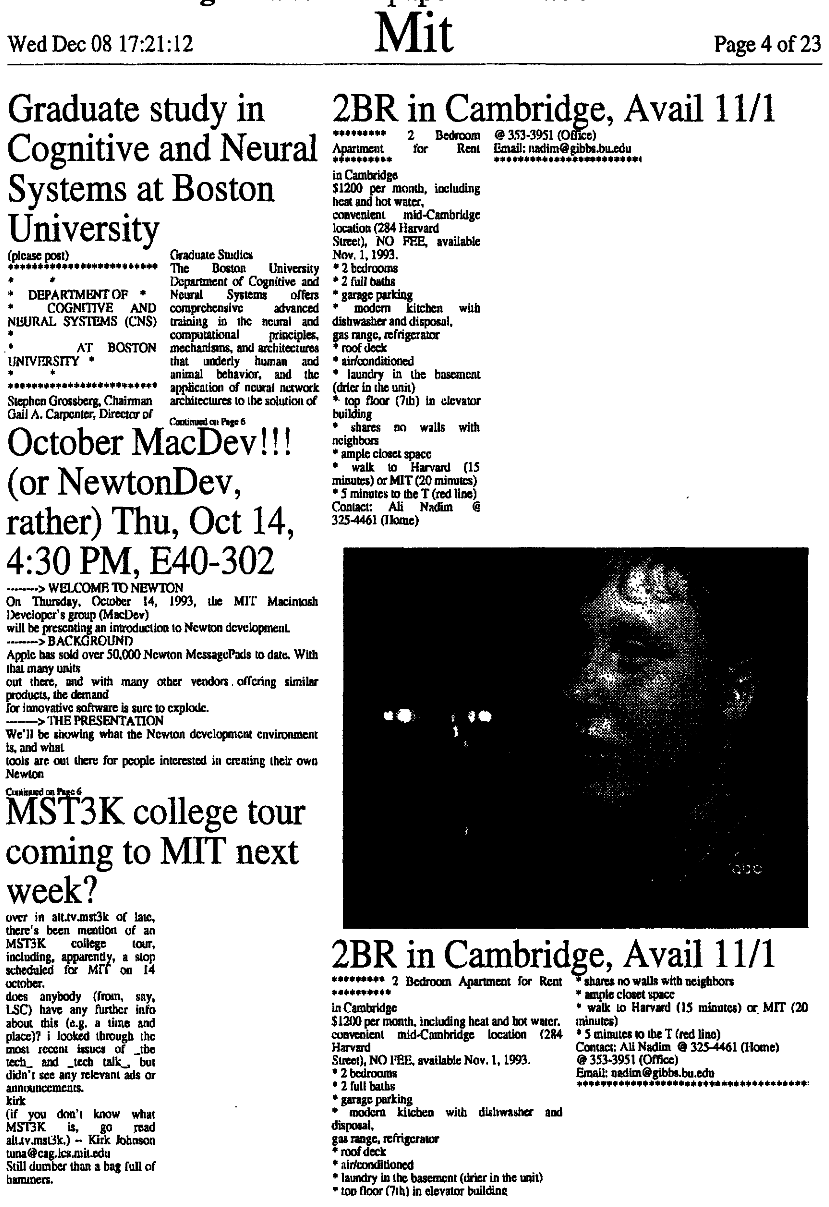


Figure D.5: Globe Template of Front Page -- 7/6/93

D.5 Pspaper 1993 -- A newspaper generated with templates

Notice the white space after the end of each article and the unbalanced columns. Headline frame heights were altered at the run time of the program. The layout program would insert articles in frames when a graphic is not available. Replacement of one type content with another is an example of a contingency. Notice, too, that continuations are present at the ends of the articles. The dateline appears, but there are no distinguishing visual characteristics such as rules between articles.

Figure D.6: Mit paper -- 10/8/93



Wed Dec 08 17:21:12 Mit Page 4 of 23

Graduate study in Cognitive and Neural Systems at Boston University

(please post)
****
*   *
*   DEPARTMENT OF   *
*   COGNITIVE AND NEURAL SYSTEMS (CNS)
*
*   AT BOSTON UNIVERSITY *
*   *
****
Stephen Grossberg, Chairman
Gail A. Carpenter, Director of
Graduate Studies
The Boston University Department of Cognitive and Neural Systems offers comprehensive advanced training in the neural and computational principles, mechanisms, and architectures that underly human and animal behavior, and the application of neural network architectures to the solution of

Continued on Page 6

October MacDev!!! (or NewtonDev, rather) Thu, Oct 14, 4:30 PM, E40-302

-------> WELCOME TO NEWTON
On Thursday, October 14, 1993, the MIT Macintosh Developer's group (MacDev)
will be presenting an introduction to Newton development.
-------> BACKGROUND
Apple has sold over 50,000 Newton MessagePads to date. With that many units
out there, and with many other vendors offering similar products, the demand
for innovative software is sure to explode.
-------> THE PRESENTATION
We'll be showing what the Newton development environment is, and what
tools are out there for people interested in creating their own Newton

Continued on Page 6

MST3K college tour coming to MIT next week?

over in alt.tv.mst3k of late, there's been mention of an MST3K college tour, including, apparently, a stop scheduled for MIT on 14 october.
does anybody (from, say, LSC) have any further info about this (e.g. a time and place)? i looked through the most recent issues of _the tech_ and _tech talk_, but didn't see any relevant ads or announcements.
kirk
(if you don't know what MST3K is, go read alt.tv.mst3k.) -- Kirk Johnson
tuna@cag.lcs.mit.edu
Still dumber than a bag full of hammers.

2BR in Cambridge, Avail 11/1

********* 2 Bedroom Apartment for Rent **********
in Cambridge
$1200 per month, including heat and hot water,
convenient mid-Cambridge location (284 Harvard
Street), NO FEE, available Nov. 1, 1993.
* 2 bedrooms
* 2 full baths
* garage parking
* modern kitchen with dishwasher and disposal,
gas range, refrigerator
* roof deck
* air/conditioned
* laundry in the basement (drier in the unit)
* top floor (7th) in elevator building
* shares no walls with neighbors
* ample closet space
* walk to Harvard (15 minutes) or MIT (20 minutes)
* 5 minutes to the T (red line)
Contact: Ali Nadim @ 325-4461 (Home)
@ 353-3951 (Office)
Email: nadim@gibbs.bu.edu
****

2BR in Cambridge, Avail 11/1

********* 2 Bedroom Apartment for Rent **********
in Cambridge
$1200 per month, including heat and hot water,
convenient mid-Cambridge location (284 Harvard
Street), NO FEE, available Nov. 1, 1993.
* 2 bedrooms
* 2 full baths
* garage parking
* modern kitchen with dishwasher and disposal,
gas range, refrigerator
* roof deck
* air/conditioned
* laundry in the basement (drier in the unit)
* top floor (7th) in elevator building
* shares no walls with neighbors
* ample closet space
* walk to Harvard (15 minutes) or MIT (20 minutes)
* 5 minutes to the T (red line)
Contact: Ali Nadim @ 325-4461 (Home)
@ 353-3951 (Office)
Email: nadim@gibbs.bu.edu
****

References

[1] F. J. Reintjes. *Computer Assisted Layout of Newspapers*, Massachusetts Institute of Technology, Electrical Engineering and Computer Science Dept., 1977.

[2] L. Silverstein. *Newspaper Design For The Times*, Von Nostrand Reinhold, York, New York, 1990.

[3] T. Cormen, C. Leiserson, R. Rivest. *Introduction to Algorithms*, Chapter 27, The MIT Press, Cambridge Massachusetts, New York, 1990.

[4] P. Chesnais and D. Koen. Strategies for personal dynamic systems: News in the future. In *NextWORLD Expo*, 1993.

[5] K. Dienes. "Newskit Reference Manual." Massachusetts Institute of Technology, 1993.

[6] K. Dienes. "Dtype++ Reference Manual." Massachusetts Institute of Technology, 1994.

[7] N. Abramson. "The Dtype Library or, How to Write a Server in Less Time than it Takes to Read This Manual." Technical report, Electroinc Publishing Group, MIT Media Laboratory, 1992.

[8] P. Chesnais. "The Freshman Fishwrap." MIT Media Laboratory, 1994.

[9] A. Gurtler. "History of the development of the newspaper," Swiss Typographic Monthly Magazine.