39

Automated Restructuring of an Electronic Newspaper

by

Douglas B. Koen

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 1994
[June 1994]

Author .....
Department of Electrical Engineering and Computer Science
May 19, 1994

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

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

Automated Restructuring of an Electronic Newspaper

by

Douglas B. Koen

Submitted to the Department of Electrical Engineering and Computer Science on May 19, 1994, in partial fulfillment of the requirements for the degree of Bachelor of Science

Abstract

The Freshman Fishwrap project at the MIT Media Laboratory makes a personalized electronic newspaper available to any interested member of the MIT community. In its original implementation the Fishwrap's software provided a mechanism for readers to manipulate the elements that would be contained in their paper, but lacked the ability to respond to their reading habits by modifying the order of the presentation of those elements.

An algorithm is explored for automated structuring of an electronic newspaper. The objective was to create an algorithm that effectively restructures the newspaper while monitering the reading habits of the user in order to alter the structure of his paper to reflect these habits. The algorithm was designed with integration into the Freshman Fishwrap project in mind.

Thesis Supervisor: Pascal Chesnais
Title: Research Specialist, MIT Media Laboratory
This work was supported in part by IBM and the NIF Consortium.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Electronic means of information dissemination will be able to compete with the more traditional mediums if they are both similar enough to be recognizable to the reader and add value by demonstrating a responsiveness to the user's interests and reading habits. This responsiveness should provide the reader with a sense that the newspaper is being prepared specifically for him instead of for massive distribution. More and more information dissemination services for home computer users are appearing, such as Prodigy, CompuServe, and America OnLine--the average computer owner is gaining access to a wealth information in his own living room. But systems often fail to tap into the potential of the computer, still requiring the user to make substantial effort to explicitly instruct the system on how to work for him. The value of the computer as an intelligent receiver instead of a passive device (like current televisions) for digital information was discussed by Lippman and Bender [LB87].

The effect of this intelligent receiver of information in the homes of millions on the way news and information is distributed is profound, and is destined to become more so as increasing amounts of information become available in a digital form.

It is estimated that only 10% of the information available to a newspaper for a given edition is actually included, and of that, only about 10% is read by any particular reader [Smith80]. The problem lies in the fact that the editor of the paper has a limited amount of total space in his publication, so articles deemed of lesser importance must be discarded.

Similarly, once an article has passed the grade to be included, the amount of information about that particular story that can be included in the article is again highly constrained by available real estate on the page. The result is that not only is the reader reading only 10% of what he's presented, but he's not even getting **all** of the information available about the stories that he does decide read, as some had to be eliminated to make room for the stories he isn't going to read. The best solution to this problem would be to have a separate edition of the newspaper for each reader, including only the parts of the paper that he wishes to read, and including extensive coverage of the material presented.

The solution to the above problem puts the computer in the place of the printed page, providing essentially the same service, only somewhat more focused and geared towards the particular reader. But by installing a computer as the medium for delivering the information we have positioned ourselves to do much more than just making this simple refinement. Generation of an electronic newspaper can be broken down into a pipeline with three major components: content selection, structuring, and display (See Figure 1.1).

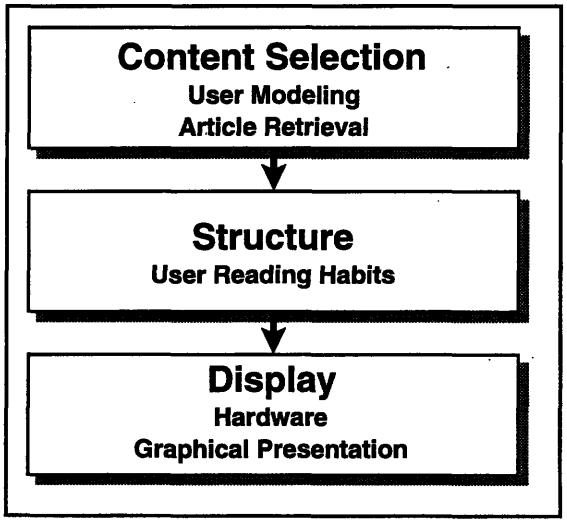

Figure 1.1: Electronic Newspaper Pipeline

Selection of content for an electronic newspaper falls largely in the domain of user modeling. Much progress has been made in this area. For example, Orwant's Doppelgänger system is used by the Media Laboratory [Orwant91, Orwant93]. Doppelgänger dynamically updates its model of a user based on information gathered from a variety of "sensors", including finger information, an active identification badge to track consenting users' movements, as well as feedback derived from applications designed to make use of the system. Orwant continues to explore algorithms and sensors for improving the system's capabilities. The other half of this part of the pipeline is article retrieval, which can be handled fairly nicely by any reasonably flexible database program. The particular news storage and retrieval database used at the Media Lab is Betty [Blount91b], a system based on a flexible Lisp-like data structure.

The third stage of the pipeline is display. Lie has done some work on improving the user interface to electronic newspapers in order to make accessing information by computer more accessible [Lie91]. His work focused on making use of large monitors with substantially more screen real estate than the average display, providing a map of the paper to the user on a second screen, and the addition to the display of visual cues to the user--such as the use of color to denote an article's source, with the saturation of the color decreasing as the story becomes older as if it were fading.

The second stage of the pipeline, the structuring of the information gathered in the first stage, was addressed by Blount [Blount91b], who investigated the use of cluster analysis for consistent grouping and positioning of news articles in an electronic newspaper. While this effort does address the issue of structure in an electronic newspaper, its emphasis is geared more towards automation and placement consistency than it is towards addressing the structuring of the electronic newspaper based on the reading habits of the user.

Once the determination is made as to what information should be included in a user's personalized newspaper, decisions must then be made regarding the structuring of those elements within the edition of the paper. In a printed paper these decisions must, to some extent, be made in tandem--as various new stories are conceived and completed during the process of the paper's creation, the editor must dynamically determine which articles belong on which pages and how much of them can be included. Not so with an electronic newspaper. Page real estate is virtual, so these processes can be decoupled.

For example, if I always read the Arts section of my Sunday paper first, then the comics, then the front page, and then the regional section; it would be incredibly convenient for me if the publishers of The Boston Globe would be kind enough to put my newspaper in that order before delivering it to me. No such luck--the effort to do this for every reader is beyond the means of any newspaper. Even if it were not, my desires and the desires of all of my fellow Sunday paper readers are somewhat fickle. I may decide next Sunday that I want to make slight changes to that ordering--maybe put the comics first because my schedule is too busy to go to any local theater events anyway, so the Arts section has become frustrating. Or maybe the change isn't that small--maybe I want to start reading my paper in a whole different order. With an electronic newspaper we can not only provide customized structuring, but we can even provide the user with a feedback loop so that he can make implicit changes to our understanding of his reading habits--just by reading the paper differently. The more he reads the paper in the same way, the more we become convinced that our assumptions are correct.

1.2 Scope

The intent of this thesis is two-fold. The first object is to test the validity of the assertion that the separation of article selection algorithms from dynamic paper structuring algorithms in an electronic newspaper is both feasible and desirable. The second is to propose and demonstrate an implementation of a particular dynamic structuring algorithm that is based solely on the relationships between elements instead of based on a system of global punishment and reward of elements for whether or not they were read.

Chapter 2

Algorithm

A newspaper can be divided into distinct elements. The most prominent division is sections, and the next is topics. In a hardcopy newspaper the sections are very clearly specified, whereas the topics can be found by looking for clusters of articles within a section. For example, articles on health care or on a war will often be found close to each other. Each of these dividing elements have subelements; the subelements of the sections are the topics, and the subelements of the topics are the articles themselves. The purpose of this thesis is to devise an algorithm that will cause the automated structuring of the elements of an electronic newspaper based on the feedback of the reader. Some possible solutions to the problem include neural networks and clustering techniques. These options may effectively solve the problem. Additional considerations that must be taken into account, however, are the issues of speed and space. For an algorithm determining the structuring of the paper on the fly at creation time to be effective it must be quick enough that the reader doesn't grow tired of waiting for his paper. This problem, as well as storage considerations, was a reason to choose against a neural network solution. Cluster analysis is a pretty good technique for finding groupings of elements when large, and often very diverse, bits of information are known about them. Unfortunately, the data points available about a user's reading habits don't seem to lend themselves to clustering. The order in which a reader reads his paper is an opportunity to derive **extremely** powerful feedback, and any algorithm, at least as a first pass, attempting to learn a user's reading habits should take advantage it. Based on these criteria, the structuring algorithm proposed is as follows:

- Generate the newspaper (Outside of Algorithm)
- Establish links between coexistent elements
- Increment coexistence counters for all coexistent elements
- Present paper to reader
- Each time an element is read, store that information (Gather Feedback)
- The next time a paper is generated
  - Increment counters indicating which of two linked elements was read first
  - Increment records of number of subelements presented and read

A representation of a relationships database showing only the coexistence links is shown in Figure 2.1. In the figure, each circle represents a different element in the database. The lines between elements indicate coexistence in an edition of the newspaper, and the numbers indicate the number of coexistences between those two elements. Two elements, Politics and Rwanda, have only appeared in papers with each other. That is, no occurrence of either of these elements has happened simultaneously with any other element but the other of the two. One element, Morbid, has no links.This means that no morbid articles have ever been presented to this reader. This is not unusual, as the determination of what elements will be interesting to a reader is often made independent of the available material.

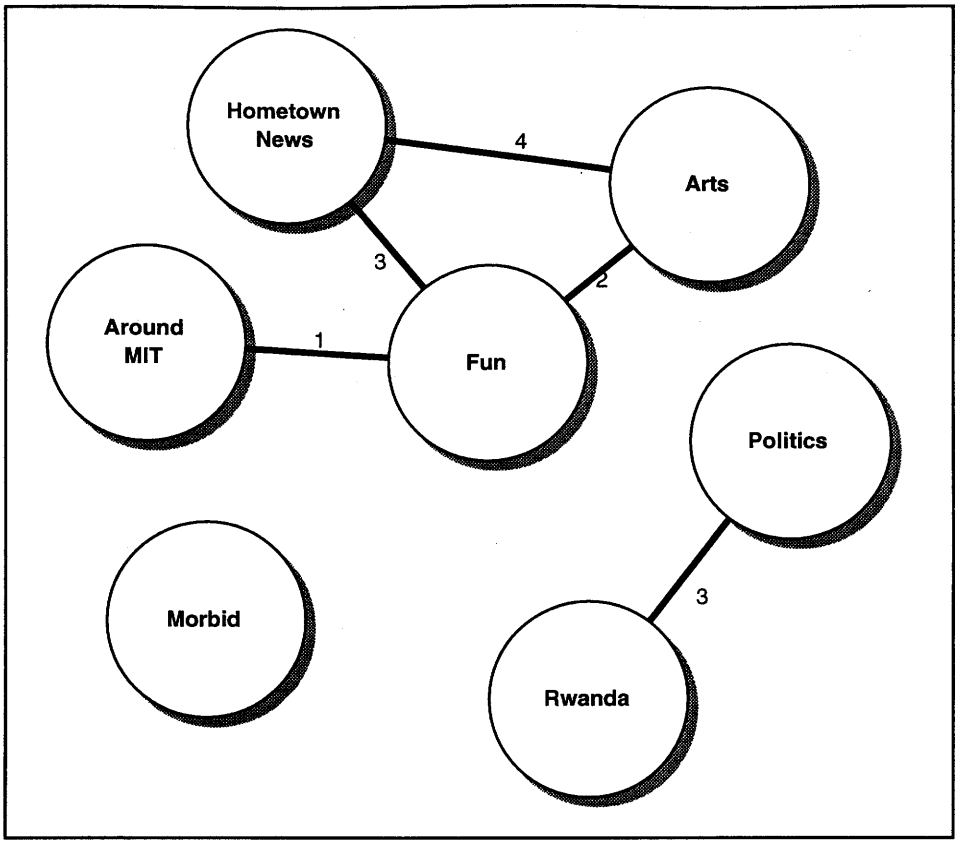

Figure 2.1: The Relationships Database

It just so happens that there has been nothing morbid in the news since this reader began using the algorithm. When an element is read, a *read first* counter is incremented in the link from the element to every other element in the current edition that has not yet been read (See Figure 2.2). These *read first* links are represented in the diagram by arrows, as they are inherently directional. This kind of link is owned by the element from which the arrow is based. Note that the numbers in these links will not necessarily add up to the number of coexistences between elements, as it is possible for two elements to be present in a paper a the same time and for **neither** of them to be viewed at all.

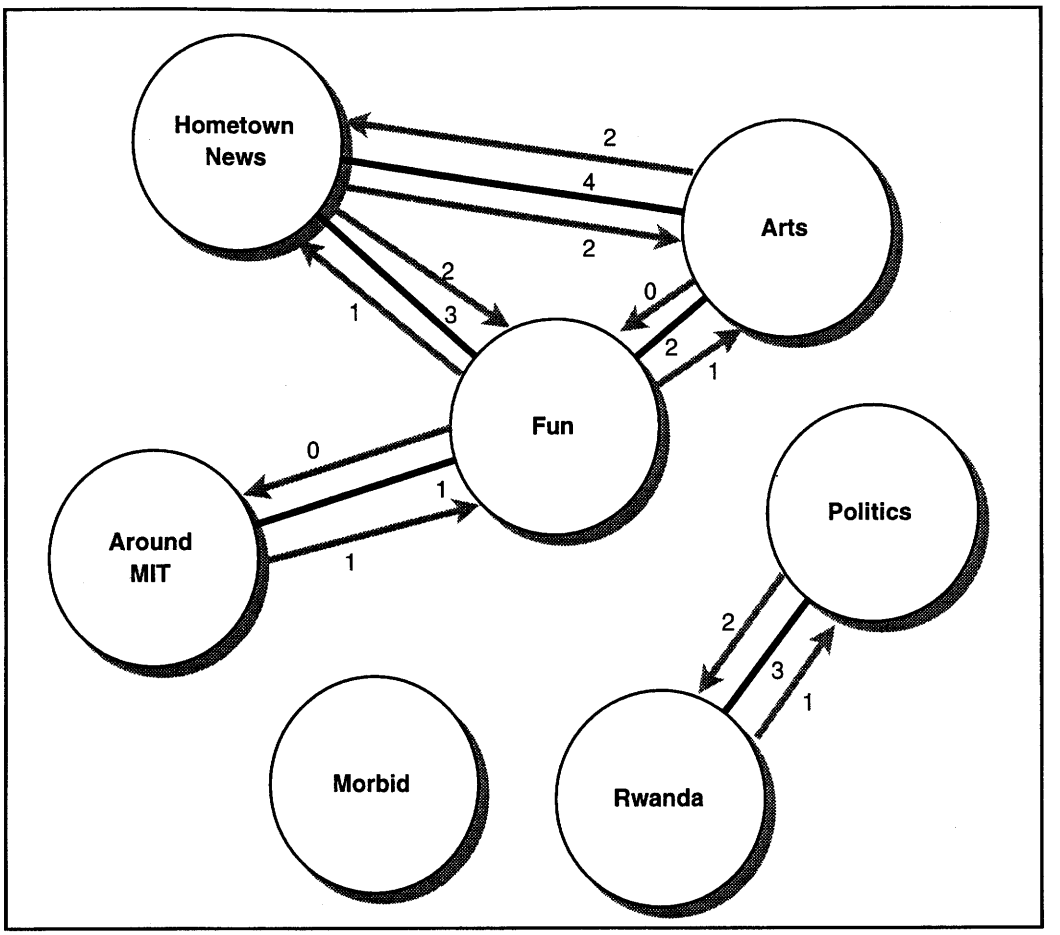

Figure 2.2: The Relationships Database with "read first" links

The above is the mechanism by which a database of relationships between newspaper elements is created. Each time a new edition of the paper is created, this database is used to determine the structure for its presentation by sorting the elements based on the ratio of presentations to *read first* counts. So whenever two elements appear in the paper simultaneously, the one that has been read first the greatest number of times that they both appeared simultaneously in the past is presented first. For example, with the database in Figure 2.2, the "Hometown News" would be presented before the "Fun" element because it was read first two-thirds of the time, while "Fun" was read first only one-third of the

time. "Fun" would be read before "Arts", however, because it has been read first half the time, while Arts has never been read before Fun. Similarly, "Around MIT" would be presented before "Fun" and "Politics" would be presented before "Rwanda". We don't have enough information at this point to deal with "Morbid", which is unconnected; to determine which of "Hometown News" and "Arts" is presented first; or to determine a relationship between the "Rwanda" and "Politics" subgraph and the rest of the elements.

This is the basic algorithm. The following modifications to it were made to allow for serendipitous encounters with new elements and for resolution of conflicts. If an element is appearing in the newspaper for the very first time--that is, if the reader has never seen this element in any previous paper before--then it is given an extremely high priority for one presentation and moved to the top of the display. This is done to make sure that the reader sees this new type of information, as the natural tendency based on the algorithm would be for it to appear at the bottom of the display where he may never see it. If he doesn't choose to read the new element, it will fall to its normal place in the algorithm's function. So this now allows us to deal with the "Morbid" element. It should go to the top of the paper because it hasn't been seen before. Additional criteria for resolution of conflicts between elements were made, as well. If the ratio of *read first* to presentation is equal between two elements, then the ratio of number of subelements ever presented for that type of element to the number of subelements ever read is compared between the two. The one which the highest subelement read ratio is considered more important than the other. If this criteria still turns up equality, then the total number of subelements presented is compared, and the one with the greater is given more priority. Failing this resolution principle, the two elements are declared equal. So now we know how to fit deal with the "Hometown News" and "Arts" dilemma. Whichever has a higher ratio of subelements read is placed first. If these are equal then the one with the most subelements total is placed

first. But this measure may be equal as well. It may sound overly flip to say that failing to determine a priority based on these criteria results in their being deemed equal. The result is, essentially, that which is displayed first is determined based entirely on which is first in the list of elements sent to the sorting routine. It must be recognized that this is a necessary evil. At some point we run out of mechanisms for comparing two elements. The "Politics" and "Rwanda" topics are connected to the rest of the graph in the same way. First their relationship to the others is determined based on ratio of subelements read to presented, then total subelements, and, failing that, randomly.

Chapter 3

Implementation of the Algorithm

3.1 The Freshman Fishwrap

The testbed used for this thesis was a project by the MIT Media Laboratory to provide personalized electronic newspapers to members of the Massachusetts Institute of Technology community that wished to participate in the experiment. The project began as "The Freshman Fishwrap" because its original directive was to provide a publication class that would benefit incoming freshman. The plan was to both provide these new members of the MIT community with news and information from their hometown, as well as to assist them in integrating into MIT and the Boston area. To this end, social and cultural information was collected about MIT surrounding neighborhoods. Also, several feeds from news organizations operating on both international and extremely local levels were acquired--such as the Associated Press, Knight-Ridder Tribune, the Boston Globe, and Reuters.

The flow information in Fishwrap is shown in Figure 3.1. The front end displays are different code segments that interact with the main fishwrap codebase to produce different renderings of the same data.

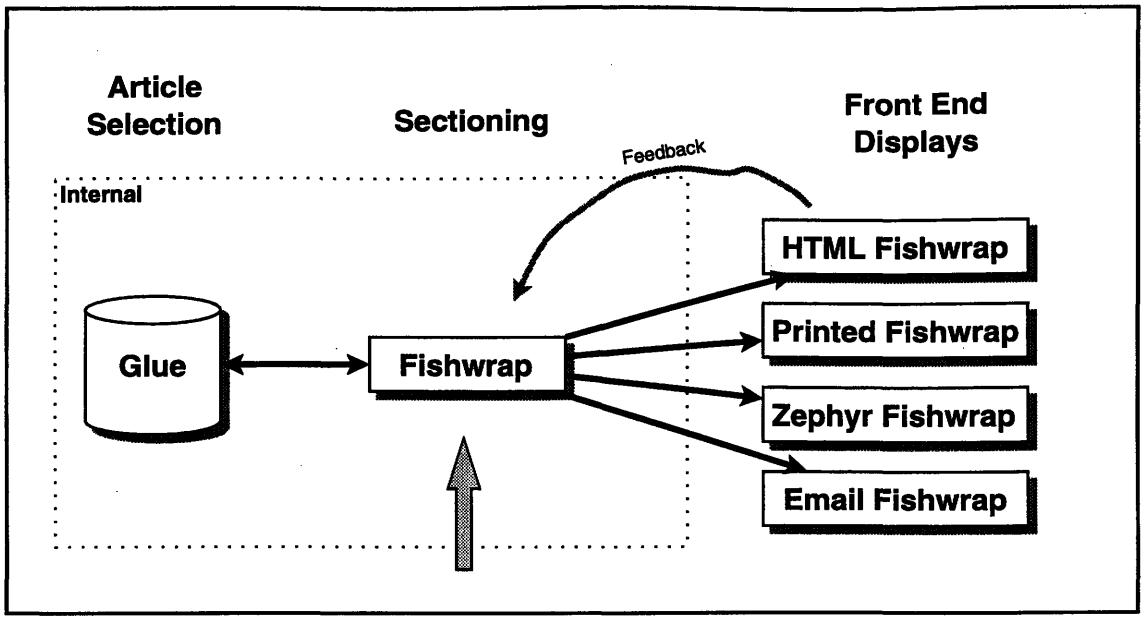

Figure 3.1: Flow of information in Fishwrap

Article selection is performed by an extensible application framework called "Glue", which takes care of interactions with user modeling system, news database article retrieval, knowledge representation, and article topic designation. The first two can be replaced at will with any system that meets Glue's specifications for the interface between itself and these elements. More detail about the jobs handled by and flow of information in Glue can be seen in Figure 3.2. For the purposes of Fishwrap, user modeling is not handled by the Doppelgänger system mentioned earlier, but instead by a much simpler user profiler. The reason for this is that the user modeling feedback possible with the current implementation of Fishwrap is limited to explicit user requests for the creation of newspaper sections and the addition of topics to those sections. Knowledge representation is handled in the form of a large set of news topics explicitly defined by the Fishwrap staff. The news server used for the project is a modified version of a server called "Betty" that is used internally for research at the Media Lab.

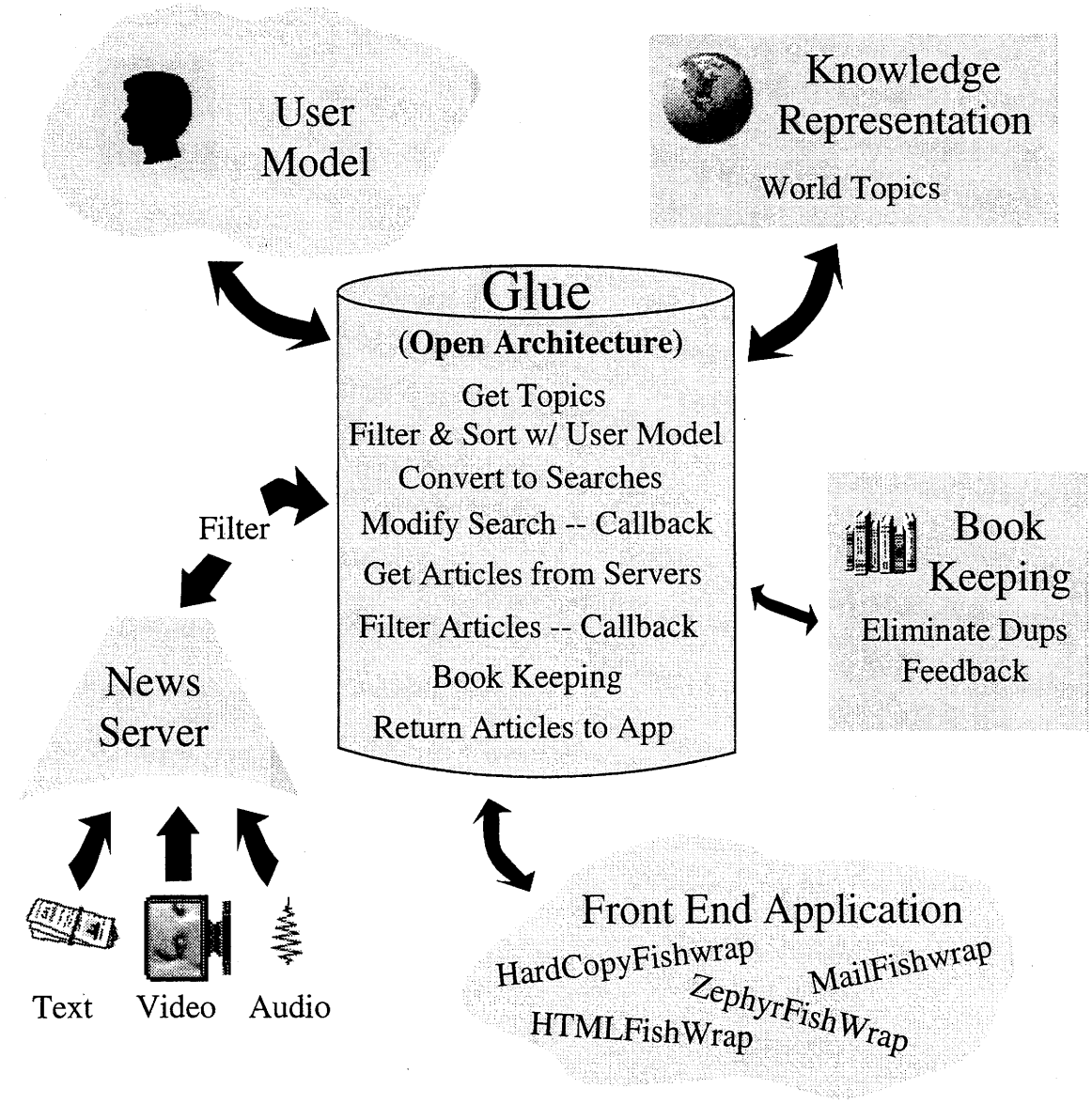

Figure 3.2: Information Paths in Glue

The most commonly used front-end application (or interface) to Fishwrap was the HTML version, which was accessed on the Athena network through an application called Mosaic. Mosaic is a popular hypertext document reader created by the National Center for Supercomputing Applications (NCSA). It allows documents to be viewed in a format called HyperText Markup Language (HTML) [HTMLPrime], which permits connections

to other documents to be inserted into a document such that clicking on particular spots in the document will result in the indicated document being retrieved and displayed on the screen. Links can also connect to documents that are not in HTML format. These documents can contain audio, video, or any other type of data. For the Fishwrap, the paper is organized around a table of contents page with links to take the user to various sections, topics, and articles in his paper. The flexibility of this document format also allows the Fishwrap to provide readers with photos along with the text of some news articles. This front-end to the Fishwrap project was also by far the most full-featured in terms of its ability to interact and be responsive to the reader's needs.

Potential readers were asked to consent to participate in the experiment, after which they were allowed access to a wide range of features. Readers could get regional news from their hometown or from other areas in the United States or around the world that interested them. This feature was implemented with a potential for localization down to a zipcode level. Readers were also allowed to create the basic structure of their paper. Using a tool called PaperBuilder, a user could create sections to add to their personalized newspaper. A wide range of topics, defined by the Fishwrap staff, were available for inclusion in these user defined sections.

3.2 Using The Algorithm in Fishwrap

In the flow information in Fishwrap, the algorithm was inserted at the location specified by the gray arrow coming from the bottom of Figure 3.1. Another aspect of the implementation of the algorithm is the gray arrow labelled "feedback" that comes from the front-end application back into the Fishwrap's back-end code. This feedback loop represents the

insertion of information derived from the user's reading session into the determination of the structure of future editions of the newspaper.

For use with The Fishwrap, the algorithm was implemented in ANSI C. Abramson's ANSI Dtype library [Abramson92], a library for allowing flexible data storage and manipulation using Lisp-like data structures and tools, was used for storage of the relationships database. The database entry for "Hometown News" from Figure 2.2 on page 13 looks as follows in Dtype format (see Figure 3.3:).

```
("Hometown News"
   (("links"
      (("Fun"
         ("both_present" 3)
         ("read_before" 2))
       ("Arts"
         ("both_present" 4)
         ("read_first" 2)))
    ("present" 5)
    ("subelements_presented" 107)
    ("subelements_read" 34)
    ("first_time" NO)))
```

Figure 3.3: A Relationships Database Entry in Dtype Format

In the actual database the "Hometown News", "Fun", and "Arts" elements are represented by a unique numeric identifier, and the "first_time" field's value is represented as a numeric constant representing "NO". This database was manipulated using Abramson's Dtype manipulation library.

Generating a Newspaper

When a user runs the SelfOrganizing version of Fishwrap a paper is generated for him using Glue, then the sectioning of the topics is done based on the information the user has provided using the PaperBuilder. Then the paper goes through the structuring algorithm described above, where both sections and their subtopics are reordered based on the current status of the database. The process of selecting articles, sectioning them, and restructuring the paper takes approximately five minutes for a user with an average sized paper. After generation, the paper is presented to the user. It will look like Figure 3.4.

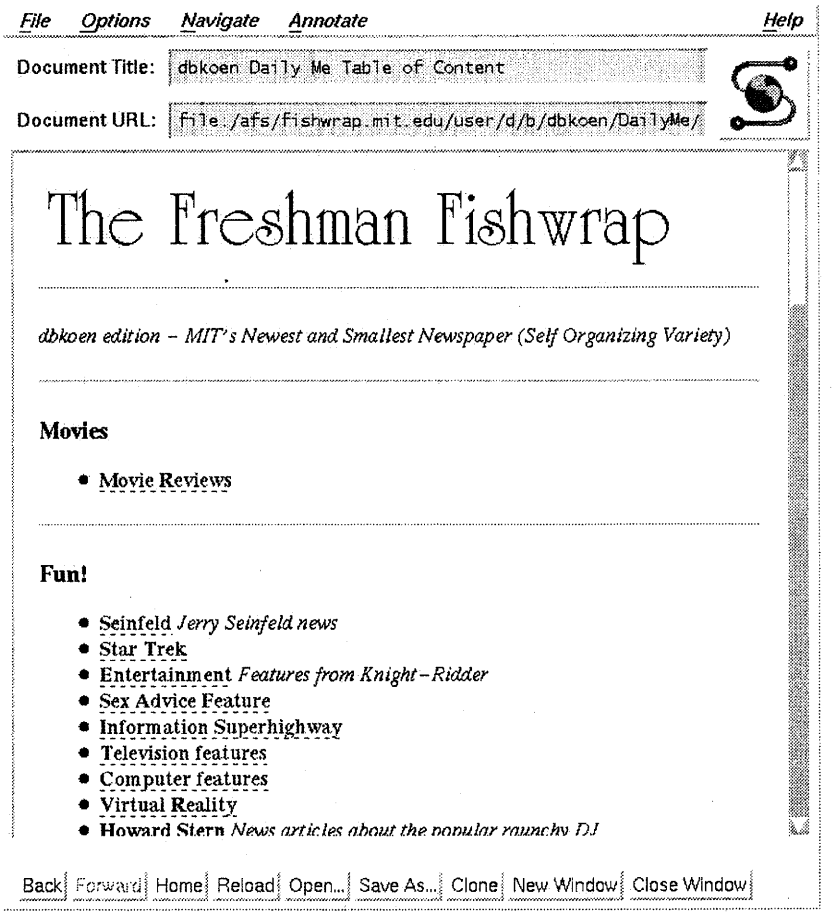

Figure 3.4: The Table of Contents Page of a Fishwrap SelfOrganizing Paper

Any text on the page that is underlined can be clicked to move to a different document. In Fishwrap, the headings like "Movies" are Sections, and the bulletted, underlined elements underneath are the topics in the section that contain articles for this edition of the paper.

Acquiring Feedback

Mosaic is not designed to be an application that interacts heavily with external programs,so certain of its features had to be utilized in a somewhat unique way. Mosaic has the ability to launch an external viewer for document types. These external viewers are specified in a table that matches a filename extension to an application that knows how to display that file type. The implementation of Mosaic for the X Windows windowing system (the windowing system used on Athena workstations) has an additional feature which implements a **very rough** capability for driving the Mosaic application externally. This is done by writing a particular file in a particular place that contains a single command to Mosaic and the path to a document to retrieve. Once this file is created, the external application signals the Mosaic application, which reads the file and performs the requested operation. The Fishwrap staff combined these two facilities by creating a special document type that would launch a script of their crafting to dispatch commands to the Fishwrap programs. This facility was used to record the user's actions as he manipulated his SelfOrganizing Paper. Specifically, each time a user entered a new section of his paper it was recorded, as well as each time he opened up a news article. In Figure 3.5 the user has

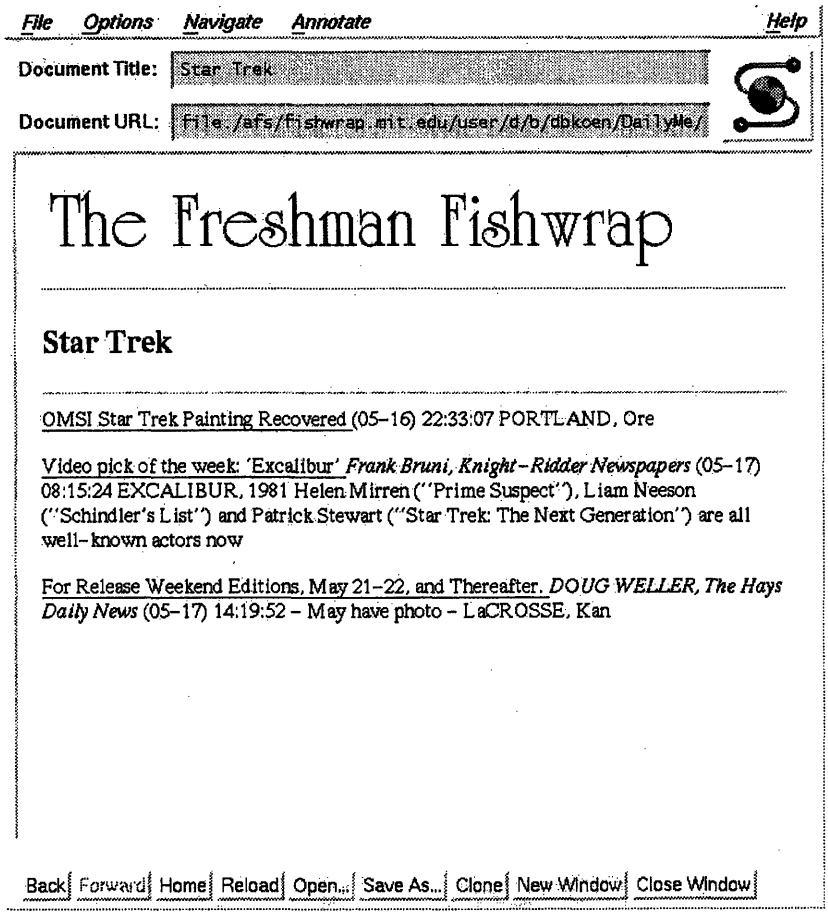

Figure 3.5: The Star Trek Section From The Paper In Figure 3.4

clicked on the Star Trek topic in the Fun! category of his Table of Contents page. Visually, this just brings up the topic contents page for the Star Trek topic, which contains the articles available in that topic for this edition. Behind the scenes, the link on the Table of Contents page does not point to the topic contents page, but rather at a file in the special fishwrap dispatch format, as described above. The flow control when the user clicked on Star Trek is shown in Figure 3.6. In the box on the bottom right information a small program spawned by the Fishwrap dispatcher records information about both the topic chosen and the section the topic appeared in. Then, the next page to display is determined (in this implementation it is determined from information stored in the special Fishwrap dispatch

format), and Mosaic is signalled to display the next page.

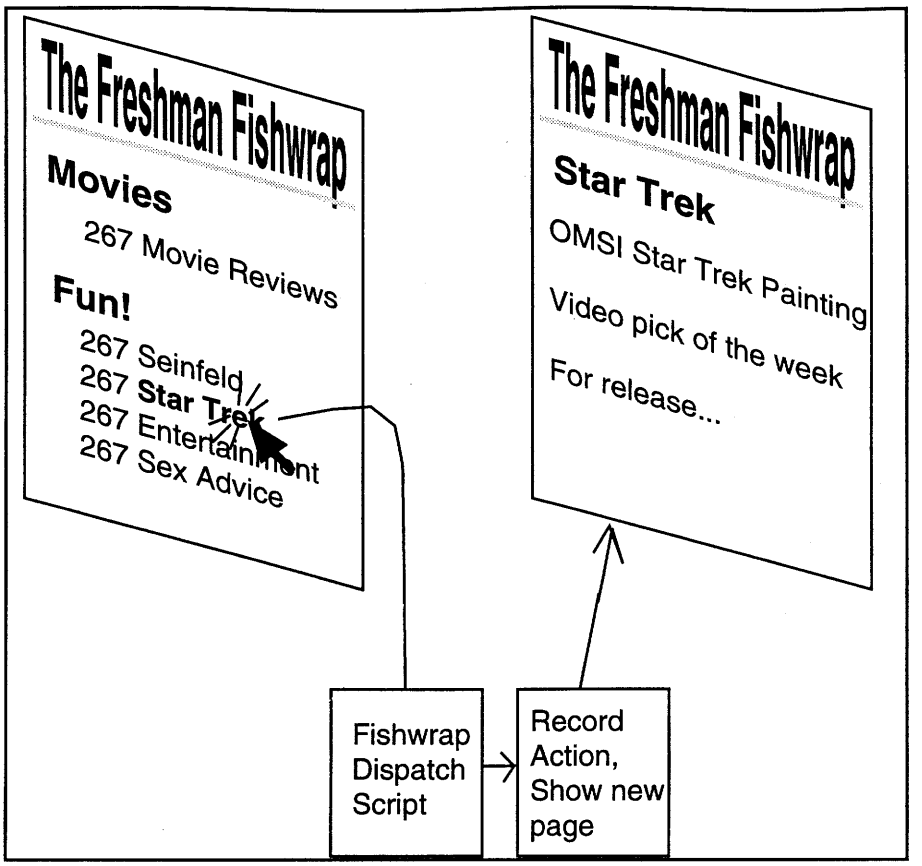

Figure 3.6: Flow Of Control Upon Topic Mouse Click

Incorporating Feedback

The next time the reader requests a paper generation, the first step of the generation software is actually to incorporate the feedback recorded into temporary files during the last session into the relationships database. Then it proceeds to the actual generation of the next edition of the paper as requested.

3.3 Results

Analysis of the success of the Fishwrap-based experiment was done by interviewing users of the Fishwrap and the SelfOrganizing Paper in focus groups with five to eight participants and two or three members of the Fishwrap project staff. This method seems appropriate, because the true successes and failures of a project of this type are best determined by the responses of users who are not familiar with the technical aspects of the implementation of the software they are using, as this is the kind of environment in which a system of this type would be expected to operate.

One problem specified by the members of the focus groups was that it was somewhat unrewarding and cumbersome to require them to peruse the entire paper to find the sections and topics they would most like to read when sometimes it would just be easier to read the topics in the order they are presented from the beginning. This is an unfortunate side effect of this particular implementation of the algorithm, and is discussed in Future Directions under the heading "Inertia". One user said that he could detect no difference in the structure of her paper after continued use of the SelfOrganizing system. This is a difficult problem to respond to because in its simplest form the algorithm may not please the user, but it **does** change the structure of the paper provided that the user does not suffer from inertia. Another problem with the algorithm that may be in evidence in this case is the lack of any mechanism for decay (see Future Directions under "Time Based Analysis -- Decaying Relationships"). If this user had reinforced the existing ordering repeatedly, and then changed the way he read his paper to test the algorithm, it would take him a very long time to overcome the algorithm's well-reinforced belief that the previous ordering was correct. Another user suggested that the problem of paper structuring could be satisfactorily solved by providing the user with a mechanism for moving the topics and sec-

tions around manually, and in this way he would be able to select the paper structure that appealed to him. While this solution might be equitable for technically inclined users, the objective of this research is to increase automation and responsiveness to the reader's wishes without the need for explicit intervention on his part.

The majority of regular users of the SelfOrganizing paper that I spoke to found that the system did accurately reflect the reading habits that they exhibited over time. They expressed that sections and topics commonly read early moved rapidly to the front of their paper. They also tended to enjoy the fact that a new, previously unseen, topic would initially appear at the top of the paper for their approval; after which it would retire to the bottom if they did not demonstrate a desire for its continued prominence.

It was not clearly established from these interviews if the first purpose of this thesis was satisfactorily established--is the decoupling of article selection from structure determination a valuable approach? The positive responses of the users indicates the success of the algorithm, but does not clearly indicate that this same success could not be achieved by a similar algorithm that includes aspect of the article selection in making structuring determinations.

Chapter 4

Future Directions

This section is in two parts. The first is geared towards ideas that are attempts to solve problems discovered in the algorithm as a result of the experiments. The second section is geared towards future research directions that are extensions of this work, but not in response to any particular problems.

Solving Problems Detected Through Fishwrap Experiment

4.1 Inertia

One problem with the Fishwrap based implementation of the algorithm is the result of the fact that, as could be seen in Figure 3.4, the reader is presented with only a small fraction of the topics in his entire paper on the visible portion of his "contents page". The rest of the Table of Contents is offscreen and the user has to use the scrollbar to get to it. A tendency that users have demonstrated is to only consider the articles visible on the first page when making an initial selection of topics to read. Only after looking at all of the interesting topics visible initially does the user scroll the page down to see what else is available. This tends to generate a cycle in which the sections and topics appearing at the top of the page stay there because they are always selected before the ones on the bottom part of the page (this resembles the notions of "above the fold" and "below the fold" from conventional newspapers). This problem might be solvable by creating a more concise initial page for the reader. Perhaps a page that showed him the sections and a one line list of the topics available in that section might lead to a more effective implementation of the struc-

turing algorithm.

4.2 Time Based Analysis -- Decaying Relationships

Another problem with the algorithm as it was implemented is that the information in the inference database is not time and date stamped. Because most news is by nature ephemeral, a reader could reinforce an element's ordering in the paper structure as a result of events in that region that are extremely interesting to him for a very specific reason. By continuing reinforcement of that element's placement, it will become less and less likely to move. If the special circumstance that made that item interesting for, say, a month, are gone then the reader must reinforce alternative placements of that element for the same amount of time as he reinforced it to get it placed there. An example of the situation described above is as follows. A reader has a relative living temporarily Rwanda. For the span of time that she is in that country, the news and events from Rwanda are more important to him. The minute she comes home they become substantially less important to him, but he must suggest alternative placements to the algorithm just as many times as he reinforced the existing one for it to be counteracted. A possible solution to this problem that may be worth exploring comes to mind. One is to cause reinforcements to be time-stamped and automatically decay. That is, with the existing algorithm if I reinforced the placement of element A before B the same number of times as I reinforced the placement of element B before element A, then they would be tied and the next conflict resolution principle would be used. However, if the instances in which each of these orderings was reinforced were time-stamped, then they could be weighted based on age to simulate decay of the reinforcement.

Research Ideas To Extend This Work

4.3 Front-End Applications -- Other Than HTML

This algorithm currently only works if a reader makes use of the HTML front-end for Fishwrap. This is because the other possible methods of reading the paper--hardcopy, zephyr, and email--have no capability for intelligence at the time of reading. That is, it is a difficult problem to get feedback to the structuring system from a newspaper printed on paper by any other mechanism than an explicit message from the user because ink on paper cannot automatically record your reading habits for future incorporation into the relationships database. Because computers are currently far from ideal mediums for accessing information--they can't be folded up and carried on the subway, nor can you comfortably can't lie down in your back in bed and read from one--many people will continue to want their papers delivered to them in hardcopy format. Does this method of distribution completely short-circuit attempts to augment the news reading process with computers? Certainly not. Article selection can clearly still be handled by computer based on a profile of the user. If the reader is willing to read his news in front of a computer when he's able to, then structuring information can be gathered from these sessions and incorporated into the paper that is presented to him when he is not at his terminal. This approach is already being explored by Schoon with "Fishpaper", the hard copy front end to Fishwrap [Schoon94].

4.4 Time Based Analysis -- Editions

Another interesting area to explore is modifications of paper structure based on the user's environment at the time that he generates his paper. Acquiring this kind of information falls squarely in the domain of user modeling. A first pass at analysis of the user's environment when he generates his paper might be to take into consideration the time when the generation occurs. For example, if a reader generates himself a "morning edition" of his paper and an "evening edition", then it well may be that the structure for his paper should change--he may want entertainment news first in the morning because he doesn't feel like dealing with the bad news and in the evening he may be ready to sit back and find out about the latest occurrences of random violence around the globe. Even if a user doesn't generate two papers a day, creating different paper structures for different times of day could still be valid. Indeed, including analysis of the day of the week or day of the month on which the paper is generated might yield differences in the desired newspaper structure. While this is a reasonable first pass at addressing environmental considerations, in fact the actual **time** of paper generation is probably not as valuable as other aspects of his environment. In truth, my "morning edition" is defined not by the time of day so much as, perhaps, when I wake up or when I'm eating breakfast. Possibly at MIT more than any other place in the United States, the notion of morning is incredibly flexible, being almost completely independent of time of day. Readers of the Fishwrap in our experiment generated papers at essentially every hour of the day. From the time of day it is completely impossible to determine if those students had awakened fifteen minutes before or if they were about to go to sleep after having been awake the entire night before. If we could find out when a reader's had free time from our user modeling software, we might be able to structure his paper such that the Arts and Entertainment section could be more prominent

so that he could find a way to spend his free time. If we know that he is extremely busy, then we might be inclined to provide him with a quick synopsis of the news so that he can get back to work. Or; if our notion of "news" is expanded to include information that is not of national or large scale importance, but of importance to our reader or members of a community of which he is a part; we might be inclined to provide him with vital information about the classes that are giving him trouble. One possible method of implementing some of these ideas would be to do a basic cluster analysis (based on a Minkowski metric for determining temporal proximity) on accumulated paper generation data.

4.5 Editionless Structuring

Another area of potentially worthwhile research would be the investigation of responsive structuring methods that do not rely on the notion of editions. The algorithm implemented in this thesis depends on a comparison of paper elements that coexist in a single edition of the paper, as was the case with the testbed used for its analysis. With electronic publication comes the potential for a completely dynamic system--a system that is continuously running and continuously updating itself. While the differences with this kind of system may seem vast, simple modifications to the algorithm could result in its continued success in this environment. Despite the fact that the system as a whole is dynamic, the choice of which subelement to read is made by the reader at a particular instant in time. At that instant, he is making a choice in favor of a particular element over all of the other options. Additional analysis of the effects on the algorithm of a system without editions is warranted.

Acknowledgements

There are so many people that deserve my appreciation for their support and direct assistance throughout this thesis that I am wary of making a list because because I know that I will inevitably leave someone out. Nonetheless, to fail to acknowledge the following people would be criminal, as without them this thesis could never have been written.

• Pascal Chesnais, my thesis supervisor, UROP supervisor, and friend; who has given me unending assistance and encouragement through some very difficult times. He was also the first gentleman I had the good fortune to meet the first time I ever entered the Media Lab. Without his constant ideas and ability to critically analyze and pin down the good ideas in hours of rambling, this thesis would have greatly suffered in focus and direction.

• Walter Bender, an Associate Director of the Media Lab; who started me on the road to this thesis when he agreed to hire me through the UROP program when I was a freshman, and who is such an incredible source of interesting new ideas that just being around him forces you to think.

• My fellow UROPers in support of Fishwrap and the rest of the framework that made this thesis possible, including but not limited to: Jonathan Sheena, Brad Bartley, Michelle McDonald, Brian Shea, Adam Cotner, James Deverell, Matthew Mucklo, and Benjamin Schoon.

• Trip DuBard, one of the "editors" of the Fishwrap, for his non-technical input on Fishwrap, my thesis, and related issues.

• Graduate students and UROPers past and present who not only made the Garden a great place to work, but also provided substantial support and infrastructure that made this thesis possible. Including, but not limited to: Jon Orwant, Nathan Abramson, Klee Dienes, Laura Teodosio, Mark Kortekaas, Erik Kay, and Håkon Lie.

• And, of course, my parents and brother. If you know them no explanation is necessary; if you don't none is adequate.

References

[Abramson92] Abramson, Nathan. The Dtype Library or, How to Write a Server in Less Time than it Takes to Read This Manual. Technical report, Electronic Publishing Group, MIT Media Laboratory, December 1992.

[BC88] Bender, Walter and Chesnais, Pascal. Network Plus. In *Proceedings, SPIE Electronic Imaging Devices and Systems Symposium*, pages 81-86, January 1988.

[Bender93] Bender, Walter. Riding the Digital Highway. *Presstime*, May 1993.

[BLBDH85] Bender, Walter; Lippman, Andrew; Bove, V. Michael Jr., Donath, Judith; Halle, Mike. Electronic Newspaper. In Home Computing: Year End Report to IBM's Entry Systems Division. MIT Media Laboratory, December 1985.

[Blount91a] Blount, Alan Wayne. Self-Organizing News. Master's Thesis, MIT Media Laboratory, 1991.

[Blount91b] Blount, Alan Wayne. Bettyserver: More news than you can beat with a stick. Technical report, Electronic Publishing Group, MIT Media Laboratory, December 1991.

[BCDHN91] Bender, Walter; Cooper, Muriel; Davenport, Glorianna; Haase, Ken; and Negroponte, Nicholas. The Newspaper of the Future: A Straw-Man Proposal in Four Parts. Technical report, MIT Media Laboratory, July 1991.

[CK93] Chesnais, Pascal and Koen, Douglas. Strategies for personal dynamic systems: News in the future. In *NextWORLD Expo*, 1993.

[Donath83] Donath, Judith S. The Electronic Newstand: Design of an Intelligent Interface to a Variety of News Sources in Several Media. Master's Thesis, MIT Media Laboratory, 1986.

[FISH] The Fishwrap Information Page. Available on the World Wide Web at Universal Resource Location:
http://fishwrap.mit.edu/

[HTMLPrime] HTML Primer. Available on the World Wide Web at Universal Resource Location:
http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html

[LB87] Lippman, A; Bender, W. *News and Movies in the 50 Megabit Living Room.* Paper presented at Globecom, IEEE, Tokyo, Japan, 1987.

[Lie91] Lie, Hakon. The Electronic Broadsheet--all the news that fits the display. Master's Thesis, MIT Media Laboratory, 1991.

[Lippman86] Lippman, Andrew. Electronic Publishing. MIT Media Laboratory Memo, January 1986.

[Orwant91] Orwant, Jon. Doppelgänger: A User Modeling System. Bachelor's Thesis, MIT Department of Electrical Engineering and Computer Science, 1991.

[Orwant93] Orwant, Jon. Doppelgänger Goes To School: Machine Learning for User Modeling. Master's Thesis, MIT Media Laboratory, 1993.

[Schoon94] Schoon, Benjamin. Fishpaper: Automatic Personalized Newspaper Layout. Bachelor's Thesis, MIT Media Laboratory, 1994.

[Smith80] Smith, A. Goodbye to Gutenberg: *The Newspaper Revolution of the 1980s*; Oxford 1980.