

Model Planes and Totem Poles: Methods for Visualizing User Models

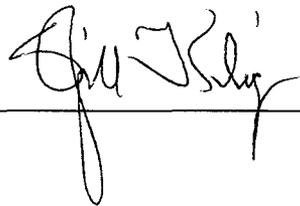
by
Jill Susan Kliger

B.S., Computer Science
The University of Michigan, Ann Arbor MI
May 1992

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
at the
Massachusetts Institute of Technology
June 1995

Copyright Massachusetts Institute of Technology, 1995
All Rights Reserved

Signature of Author



5/15/95

Program in Media Arts and Sciences
12 May 1995

Certified by



Walter Bender
Associate Director of Information Technology
MIT Media Laboratory
Thesis Supervisor

Accepted by

Stephen A. Benton
Chairperson
Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Model Planes and Totem Poles: Methods for Visualizing User Models by Jill Susan Kliger

The following people served as readers for this thesis:

Reader: Glorianna Davenport 5/12/95

Glorianna Davenport
Associate Professor of Media Technology
Program in Media Arts and Sciences

Reader: D. Riecken 5/9/95

R. Douglas Riecken
Principal Research Scientist
AT&T Bell Laboratories

Foreward & Acknowledgments

This work has provided a good departure from my previous focus on hypermedia for education. The field of user modelling is growing rapidly and I hope the concerns of the end-user are not overlooked. During the process of building PeerGlass I have learned much about visual representation, graphics programming, project management, and working with focus groups. I hope this document sheds light on ways to represent user models to end-users both in terms of the content and graphic design. The section on potential features contains insight on where future work in user modelling should be directed.

A number of people have supplied input and support for this work. The direction, criticism, and suggestions provided by Walter Bender have significantly affected the evolution and improvement of this work. Glorianna Davenport has provided much insight on the needs of news readers, and strong direction in the creation of a videotape of PeerGlass. Doug Riecken's advice on how to craft ideas into a functioning system, along with his direction on the focus of the project have been invaluable.

In the Garden, Michelle McDonald and Klee Dienes have been endlessly patient with their design criticism and programming advice. Henry Holtzman and Shawn Becker were most helpful with technical support of the SGI and advice on Open Inventor.

In the Visible Language Workshop, Earl Rennison has provided suggestions and advice since the outset of this work. Lisa Strausfeld and David Allport have also been helpful with technical and design input.

The twenty-two people who participated in the evaluation of PeerGlass supplied generous quantities of criticism, vision, advice, suggestions, and time. Their involvement significantly improved the quality of this work.

My family, friends, and colleagues around the country who have provided moral support and direction for the past two years can not be thanked enough.

Trademarks contained herein are the property of their respective owners. Words that are copyrighted or trademarked are capitalized in this document. These include but are not limited to: Silicon Graphics (SGI), Open Inventor, Motif, Rolodex, NeXT, Macromedia Director, Diagram!2, Boston Globe, New York Times, Psion, and Byte. No infringement of copyright or trademark is intended.

Table of Contents

Section 1: Description

Document Overview	6
Features & Motivations	10
Figures 1 to 10	15
Evolution & Inspirations	22
Figures 11 to 21	27
Evaluation: The Surveys	35
Evaluation: Future Issues	44
Evaluation: New Designs	54
Retrospective	68
Bibliography	71

Section 2: Implementation

	73-78
Basic Flow of the System	
Overview of Classes	
Article Format	
User Model Format	
Adding New Model Planes	
Navigating the Model Planes	
Description of User Interaction	
User Input Features	
Enhancing the Newspaper	
Upgrade Issues	

Color figures are available on-line from the author's page at <http://www.media.mit.edu>.

Document Overview

In the near future more data collection devices will exist to monitor an individual's actions and interests. This data is useful for tailoring information to the individual consumer. A user model is a growing set of data that describes an individual. For many years user models have been a tool for direct marketing organizations to target magazines and other mailings at consumers who have previously shown interest in similar products. Currently, the user modelling done by direct marketers is crude and inundates the consumer with much unwanted mail. Assuming that better user modelling methods exist and are focused on helping consumers filter information down to their needs, the near future will bring many personalized systems to market. News, entertainment, and correspondence are each ripe fields for personalization.

Regardless of the domain, consumers will not use a personalized system if they do not trust it. An executive would not allow a new secretary to shred "non-useful" mail and block "non-important" phone calls, if the executive did not trust the new secretary. To trust a filtering mechanism it is necessary to feel in control of the filter's behavior and to understand some of the criteria under which the filter operates.

PeerGlass is an architecture for visualizing user models. The concept and design can be applied to any personalized information system. The work described in this document is an application of PeerGlass to an electronic, personalized newspaper that provides users with access to their user model. The intent of PeerGlass is to help users understand their user model so that trust between the users and the system grows. PeerGlass incorporates two methods for visualizing user models. The first involves graphical widgets called "totem poles" to help answer the question, "Why did the system choose this article for me?" The second mechanism is a set of "model planes" to explain, "What does the user modelling system think it knows about me?" The totem poles are customized for each article while the model planes are particular to each section of the newspaper.

The following overview briefly introduces the key points that appear in the rest of this document.

Previous Work in Representing User Models

Earlier work in user modelling and agents has tried to answer these questions by providing the user with historical reasoning. That is, if the user asks, “Why did you do this?” the answer is a specific reference to a similar action the user made at a previous time. In case that is insufficient, other systems have allowed users to access an entire historical log of their interactions with the system. The cause of these seemingly dry responses is that user modelling information is often simply a set of statistics that are either taken from a user’s explicitly declared interests or from inferences based on observations of the user’s behavior. These statistics are not necessarily interesting to look at or even decipherable to people who are not experts in user modelling. Earlier versions of PeerGlass represented users’ preferences with a flat-screen approach that emphasized control and feedback functionality. The 3D totem poles and model planes of the current version of PeerGlass aim at helping people understand how the user modelling impacts their news.

Seeing Why an Article was Selected: Totem Poles

Totem poles show the user the strongest matches between the features of an article and the preferences in their user model. The goal is to explain why a particular article made it into the personalized newspaper. A totem pole is a multi-sided 3D widget that can be freely rotated or controlled through prescribed camera angles. The features of an article are grouped into four categories. The four sides of the totem pole correspond to the four categories:

Source: Boston Globe, Wall Street Journal...

Topic: Basketball, Bosnia, Clinton...

Type: Interview, Expose, Press Release...

Nature: Liberal, Humorous, Depressing...

By looking at the totem pole for an article, the user should be able to quickly determine why the article was selected by the user modelling system. Totem poles also have the [non-intentional] function of helping the user browse through articles.

Exploring the User Model: Model Planes

Each model plane is a different slice of the user's model for a section of the newspaper. The model planes are organized along the same axis to form a Rolodex-like mechanism. The user spins the Rolodex to see the planes. The model planes that are currently implemented in PeerGlass include:

Explicit Interests: items that the user tells the system to include

Explicit Expert Sources: a list of publications that the user explicitly trusts

Implicit Interests: the interests that the user modelling system has inferred by watching the user

Implicit Communities: the observed set of communities that the user modelling system has inferred to be close to the user's interests

Users who want to know the contents of their user model can explore the model planes to find this information and learn about how the filtering is done. The architecture of the system supports the addition of more planes to represent other angles of user modelling.

Implementation Overview

The electronic newspaper that uses PeerGlass as an architecture for visualizing user models is more of a framework than a fully functional system. The articles that are filtered have been pre-tagged for the approximate Source, Topic, Type, and Nature of their content. If articles coming off of a live feed were appropriately tagged, the newspaper system could read them but currently the articles exist as archived files. The tags are not simply based on keywords that appear in the article. For instance, it is very unlikely that the word "depressing" would appear in a depressing article. Automatically tagging the features correctly requires an algorithm that does more than find keywords.

The newspaper system compares the available articles and user models to select the articles for a personalized newspaper. Neither the process for comparing this data nor the process for laying out the newspaper is sophisticated but both are sufficient for the current needs.

PeerGlass visualizes the user models of the newspaper's readers. The user modelling data is stored in a file that is not updated, nevertheless PeerGlass is ready to handle any user modelling data that is in a prescribed format. Prior to selecting the articles,

PeerGlass builds a set of model planes for each section of the newspaper. A totem pole is created for each article once the article surpasses a threshold for relevance to the user model. Adding a new type of model plane would require the addition of a new plane-layout routine and an update to the parse routines for the user model.

PeerGlass runs on a Silicon Graphics workstation and has a user interface that combines Motif and Open Inventor.

Extensibility

Although the current implementation of PeerGlass is particular to personalized newspapers the concept can be generalized to other domains. If the content were television listings, the totem poles could be re-designed to address Actors, Topic, Type, and Reviews. The intent of the totem pole would remain the same: to help viewers understand why a particular TV show or movie was selected for them. Most types of model planes (explicit, implicit, communities, critics...) are applicable across domains. Envisioning how to apply the PeerGlass framework to e-mail filters, song advisors, Web searchers, and other content-overwhelming fields simply requires defining the sides of the totem poles and the necessary types of model planes. Regardless of the kind of algorithm used in the modelling process, there exists a need to explain to users how personalized content was selected and what are the contents of their user model.

Evaluation

Since the point of PeerGlass is to help people understand their user models, it is necessary to investigate which parts of PeerGlass make sense to users and those aspects that need revision. Twelve graduate students from the MIT Media Lab reviewed PeerGlass and provided their suggestions on future improvements. The students were selected based on their specializations in visual design, interface agents, epistemology, and/or human-computer interaction; members of the Electronic Publishing group were excluded due to their closeness to the project. While reviewing the system in pairs, the students rated how effective they thought the totem poles and model planes were at meeting their goals. These dozen evaluators later participated in focus group meetings to discuss the value of potential additions to PeerGlass and to propose improved designs for the model planes. Five pairs of undergraduates subsequently evaluated the system and reviewed the suggestions from the focus group meetings.

Features & Motivations

Approach

The design of PeerGlass is centered on the idea that people will not trust[21] a personalized system unless they understand how the filtering is done. To this end, PeerGlass provides support for users to explore their own user model. It is unlikely that users will frequently look at their user model. Users will probably explore their user model in the following situations:

1. The users want to find out why an unwanted article was selected by the filter.
2. The users have a basic curiosity about their user model and the filtering process.
3. The users want to expand their user model and re-define their interests.

PeerGlass addresses Situation 1 with totem poles to help the user understand why an article was selected. Each totem pole lists the strongest matches between an article and the user model. The four sides of the totem pole correspond to four ways to view an article's content: Source (Figure 1), Topic (Figure 2), Type (Figure 3), Nature (Figure 4). Totem poles are an article-specific view of the user's model, while the model planes are specific to a section of the newspaper.

The user in Situation 2 who is trying to understand the criteria for filtering is aided by the set of model planes. Each model plane is a different slice of the user's model. During the filtering process, article features are compared to the items on each plane. Before it is fair to claim that PeerGlass sufficiently meets the needs of users in each of these three situations, it is necessary to add other features to the system. The *Evaluation: Future Issues* section of this document provides more detail.

User Interaction with the Newspaper

Articles and user modelling data are stored on the computer at all times. When PeerGlass is launched, the system selects articles based on the user's model and then displays the first section of the newspaper. The user can change sections by clicking from a list along the left edge. Since the text of the articles is cropped to fit a tight layout,

there is a pop-up window for viewing the full text of the article. Clicking on the article, its headline, or its totem pole will bring up the pop-up window. Along the bottom of the main window there is context-sensitive instructive text for the operation of the interface. See Figure 5.

Totem poles for each article in the section appear between the pushbuttons and the articles. Users can look at different sides of the totem pole by grabbing and turning the totem pole with the mouse. The arrow keys provide a smoother interaction because they put the viewer at a perfect camera angle to see the next/previous side of the totem pole.

At any time, the user can access the set of model planes for the current section. By clicking the “Examine Profile” pushbutton, the text of the articles disappears from the screen and is replaced by a view of the model planes for the section. See Figure 6. The model planes are organized along the same axis to form a Rolodex-like mechanism. The user has two interface options for viewing the planes. There is a dial widget for spinning the Rolodex incrementally with the mouse, and arrow-key support for viewing the next/previous model plane.

Reality Check

The current implementation of PeerGlass has very limited functionality but is designed to be extensible. The realities of the program are described here:

First, the system loads the set of all articles from a particular directory. The approximate Source, Topic, Type, and Nature of each article is manually tagged. PeerGlass can handle any articles or user modelling data files that appear in a prescribed format.

Second, the user modelling data files are organized according to the content of the model planes for each section of the newspaper. The user model is built from the contents of user modelling data files. The articles are filtered using a simple multi-variable equation that compares the tags of the articles to every model plane in the user model. Each match between an article and the user model is weighted (e.g, a match between the article and an explicit preference would receive more weight than a match between the article and an implicit, community-based interest). The list of matches is stored along with the article. A score is computed for the article by totaling the weights of the matches on the list. If the score surpasses a threshold, the article appears

$$s = \sum_{m=0}^n w_m \cdot c_m$$

n = number of matches (between article and user model)

w = weight of the matching model item

c = confidence of the matching model item

in the newspaper. Third, the model planes are built by running a render-and-layout routine on each of the planes described in the user model.

Fourth, the list of matches between each article and the user model is used to construct a totem pole for each article. Only the strongest matches between the article and the user model appear on the totem pole.

In the current system there are no feedback mechanisms (passive or active), new inferences are not possible. Likewise, until feedback mechanisms are built, PeerGlass can not automatically update the contents of the user modelling data files based on how the reader uses the newspaper. PeerGlass is also lacking sufficient reasoning algorithms and community clustering routines. Lastly, there are no user interface mechanisms in place for users to enter their explicit interests or any other data. All of these missing features will be discussed in subsequent sections. These realities should be kept in mind while reading the rest of this document.

Descriptions of the Currently Implemented Model Planes

Explicit Interests Plane

Only items that users tell the system to include in their newspaper will appear on this plane. Likewise, any changes to this plane must be done by the user. This plane covers the categories of Nature, Topic, and Type. All interests on this plane are given equally high preference. The layout order on the plane is simply alphabetical. See Figure 7.

Explicit Expert Sources Plane

This plane contains a list of publications that the user explicitly trusts. The list of sources is very similar to the lists for Topic, Type, and Nature on the Explicit plane. The difference is that sources have added information associated with them. That is, an explicitly preferred source does not just have a name, it has a scale too. PeerGlass assumes that the human editors of a source have a scale for rating articles. For example, many sources rate a “cover story” more highly than a “buried” article. The scale is probably similar for many newspapers but may vary significantly for magazines and other sources. Without knowing the scale of a source, it is not possible to determine the importance assigned to the article.

Each column on the plane represents the opinion of an expert. The rating scale is vertical. See Figure 8. If the editor of the news source rates an article highly, then the

user's model will also give the article a high score. In the event that the same news is covered in multiple sources, the user's model will try to take the articles written by the explicitly preferred sources. Without this plane, PeerGlass would not be able to recognize high-priority articles on new topics. Moreover, this plane allows users to receive serendipitous articles from their favorite sources.

Implicit Interests Plane

Inferring accurate interests by observing a reader's behavior is very difficult[7][12]. PeerGlass simply records which articles the user spends more time reading. This plane shows the results of the observations. See Figure 9. Items that appear near the top seem to interest the user more than items that are near the bottom. The font grows larger when the system gains more confidence in the inference. Confidence grows when users confirm their preferences by reading (or ignoring) the same item over a period of time. Here are a few examples to help explain the way this plane works:

Example 1. The user spends a lot of time on each one of the many articles presented on the Topic of basketball. The system infers that the user has a very high preference for basketball and has high confidence in this inference. The Topic of basketball appears at the top of the plane in a large font. Any match with "basketball" earns a high score.

Example 2. The user spends a lot of time on each one of the few articles presented on the Topic of cycling. The system infers that the user has a very high preference for cycling but has low confidence in this inference. The Topic of cycling appears at the top of the plane in a small font. Any match with "cycling" earns a medium score.

Example 3. The user seems to ignore each one of the many depressing articles that are presented. The system infers that the user has a very low preference for depressing news and has high confidence in this inference. "Depressing" appears at the bottom of the plane in a large font. Any depressing articles would earn a low score.

Example 4. The user alternates between reading interviews and ignoring them. Since only a few "interview" articles have been presented, there is not much data on which to base an inference. The system suggests that the user has a medium preference for interviews and has a low confidence in this inference. "Interview" appears in the middle of the plane in a small font. Any interview articles would earn a medium score.

To summarize, there is a confidence level associated with each inference made by the user modelling system. If there is a lot of contradiction in the data or there are very few

data points, then the system computes a low confidence level for that inference. This is important because the system draws numerous inferences that are so weak (low confidence) that a PeerGlass user should not be distracted by them.

Implicit Communities Plane:

The system recognizes significant clusters of readers who seem to share common interests. These clusters of readers are implicitly-formed communities (not to be confused with clear-cut, demographic communities based on age, gender, etc.). PeerGlass calculates a distance measure between a user and each implicitly-formed community. On the model plane for this information, there is a spiral with the word “you” in the center. The distance of the community from a user corresponds to the represented distance of the community from the center of the spiral. See Figure 10. The communities that are far from the user appear near the tail end of the spiral. Each community is represented by a three-word grouping of the most prominent common interests of the community. For instance, a community represented by the tags “LA Times”, “Celebrities” and “Scandalous” would include people who tend to read scandalous news about celebrities in the LA Times.

Articles that match the interests of a community are given higher scores based on the proximity of the community to the user. This community mechanism allows for PeerGlass to try out new sorts of articles and expand the user’s model. Likewise, looking at community interests helps to increase the variety of news that a user receives.

Focusing on Words

The design of the totem poles and model planes is centered on the idea that news content can be sufficiently described in discrete units represented by words. That is to say, that all of the features of the articles and the content of the user models can be described by words. For the domain of news, these words fall into the categories of Topic, Type, Nature, and Source.

There is no evidence that representing the content of articles or user models with words is sufficient. An alternative approach is to base the user modelling on examples of articles that the user likes or does not like. Additionally, PeerGlass does not attempt to balance the article selection or layout. A well-balanced newspaper takes into consideration the entire set of suitable articles and tries to select those with enough breadth and depth for an enjoyable news reading experience.

Figure 1: The Sources that strongly match the current article and the user's model appear on this side of the totem pole. In this case, the article about Psion appeared as a featured article in Byte magazine. About five Sources can be neatly listed here.



Figure 2: The Psion article covers the Topics of: palmtop, PDA, technology, and computers. There are other Topics that appear in the article but only those that strongly match the user's model appear here. About eight can be legibly listed.



Figure 3: The Types of news that appear in the Psion article and match the user's model are: Press Release and Review. The angles for this side of the totem pole are the hardest to read. The difficulty is that changing the angles may cut-off more of each word. Still, some improvements are necessary.



Figure 4: Adjectives that describe the Nature of an article appear on this side of the totem pole. The Psion article is Promotional and Consumer-oriented. Both of these descriptions match the user's model. At most, six adjectives can fit well on this side.



Figure 5: This is what the user sees when PeerGlass is launched. The section choices appear down the left side. Which sections are listed is dependent on the user's model and the available news. Currently, the "Tech" section is being displayed. The six totem poles correspond to the six articles shown. There is text on the bottom of the screen explaining the purpose of the totem poles and instructing the user of the available options. The user has four options now:

1. Change sections by clicking on a section button.
2. Display the full text of an article by clicking on the article or its totem pole.
3. Rotate or manipulate a totem pole with the mouse or arrow keys. This will provide insight about why the article was selected.
4. Click the "Examine Profile" button to see the user profile for this section.

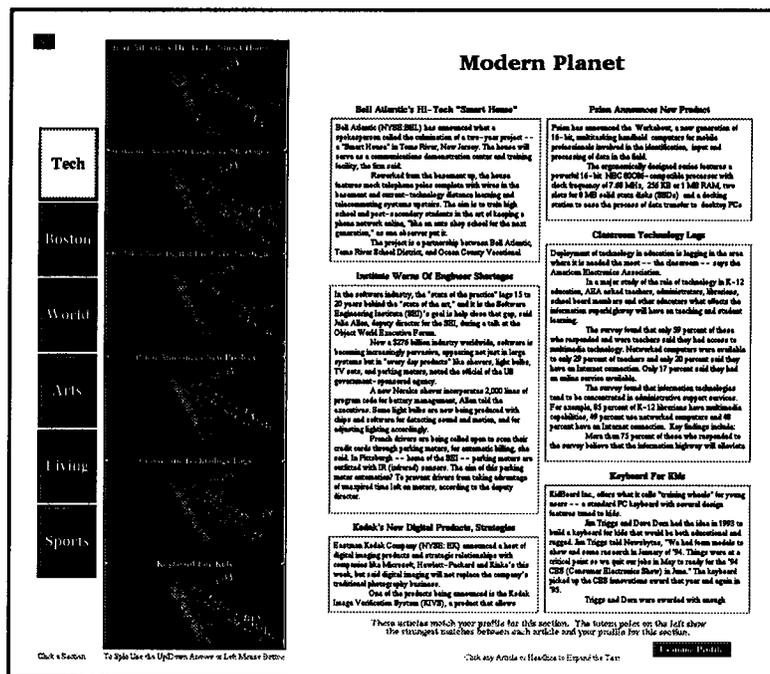


Figure 6: The Rolodex of model planes becomes visible after the “Examine Profile” button is clicked. Above the Rolodex, there is a paragraph explaining the contents of the current model plane. In this example, the current plane is the Implicit observations plane. Below the Rolodex, is text that explains the use of the Rolodex. The totem poles remain visible so that the user is able to compare their content to the model planes. The options available to the user are:

1. Spin the Rolodex with the dial.
2. See the next/previous model plane using the arrow keys.
- 3*. Change sections by clicking on a section button.
- 4*. Display the full text of an article by clicking on its totem pole.
- 5*. Rotate or manipulate a totem pole with the mouse or arrow keys.

*In this case the Rolodex would be replaced by the six articles.

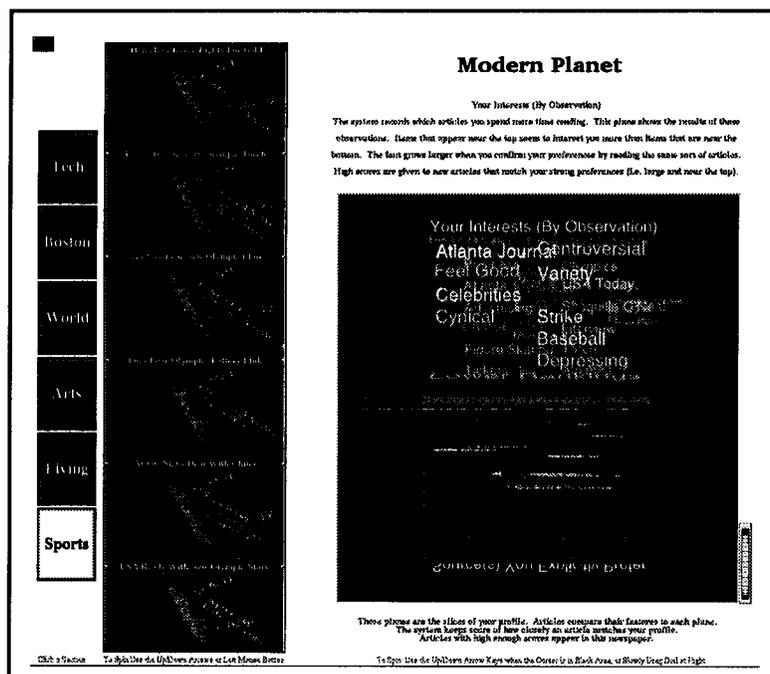


Figure 7: The Explicit interests plane lists the interests that the user has told the system to include. The categories of Type, Topic, and Nature appear on this plane. Since the levels of preference and confidence do not vary with explicit information, the preferences appear in alphabetical order. In response to the user evaluation, there will be a scale of preference in future versions. Also, the design will have to be altered to fit more than a dozen items in each list. Currently, no interface is implemented for the user to explicitly select interests but there are a variety of possible interfaces discussed later in this document.

Your Explicit Interests

The interests you have told to the system are listed on this plane. These items will not change unless you change them explicitly. All of them have equal priority. The system gives positive scores to articles that match items on this list. (Aside: The user interface for entering your interests explicitly is not implemented at this time, so just imagine that it is dialog box, etc.)

Your Explicit Interests

Type

- Expose
- Interview
- Press Release

Topic

- Television
- Technology
- Movies
- Kids
- Celebrities
- Broadway
- Advertising

Nature

- Environment
- Health
- Business
- Government
- Science

Lower ranking

High ranking

Lower ranking

Communities Near You

To Spin: Use the Up/Down Arrow Keys when the Cursor is in Black Area; or Slowly Drag Dial at Right

These planes are the slices of your profile. Articles compare their features to each plane. The system keeps score of how closely an article matches your profile. Articles with high enough scores appear in this newspaper.

Figure 8: The Explicit expert sources plane shows the sources that the user has told the system to include. The vertical scale along each column represents the priority assigned to articles by the expert source. Although the priority scale for most newspapers will be the same, it may vary for magazines and less frequent publications. Without the priority scale, the system cannot determine the importance of an article. If the user does not declare explicit expert sources, it is impossible for the system to incorporate headline news that involves fresh stories (not likely to match the existing set of explicit or implicit interests). Declaring favorite sources also helps to prevent redundancy in article selection.

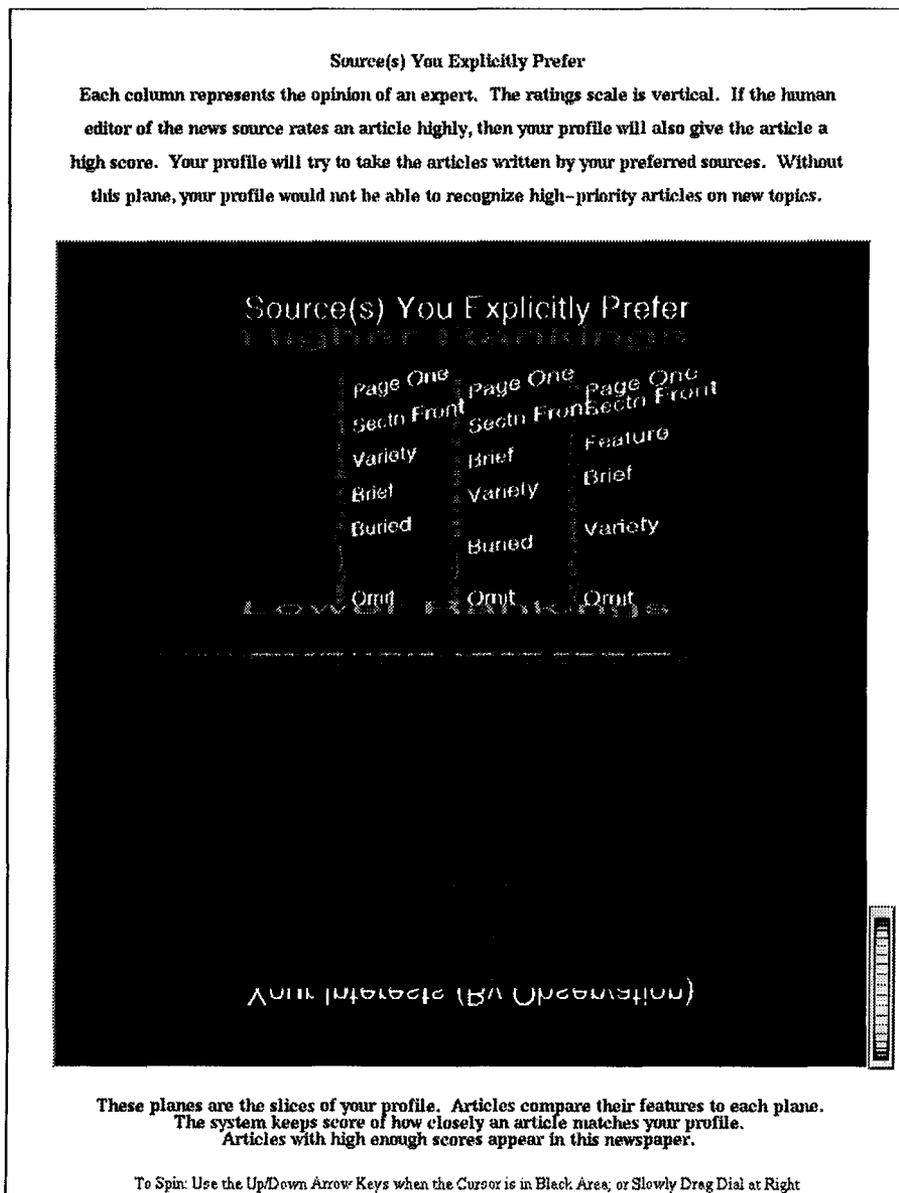
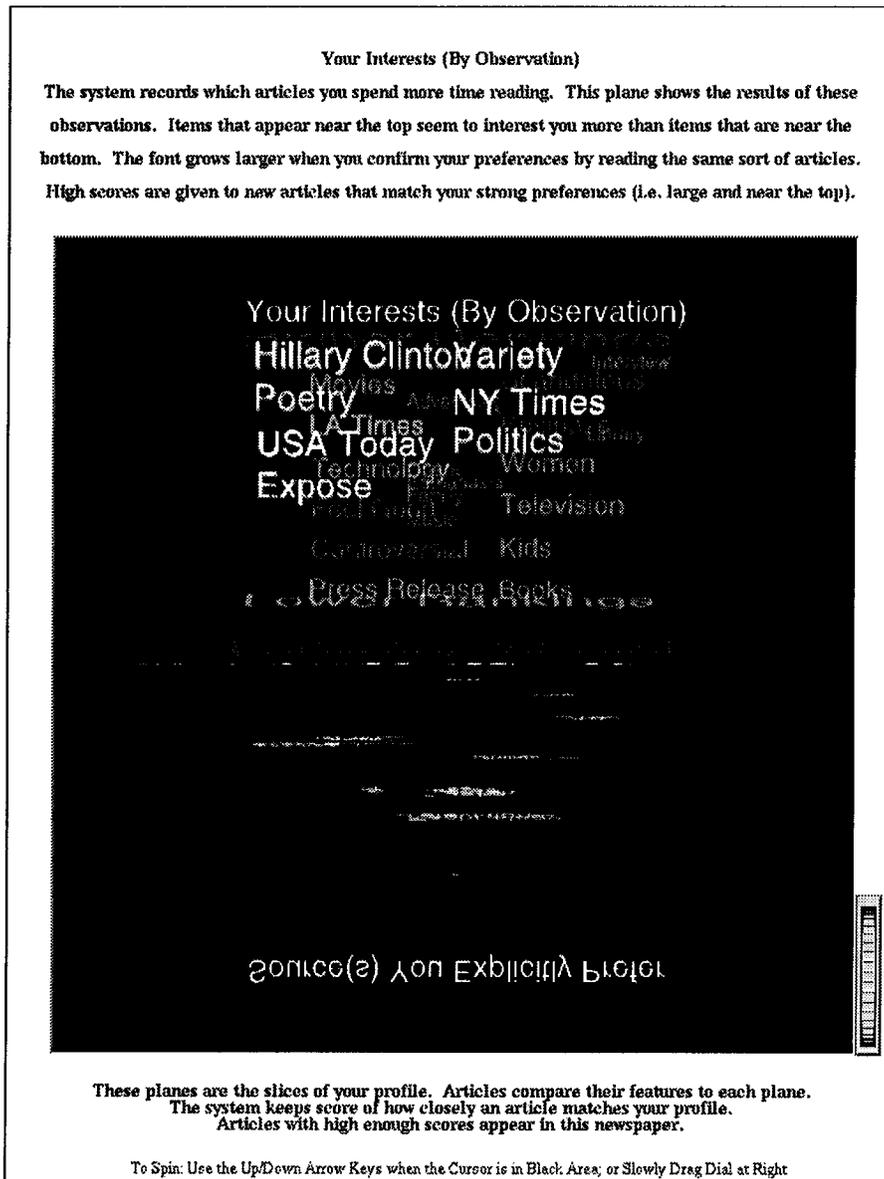


Figure 9: The Implicit interests plane shows the results of inferences the system has made by watching the user's behavior. If the user seems to like a certain Topic/Type/Nature/Source of news, then that item is presented near the top of the plane. As the system confirms these inferences by making more observations, the item appears larger. The user study has shown that this plane is very confusing for many users. New, clearer designs are presented later in this document. The goal of these designs is to present the user with the high confidence inferences and avoid distracting the user with low confidence, non-influential items. Another upcoming change is the removal of the "off-sides" view, currently showing the spiral.



Evolution & Inspirations

Background

The field of user modelling involves collecting data about people and making inferences based on these data points for the purpose of adapting an application to fit the user's needs[6][7]. Independent of whether user modelling is applied to improve the focus of advertisers, optimize the relevance of newspaper articles, or assist viewers in quickly finding an interesting show on cable TV; consumers should have the opportunity to explore their own user models and be aware of what bits are being collected about their actions and interests.

An assumption underlying the creation of the PeerGlass system is that users who understand how their personalized application operates, will be able to improve their interaction with the personalized application. This point is made by Cook & Kay[2] and Bull, Pain, & Brna[1] with their tutoring systems. They argue that students who explore their own user models have the valuable opportunity to reflect on the gaps in their knowledge. It is further argued by Orwant[3] that visualizing user models enhances the trust between the system and the user.

A user modelling system has three main components: a set of data points about the user, a set of inferences about the user, and the algorithms necessary to derive the inferences from the data points. User models are mostly based on observations about the user that the computer can make; this is a very limited representation of the user. For instance, it is possible for a user modelling system to observe that a user is interested in advertisements for heavy sunblock lotion but a system can rarely discover that a user has fair skin or is traveling to the tropics. Consequently, user models are not true reflections of the people they describe but only an approximate representation based on observations and inferences[5]. Users who understand the limitations of the system and who adjust their expectations of personalized news accordingly, may be more satisfied with the application.

Demystifying user modelling is difficult, because most user modelling systems are deeply rooted in statistics and inference mechanisms. The processing of raw data into useful inferences follows a line of reasoning that is often hidden from the user. Previous

attempts to visualize user models[1][2][3][4] have relied on textual lists, two-dimensional trees, and flat graphs. PeerGlass is distinct from other systems because PeerGlass tries to represent a user's model from different perspectives. Each angle is focused on answering reasonable questions that average users may pose about their user model:

“Why did I get this article?”

“What does the system think it knows about my reading preferences?”

Early Versions of PeerGlass

Early versions of PeerGlass have been shown at meetings of the News in the Future consortium (a gathering of MIT Media Lab sponsors). In December 1993, consortium members were invited to provide feedback on six screen designs that were done using BadWindows (a graphics workstation environment created at the MIT Media Lab). Each design demonstrated a different way of visualizing the topics in a sample user model for one section of a personalized newspaper. The designs surrounded a grid-of-keywords approach that optimized the use of screen space while using the vertical position of a keyword to represent its relative importance to the user. The grids included boxes for showing correlated topic data (e.g., ways to specify preferring news about just a couple of football teams not the whole league; or ways to specify interest in a topic with an ambiguous keyword like “Newton” or “Java”). Spatial orientation, color contrast, and transparency techniques were varied across the designs. See Figure 11.

Another system for visualizing user models was built for the May 1994 consortium meeting. This version was an Athena Widgets X-Windows program that provided a user interface to the user modelling system [built by Michelle McDonald] that was integrated with the newspaper server [written by Klee Dienes]. This user interface approached user modelling for newspapers from two perspectives. From one angle, the system provided a grid-of-keywords approach to the newspaper section model of a particular user. See Figure 12. Implemented features for the section model included viewing the models of other users, adding new topics, marking items ‘private’, and providing direct feedback. From a second angle, there was a separate user interface for accessing the user model as it pertained to the currently selected article. See Figure 13. The point here was to allow users to investigate why a particular article was selected for their newspaper. Again, direct feedback, and adding new topics were encouraged. The PeerGlass system in its 1995 version places more emphasis on demystifying the user model and focuses less on user feedback issues.

Changing Focus

The user modelling information shown in the earlier versions of PeerGlass were oversimplified representations of the user model. The next goals for PeerGlass included showing the user (1)the sources of the data, (2)how the system reasons or draws inferences from the data, and (3)the resulting inferences.

Abstractions and Contraptions

The work done in Fall 1994 tried to improve upon the earlier versions by adding more details about how the user modelling system filtered news. The grid-of-keywords approach was expanded from denoting the user's level of preference along the Y-axis to also having the Z-axis express the system's level of confidence in a topic. Transparency effects were used to show depth in a 2D environment. Low confidence items appear faint since they barely influence a user's model.

More information was added to the representation to account for each factor that influenced a user's model. For example, influencing factors might be records of every time a user has read an article on a certain Topic. A second example might be a community influence that prefers a certain Type of news. These kinds of influences (among others) are represented by abstract shapes in Figures 14 to 17 (built in Macromedia Director). The size and color-coding of the shapes identify the type of influence (explicit, implicit, etc.). In Figures 14, 15, and 16 the idea is that the influencing factors in favor of a Topic are shown pushing it upward while the degrading factors weigh it down. Again the Y-axis was intended to be the level of preference. Figure 17 adds a textual explanation behind each influencing factor. Perhaps the user could move to see the side of the representation and then read the text that provides the history of each influencing factor. For these diagrams the level of abstraction is dizzying: shapes, colors, fonts, and positions; all having meaning. Although visually appealing and packed with information, these representations were too complicated to decipher.

Next, a representation based on concrete, physics-based contraptions was planned: Each influence (e.g., every instance of an article being read on a certain Topic) should appear as a little ball. The balls in favor of a Topic would be stacked in one bucket, while a bucket of opposing balls would rest on the other side of a scale. This way the user could see how the scale was tipping as the system recorded the actions of the user. This

contraption idea was abandoned for being too grounded in a multi-variable equation representation and not extensible to represent a variety of algorithms for user modelling.

New Directions: Slices

Trying to show the sources of the data, the reasoning applied to the data, and the resulting inferences all at once seemed to be the failure of the abstractions and contraptions. The next attempt to visualize user models was aimed at representing the sources of the data and the inferences that resulted from the reasoning, but did not try to visualize the actual reasoning.

The boundaries of data collection in the current system are these four sources of input:

- Explicitly declared interests

- Implicitly observed interests

- Externally recommended interests (from editors of news sources)

- External influences from communities similar to the user

Additional data collection mechanisms (e.g., positional sensors and tracking devices) could be appended to this list.

Breaking down the user modelling data into these categories appeared to simplify the representation (as opposed to using hordes of abstract shapes or contraptions). Inspired by Lisa Strausfeld's visualizations of mutual fund data[19], the placement of information for each category onto separate planes [graphically] made sense.

A new version of PeerGlass was built in C++ using the Open Inventor libraries for the SGI. All previous instantiations of PeerGlass used transparency to represent depth, but in a 3D environment it is possible to allow the user to experience depth. 3D also allows users to rotate and examine their user model, this is not only fun but also provides users with a greater sense of control. Earl Rennison's work[17], *Galaxy of News*, where the user flies through a space of newspaper articles (not user models) was an inspiration for PeerGlass. An important navigational difference between the user interfaces is that PeerGlass can be entirely controlled with arrow keys that choose pre-selected points of focus; *Galaxy of News* is a free-flying environment.

There are three major differences between the early designs and the new “slices” model. First, the user model viewer is no longer generalized and abstract but is instead linked into a newspaper system. See Figure 18.

Second, the user modelling information appears as a series of planes in space. See Figure 19. Each plane corresponds to a different aspect of a user’s model. There is a plane for explicitly declared interests of the user, a plane for implicitly observed interests, a plane for community influences, and a plane for each expert source (human editor) who has stated the importance of the article. The intention is that each article is compared to each plane, a score is accumulated, and articles make it into the newspaper by surpassing a threshold. This is explained to the user in text above the model representation. A big problem with this implementation of the model viewer is that zooming into planes is a clumsy, modal process. See Figure 20.

The third addition to the slices design is the totem widget. The totem shows the most salient matches between the user model and the selected articles. The four sides are: Source, Type, Nature, and Topic. See Figure 21. The user can spin the totem to see a different perspective on the articles. This way of placing information on different sides of an object is reminiscent of Yin Yin Wong’s work on indexing a large set of articles [16].

Current Design: Rolodex

Two necessary design changes became apparent while reviewing the troubles of the slices design:

First, the single-totem approach of the slices model showed information about the salient features of a selected article, subsection, or section. This information is most useful as a means of comparison with other selections. Having only one totem made such comparisons difficult. The totem-per-article approach of the current design helps address this problem. Yet, a side-effect of trying to display multiple totems is the number of articles in a newspaper section has been reduced.

Second, the slices method for examining model planes was both modal and clumsy. The intention was for users to be able to focus solely on a single plane. Since the planes were arranged like a loaf of bread in the slices method, it was impossible to examine one slice without choosing it, and pulling it out. That interaction required a modal interface. By arranging the planes on a Rolodex, it is possible to take a look at each plane in turn, just

by rotating the point of view. Another advantage is that no plane seems more important than another, since no order is assumed in the circular layout.

The Next Step

The Rolodex implementation tries to show users the different angles of their user model in an understandable way. A usability evaluation was necessary to verify/invalidate this assumption. Determining which features should be the focus of future work is also important. The push for evaluation and re-design comes from the assumption that users may not find the information in the Rolodex particularly compelling. Subsequent sections of this document address the heuristic evaluation and usability study of PeerGlass.

Figure 11: This is one of the six original designs done in BadWindows. This design only shows topics and makes no effort to distinguish implicit from explicit information. As shown on the left-side scale, the items near the top are high priority while those at the bottom are low priority. The gradient background and use of color allows the high priority items to be emphasized with high contrast, as the low priority items fade into the background. Users can slide items up and down. The X-axis has no meaning. The boxes that contain more than one item are meant to convey a correlated topic. The header appears on the top while the items within the box are organized in a miniature version of the grid (higher is better, etc.).

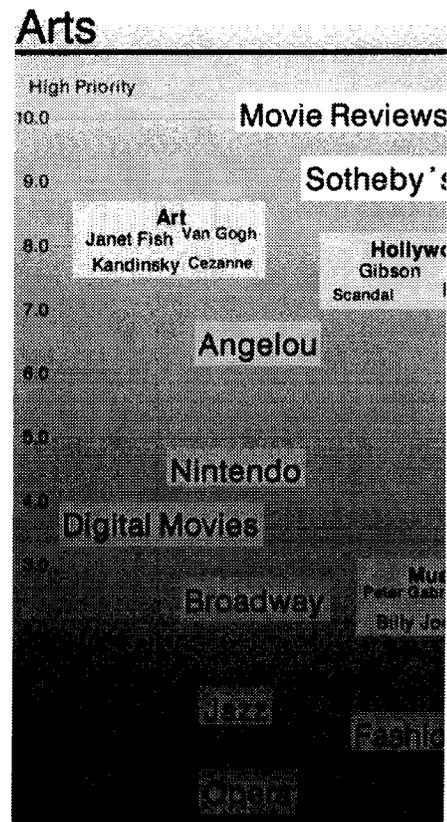


Figure 12: In this earlier 1994 work, the newspaper appeared in one monitor while the user model was displayed on another monitor. When the section of a newspaper was selected, the “section” view of the model was shown. The user can slide any of the Topics up/down to raise/lower its importance. The buttons along the left side show available options (some are grayed-out):

Add Topic from a scrolling list.

Add Cluster... for a correlated set of Topics.

ReFilter News Now to filter the latest articles through the latest model.

Make Public vs. Private, with regard to other people viewing it.

Undo.

Show Filter of... another user or community.

Copy to My Filter something shown with the *Show Filter of...* option.

Join Community will cause the currently selected community to influence the model for this section.

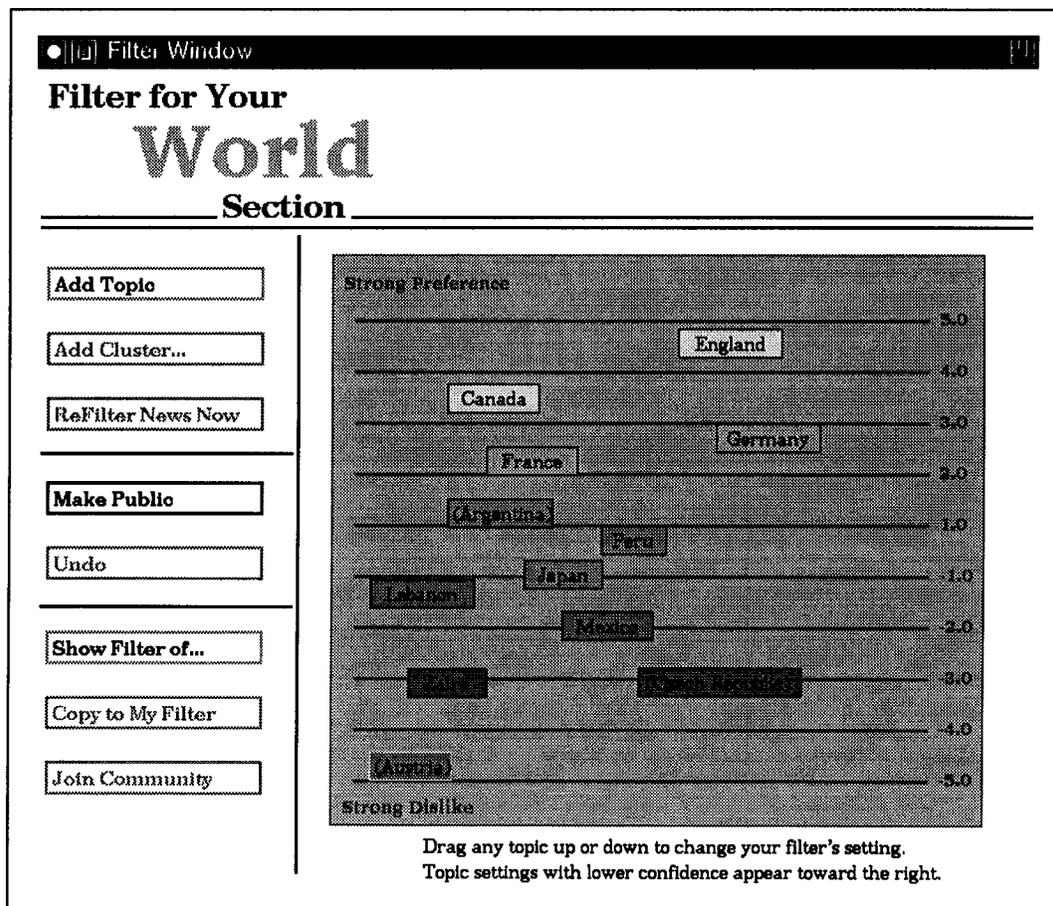


Figure 13: In this earlier 1994 work (described in Figure 12), when an article was selected, this “article” model window would replace the “section” model window. In the upper-right, the top reasons why the article was selected appear. The thumbs on the left are for the user to explicitly provide quick feedback about the article with an option to attach comments. The matches between the article and the user model are shown in the main area. Users can provide feedback by sliding any item up/down. The Topics that appeared in the article, but not the model appear at the bottom. Users can choose to add these items to their model. The buttons along the left side show available options:

Send To... a friend/colleague via e-mail.

Archive... this article for later retrieval.

Keep Me Updated... will alert the user to any updates on this article.

Get Related News for this story.

Get Preceding News for this story.

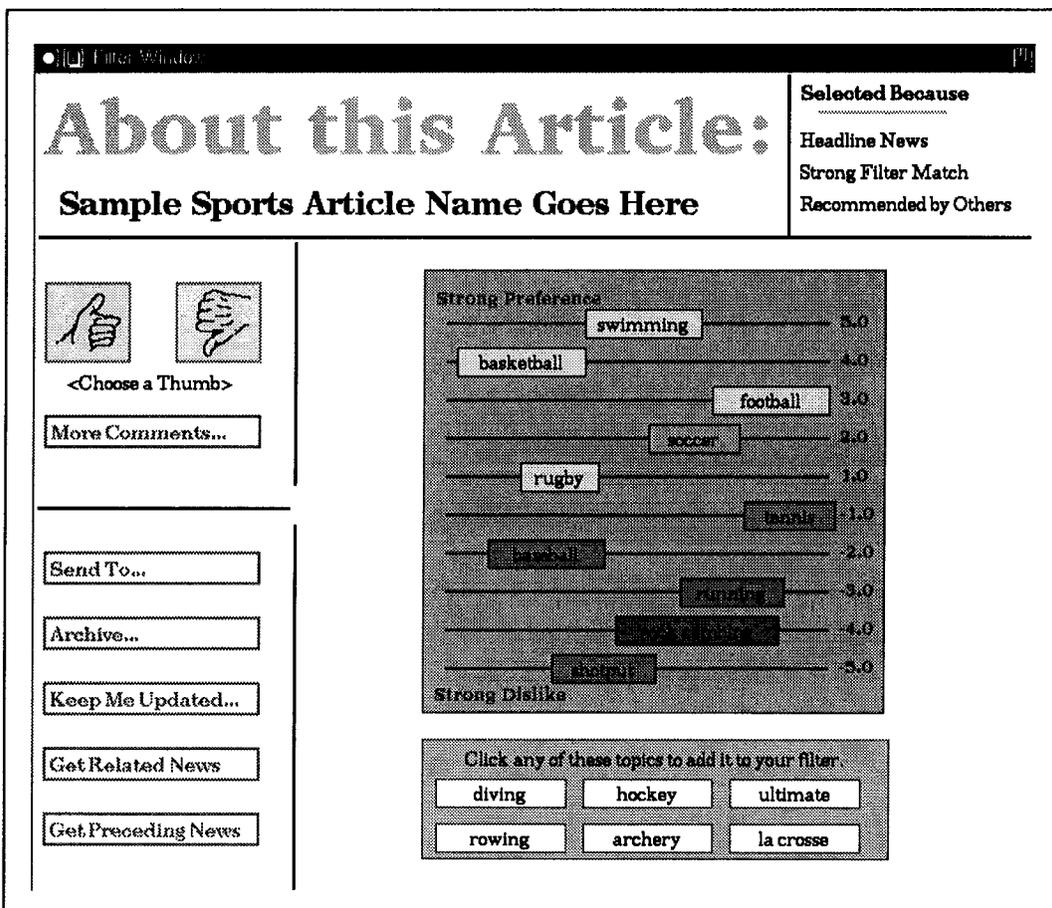
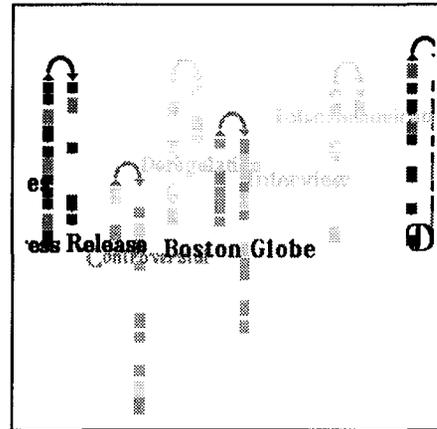


Figure 14: This is one of the abstract attempts to represent user modelling information. The vertical axis is preference, words nearer to the top of the screen are more preferred by the user. Each block represents some event that influenced the system's inference. The left column of blocks represent the positive influences, while the right column contains the negative influences. Notice how the word appears at the vertical point of difference between the two columns.



The colors represent the kind of influence (explicit, implicit, community, etc.). The fonts represent the categories of Topic, Type, Nature, and Source. Transparency shows confidence; the low confidence items fade into the background. This design is just too complicated.

Figure 15: As above, the fonts represent the categories of Topic, Type, Nature, and Source. Transparency shows confidence; the low confidence items fade into the background. In this design, the colors and shapes combine to represent the kind of influence (explicit, implicit, community, etc.). For example, in Figure 14 a green item may be an explicit influence but in this case a green triangle represents an explicit influence. The Y-axis is still preference. The positive influences push up from the bottom, the negative influences push down from above. The words that appear nearer to the top are more strongly preferred. Along with Figure 14, this design is overly complex.

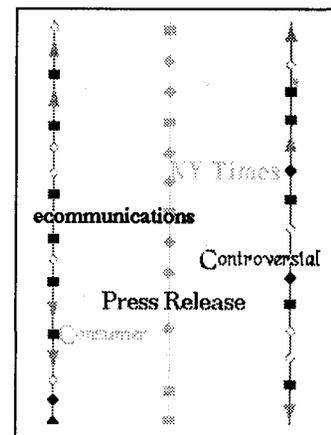
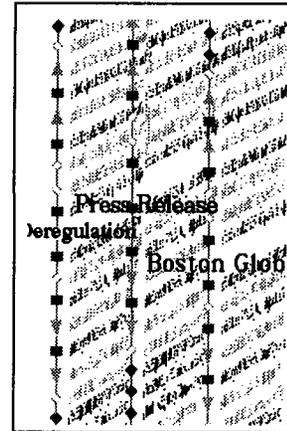


Figure 16: The basic representation is exactly the same as the previous figure. The addition to the design is the angled text that appears behind each “influence shape”. This text is a description of the influencing event. For instance, if a red square represents an implicit influence, then the text behind the red square describes when and how that influence occurred. Perhaps it was an observed instance of the user preferring a certain article. With this representation, users could find out precisely which data is being recorded about their actions and could also observe the impact of their actions on their user model.



To read the angled text, the user would move the viewer to a sideways view.

Although there is a great deal of information in this compact representation, the design is unbearably complicated.

Figure 17: This is another variation on Figure 15. In this case, a plane is drawn in 3D space to connect correlated items. Since no added information is shown about how the correlated items compare to each other, this is not a particularly useful design.

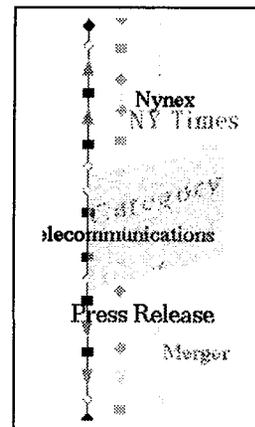


Figure 18: The main window of the “slices” design is shown here. The distinctions between this design and the currently implemented version of PeerGlass are the use of a sidebar and the display of only one totem pole. The sidebar is a very useful browsing device. There are four subsections to each section, and at most six articles in each subsection. By clicking on the sidebar, the user can select which subsection is displayed.

(Please disregard the blank areas below, they do not affect the design.)

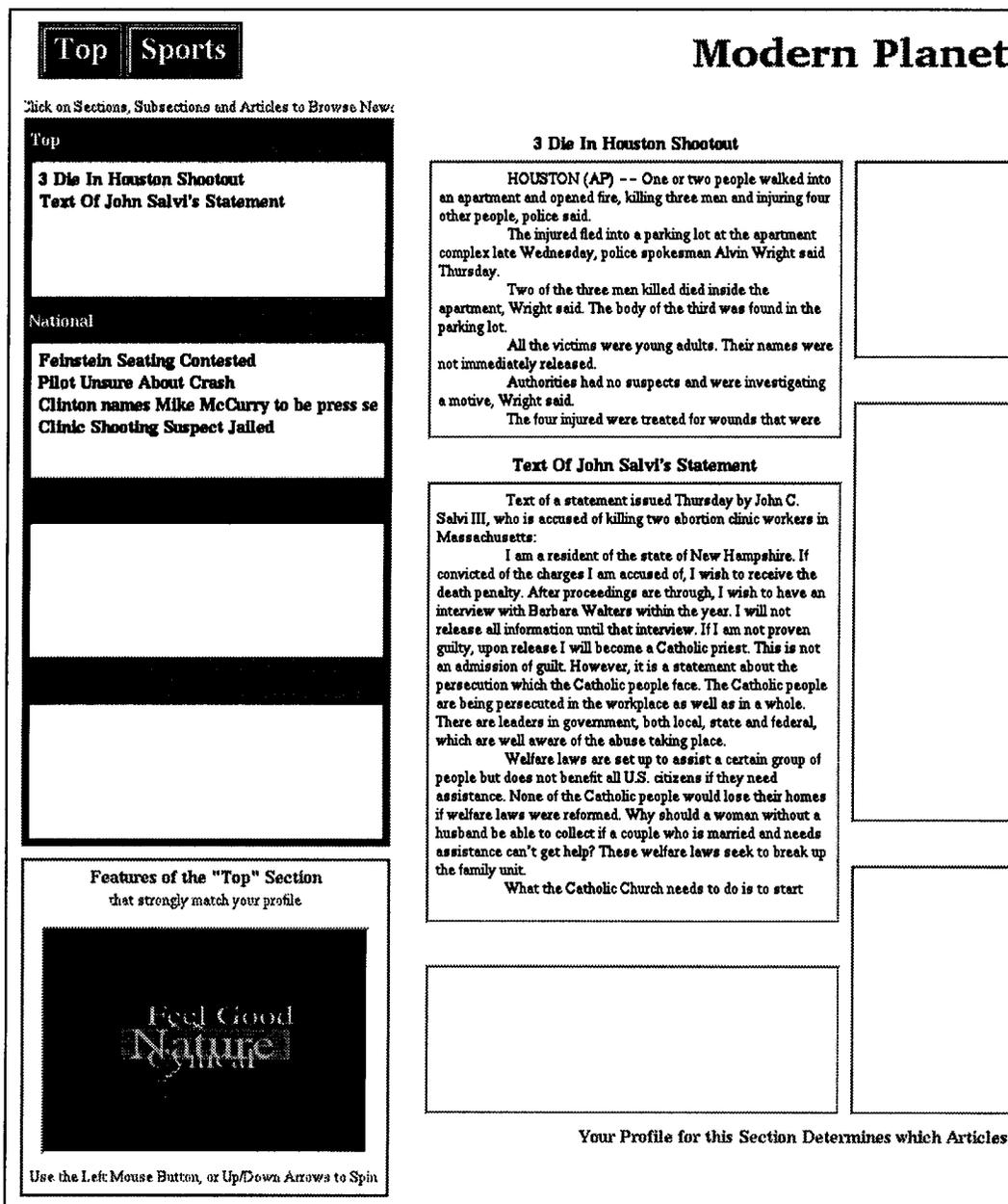


Figure 19: This is a view of the model planes in the “slices” design. The only difference between these model planes and those in the currently implemented Rolodex design is their orientation. A problem with the slices design is that only a few planes can be displayed legibly in the stacked layout. The user can click the “Examine Profile” button to enter a mode where the arrow keys select a plane from the stack. Once the user has selected a plane, pressing the Enter key will launch an animation that brings the selected plane to the front and pushes all others downward. From this point it is assumed that the user can zoom into the current plane for closer viewing. The modal nature of the slices interface is a major drawback.

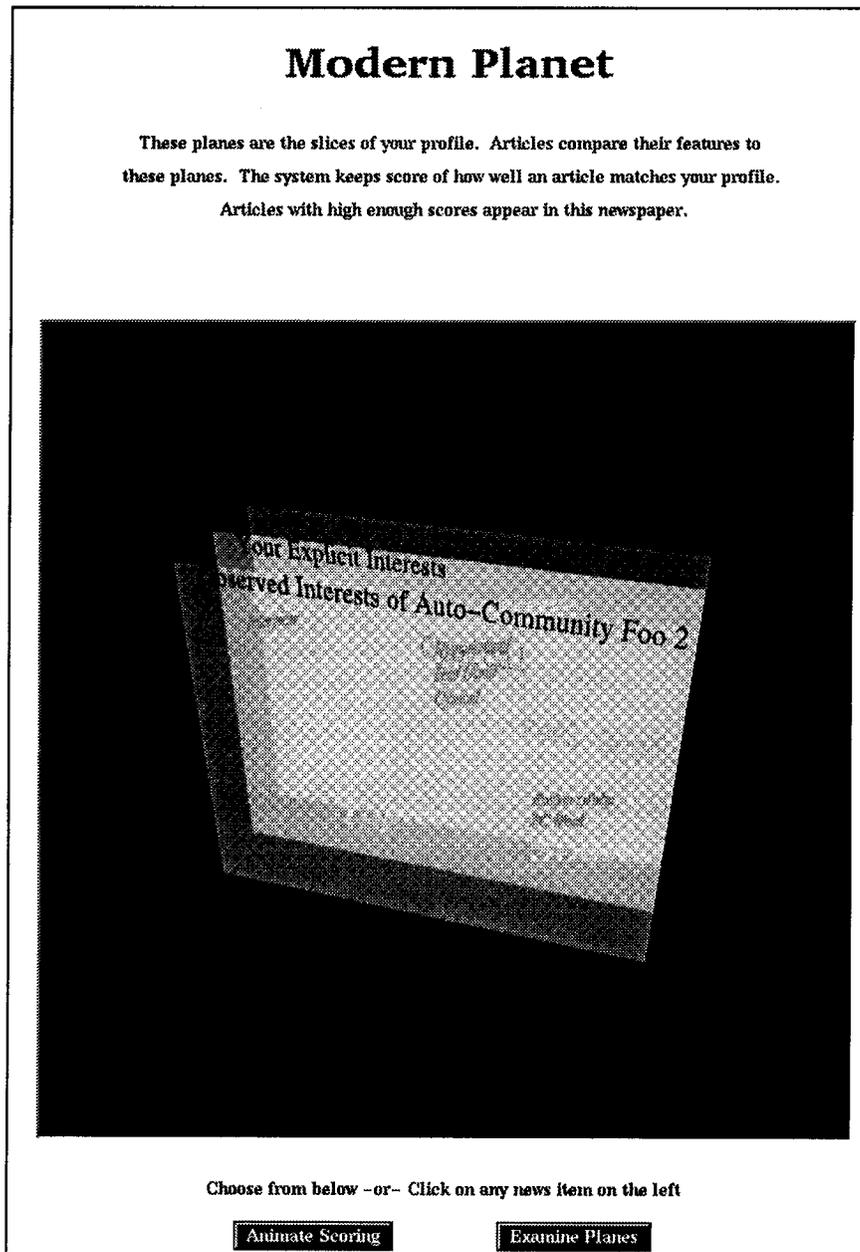


Figure 20: A close-up view of a model plane from the “slices” design (see Fig 19).

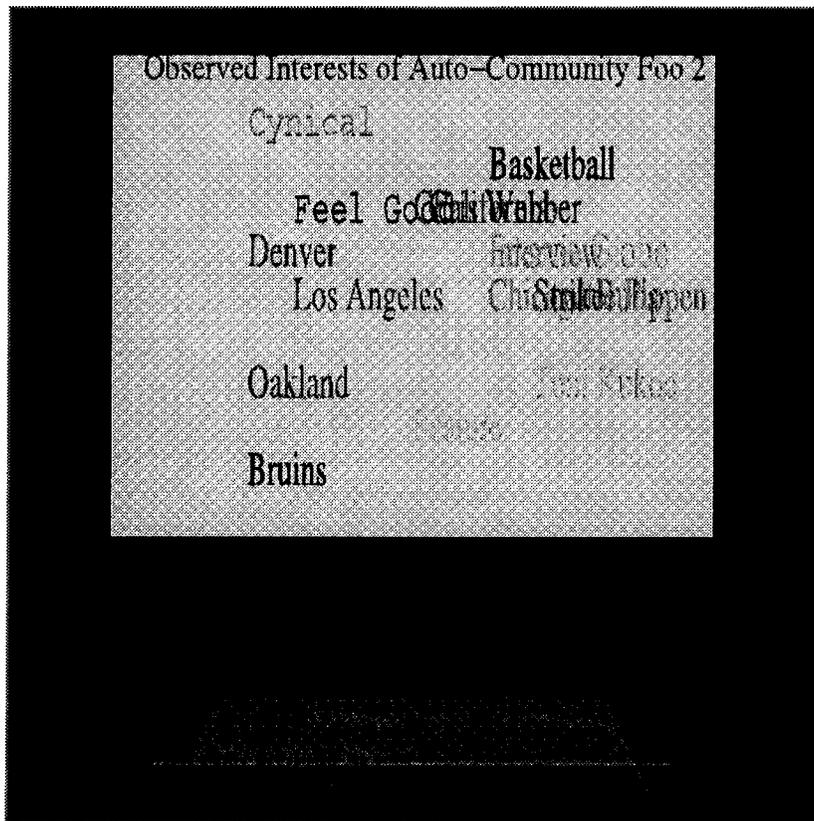
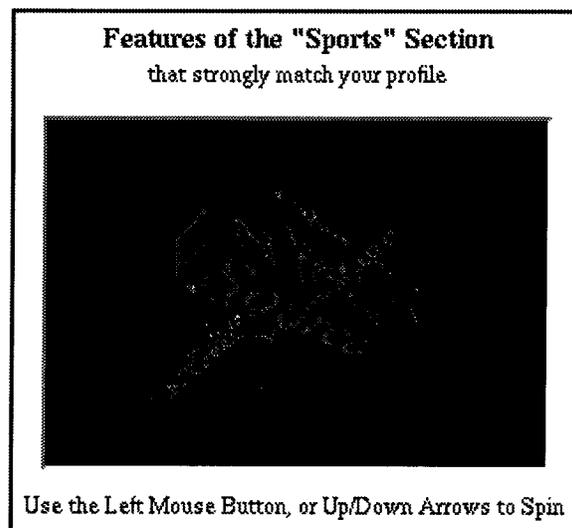


Figure 21: When an article is currently selected, the one totem pole from the “slices” design displays the strongest matches between the article and the user’s model. If a subsection or section is currently selected, then the totem pole displays a summary of the totem poles for the set of articles in the subsection or section, respectively.



Evaluation: The Surveys

Description

Two sets of MIT students evaluated PeerGlass. Each set of students participated in a three-part study. The first part involved exploring and criticizing the system. Next they were asked to consider a set of potential features and rank whether each was important, interesting, or both. Lastly, they considered how to improve the system.

A dozen graduate students were selected based on their skills in visual design, interface agents, epistemology, and/or human-computer interaction; members of the Electronic Publishing group were excluded due to their closeness to the project. To facilitate critical discussion and ease potential frustrations with the system, the students were divided into pairs[22]. Each pair spent an hour filling out a questionnaire while they used PeerGlass. During that time, they were invited to ask questions as long as they had already answered any survey questions that pertained to their frustration. These dozen students later participated in focus group meetings to evaluate potential features for PeerGlass and propose new designs for the model planes.

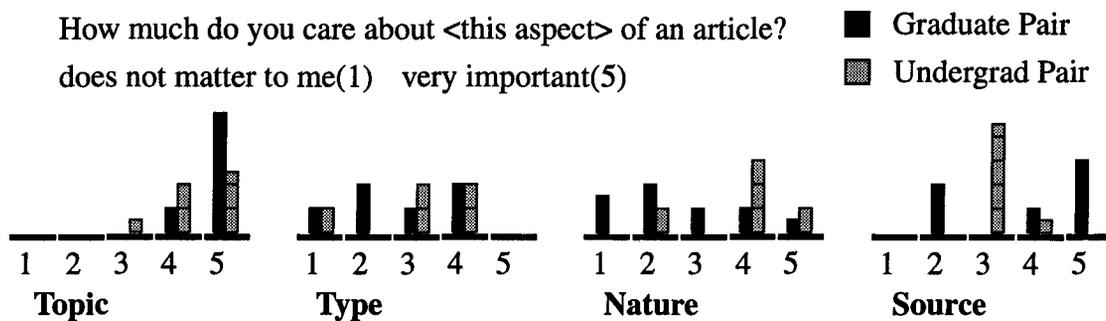
After the graduate students had finished their involvement, the opinions of ten undergraduates were also collected. In this case, none of the undergraduates was from the Electronic Publishing project or any project remotely related to PeerGlass. Just as before, each pair spent an hour filling out the same questionnaire as they explored the system. They were also encouraged to ask questions about any points of confusion or frustration. Subsequently, they were given a set of ten potential features and a set of new model plane designs [a by-product of the graduate students' focus group meetings] to rank. Since they did not have to be in front of PeerGlass for these rankings and did not participate in any focus group meetings, they filled out the forms alone but were allowed to discuss each issue with their partner. (The involvement of undergraduates for this study was approved by the MIT human subjects committee who provided the study identification code: 2239.)

Note the survey questions in this section are not described in the same order they appeared on the survey forms.

Audience

Background information was part of the questionnaire each pair of users filled out while evaluating the system. Four questions inquired about how important each pair regarded the Source, Nature, Topic, and Type of articles. The current implementation of PeerGlass weights each of these factors equally. The data suggests that the weighting of these aspects of an article should be adjusted based on user preference.

The “nature” question involved some confusion as to whether the political slant on an article is considered part of its “nature”. In fact it is regarded as part of an article’s nature but was not clearly shown in the current implementation. Users expressed a high interest in this information but may have answered the “nature” question assuming political slant was not included. A surprising result was the limited interest in the Source of an article. It is strange that some pairs did not place higher priority on this information, considering that the reputations of news sources for accuracy and political slant widely vary.



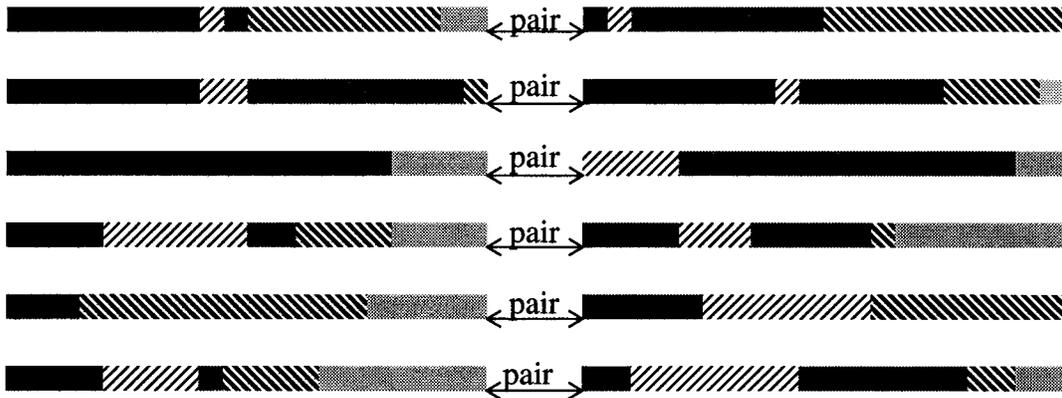
The opinion of each pair is represented by a box. If the partners could not agree on an answer, each of their opinions is represented by a separate box, half as tall as the norm. Also, if a non-integer answer was reported then a half-height box would be used to distribute the difference. For example, an answer of 2.5 would be represented as a half-height box on 2 and a half-height box on 3.

Additionally, information about the news reading habits of each evaluator was collected. Individual students were asked how they receive news and about how much time they spend consuming news on a weekly basis. The data is shown on the next page. No correlations between students’ news reading habits and their opinion of PeerGlass have been found. In future evaluations, particularly those that do not involve MIT students, this background information may be more useful and interesting.

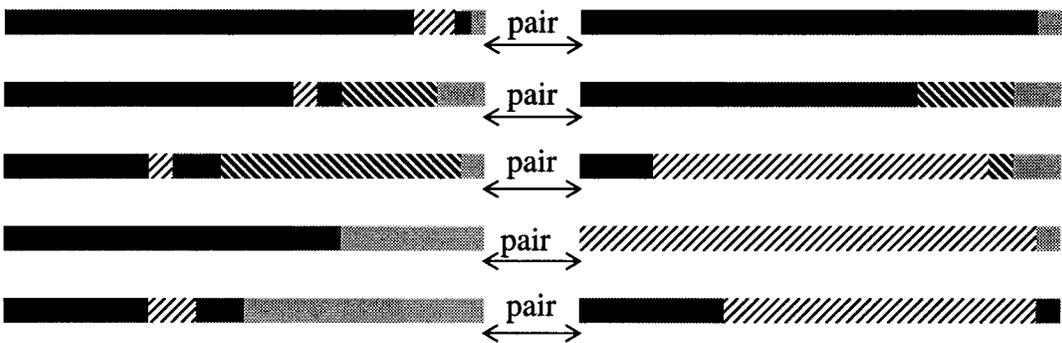
Where do you get your news? (sum of percentages should equal 100)

Focus Group 1

Focus Group 2



Undergraduate Evaluators

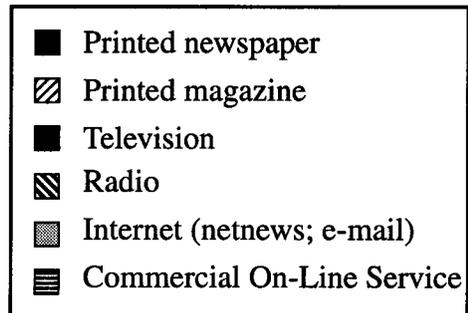


Average estimated number of hours devoted to news (of any kind) each week

FG1 5.2 FG2 6.5 Undergrads 3.8

Average estimated number of hours spent reading printed newspapers each week

FG1 2.3 FG2 1.5 Undergrads 1.5



Some data was rounded to the nearest 5%

Notes about the data: One student desired a category for receiving news by word-of-mouth. One student received 20% of his news from the system built for his thesis but was able to put spread this 20% across radio and Internet sources. One student commented that she mostly receives news while doing something else (e.g., radio while she is getting ready in the morning, and newspaper on the subway).

To measure how the news habits of PeerGlass evaluators compare to the general public, an additional set of background questions will be asked of the twenty-two people who have already evaluated PeerGlass, along with any new evaluators. These questions will match the surveys used in the May 1994 Times Mirror study on The Role of Technology in American Life[15]. This comparison will appear in future publications on PeerGlass.

Totem Poles

An unexpected use of the totem poles became apparent as the graduate student pairs evaluated PeerGlass. Some students were using totem poles as navigation devices. By glancing at the six totem poles for a section, they tried to determine which article they wanted to read. Totem poles were designed to provide a quick answer to the inquiry, “Why was this article chosen for me?” by showing the list of top matches between the article and the user’s model. After the user study it became apparent that the totem poles do not provide sufficient information for either function.

For the purpose of navigation the totem poles would have to convey a specific list of features that apply to the article; not just those that match the user model. Also, the totem poles would have to be more amenable to being manipulated as a comparative device. A simple solution is to have the set of totem poles for each article spin in unison at the press of a key so that the reader can compare all of the Topics across articles, all of the Sources across articles, etc.

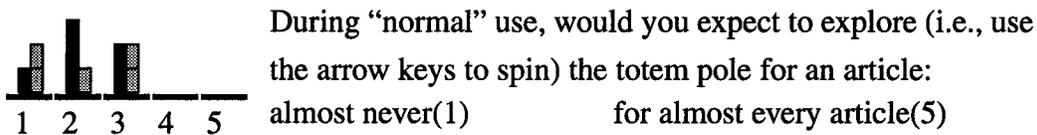
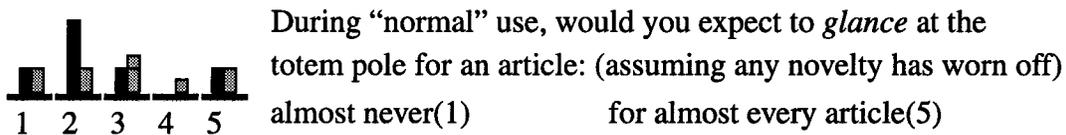
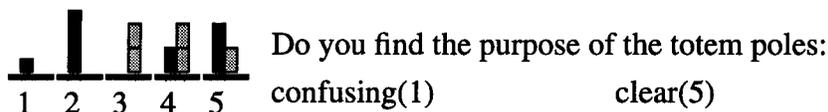
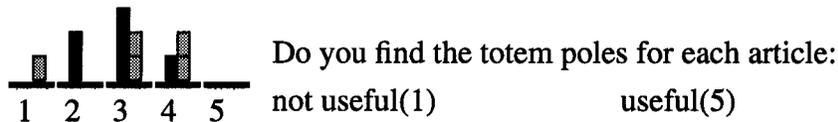
In an older version of PeerGlass there was a single totem pole and a sidebar with article headlines and short descriptions. A few undergraduate student evaluators have suggested a return to this approach as a way to enhance the article browser. Reverting to this design would also eliminate the use of totem poles as a navigational device since only one totem pole would be visible at any time. The downside of this approach is that it would impede the between-article comparison of why articles were selected.

As an inquiry device, many graduate student evaluators pointed out they were only interested in investigating their user model in the event they received unexpected, undesirable news. For the purpose of pinpointing a problem in the user model, the totem pole is insufficient. It does not provide enough information about the filtering process to determine the cause of the unexpected result.

On the other hand, *none* of the undergraduates (each of whom was debriefed but not to the same extent as the focus groups) suggested they thought the totem poles were

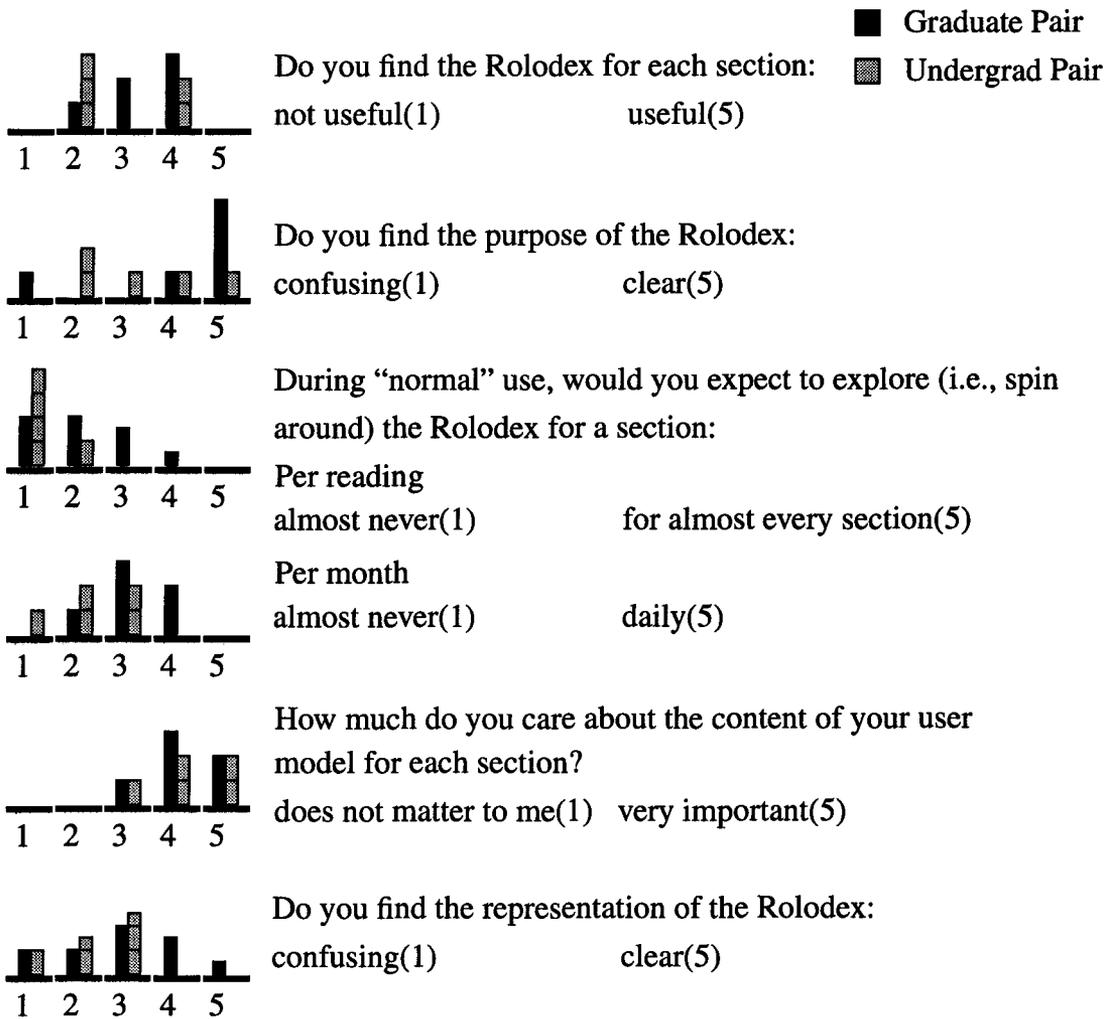
navigation devices. Furthermore, they seemed to be satisfied with the quantity of information conveyed. Their concerns had to do with the lack of a navigation device (such as a sidebar) and the large quantity of screen real estate devoted to totem poles.

The differing concerns of the undergraduate and graduate student evaluators are reported below. Both groups seemed to agree that totem poles are fun widgets that deserve a place in PeerGlass but the legibility of the angled text should be improved.



Rolodex

The idea of representing the user model across multiple planes made sense to most users. Nevertheless, the interface to the Rolodex and its presentation drew criticism. To spin the Rolodex there was a dial widget provided. Users had little difficulty figuring this out but complained about how spinning the Rolodex was often jumpy (due to slow hardware). Conversely, the arrow-key support for jumping between planes was fast but the Rolodex representation was lost since there was no transition or rotation required between planes. Additionally, users complained that while one plane was shown front-and-center, it was possible to see the "off-sides" of planes that were not the point of interest. (This is best shown in Figures 7 to 10.) An improved interface for browsing and zooming the planes was suggested during a focus group meeting and appears in the *Implementation Details* section of this document.

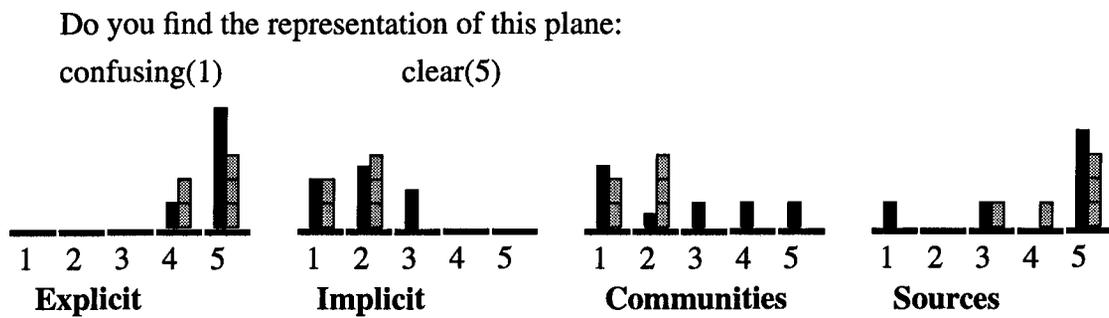
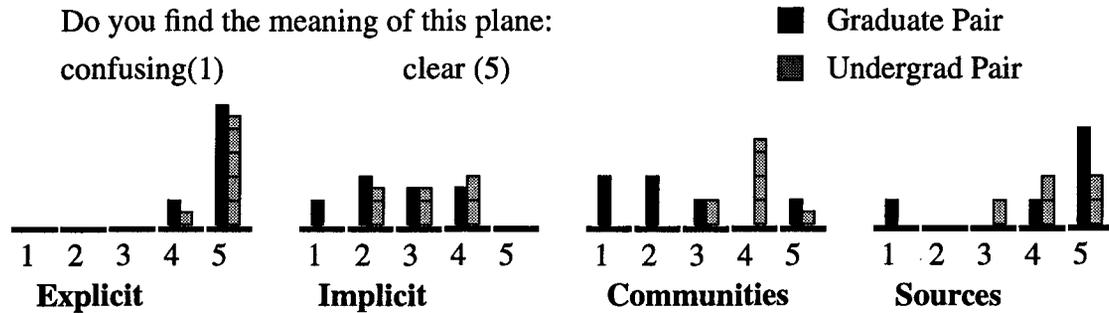


Some evaluators failed to realize that there was a different Rolodex of model planes for each section of the newspaper. Labelling the Rolodex for each section is a quick fix. Another issue was the serial nature of the Rolodex. One user suggested scrapping the Rolodex idea altogether and providing direct access to any plane via a menu to reduce any frustration resulting from having to spin before finding the desired model plane.

Some users wanted to see the articles and the Rolodex simultaneously, this could be accomplished by showing the Rolodex in a separate window. Although this was done during an earlier version of PeerGlass that required two monitors; it is possible to keep both windows on the same monitor, it just requires more window-shuffling on the part of the user. Since the data shows that users do not expect to look at their Rolodex very often it makes sense to provide access to as much information as possible during the times that the users do want to explore their user model, therefore having a separate window is a good idea.

Model Planes

A set of four questions was asked about each of the model planes. There is a correlation between the results; the data is grouped together to show trends.



The scores for the question about the *meaning* of the planes are better than the scores for the *representation* of the planes because there is a short paragraph of explanatory text that appears above each plane (shown in Figures 7 to 10). This text explains the function of the plane and tries to explain the representation too but is apparently insufficient. Not surprisingly, the complexity of representation directly correlates to the survey data. Both the Implicit and Communities planes attempt to represent more than the simple concept of preference, and both of those planes received medium and low scores.

The Explicit plane received high scores because it is very simplistic, just a list of the user’s explicit interests. Users found the color coding (for Topic, Type, and Nature) most effective on the Explicit plane. Likewise, the Source plane was rewarded for its simplicity but it was hard for some users to interpret the editorial ratings scale shown with each source. Most users understood that a “cover story” should be more important than a “buried” article but were confused about why it was important to know if an article, was “omitted”. The intention was to find out how a piece of news rated for the users preferred sources and perhaps an article that was omitted should not appear in their personalized newspaper.

The reasons why the Implicit plane was so confusing were later pronounced in the focus group meetings. Two axes were represented in the design. One, the axis of preference, simply corresponded to the Y-axis. More preferred items appeared at the top, less preferred items appeared near the bottom. Two, the axis of confidence was represented by a Z-effect. That is, the items drawn from high-confidence conclusions appear bigger and brighter while those that pertain to less-confident conclusions are small and dim. There were few qualms with the first axis but combining the two axes created a mixed-metaphor. It makes sense that “more is up” and “big is better” but most users were confused about how it was possible for there to be a big item at the bottom (i.e. low preference, high confidence). Moreover, all users noticed there was a correlation between size and brightness but many could not figure out what it was representing since the concept of confidence is not common outside of the user modelling field.

Other comments about the Implicit plane included confusion about the color coding of the categories: Topic, Type, Nature, and Source. Users have suggested including a legend for the color coding but if the implicit information is sorted by category (similarly to the Explicit plane) then a legend should not be necessary.

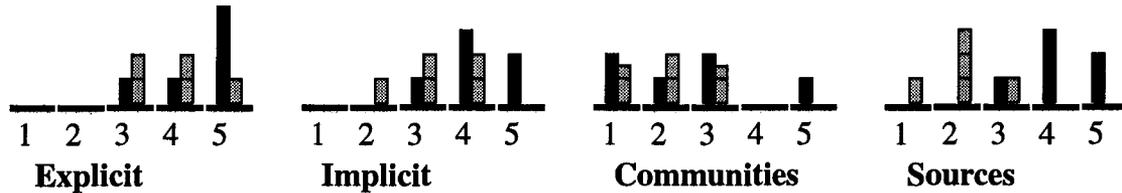
The Implicit Communities plane proved almost as challenging to many people as the general Implicit plane. Part of the problem was the illegibility of the text due to some overlapping clusters of communities. Representing communities as clusters also was confusing but most users suggested they would have a better time understanding the concept if the representation were clearer. The use of the spiral was too complex for the one dimension it was trying to communicate. The idea was communities that are closer to the user would appear closer to the “you” at the heart of the spiral. While communities that were farther away from “you” were farther away from the user’s interests. Some confused users tried to assign meaning to the X and Y axes of the spiral.

New designs for the Implicit plane and the Implicit Communities plane are discussed in the *Evaluation: New Designs* section of this document.

The next pair of questions addresses whether the information on each plane is important to the user.

How much do you care about the content of this plane for each section?

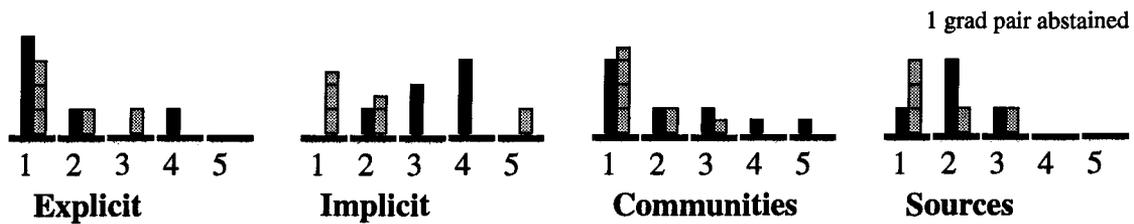
does not matter to me(1) very important(5)



During “normal” use, would you expect to go to look at this plane for a section:

almost never(1)

for almost every section(5)



The Explicit and Implicit planes seem to matter most to users but they have very little interest in looking at their Explicit plane probably because it would not change without their direct interaction. In the current system all planes are weighted equally for the filter scoring but it is clear that the user should be able to customize these weights.

Users have supported the addition of more detailed information on the model planes. First, users are not satisfied by the article rating scale that appears on the Source plane. Perhaps a description of the common Topics, Type of articles, and Nature (especially political slant) of the Source is useful and interesting. Clearly representing this information is a challenge. Perhaps the Sources that the user explicitly prefers should appear as a fourth column on the Explicit plane while a separate plane for each preferred Source provides a full description of that Source.

For the Implicit plane it would be interesting to show the correlations between items. Early attempts to show correlations appear in Figures 11 & 17. A suggestion was made to have a feature where clicking on one word would illuminate it and show the other items that relate to it, while temporarily dimming everything else. Another requested addition to the Implicit plane is a way to show the user the reasoning that lead to the system’s conclusion. Unfortunately, no agreement for how to cleanly represent this information occurred during the focus group meetings.

Evaluation: Future Issues

After the survey-based evaluations were concluded, the graduate students participated in focus group meetings to discuss future features of the PeerGlass environment. There were six people in each focus group; the groups included one member of the six pairs of evaluators. Some features that needed to be implemented were obvious while others required discussion. The focus groups rated ten potential features on two scales: one scale describing how “important or useful” a feature is with regard to the operation of PeerGlass; the other scale pertained to how “interesting or cool” a feature would be to see a few times. Not knowing the outcome of the focus group meetings, the ten undergraduate evaluators each ranked the same set of potential features on the two scales.

Necessary Additions

Before PeerGlass can become a fully functional system the following features must be added (in this order):

1. PeerGlass does not receive news from a live wirefeed because it requires news to be tagged in a particular format. Building a parser to put wire articles into the format would be easy. The user modelling would not be interesting, however, unless the parser could identify the Topic, Type and Nature of the news. The latter items are particularly challenging. Another piece of necessary information would be the list of common Sources (e.g., newspapers and magazines) that have picked up the story and the priority they have assigned to the article (including whether the article is a “variety” piece).
2. Feedback mechanisms have to be added to PeerGlass before user modelling can occur. Active feedback in the form of a thumbs-up/down pushbutton for each article is an easy step. Monitoring whether users choose to view a section or open an article to see its full text are simple ways to gather passive feedback. More sophisticated feedback mechanisms are needed to gather better information (e.g., finding out which features of a particular article interested a user).
3. Inference and reasoning algorithms also have to be added to PeerGlass. Making inferences about an article’s importance based on the explicit, collaborative feedback of multiple readers is a way to start. Building communities implicitly using clustering

algorithms is another direction PeerGlass must pursue. Finally, reasoning algorithms that balance explicit, implicit, community, and expert data are a requirement of any personalized newspaper that grows with the user.

4. User interface necessities include a method for the user to mark explicit interests (probably from a list) and a way for users to assign weighting functions to the various model planes. Consequently, users would be able to set the Explicit plane to always be more influential than the Implicit plane.

5. Necessary newspaper functionality includes the ability to query for certain articles, archive them in a scrapbook, or forward articles to other people. Other less-pressing features include the ability to display articles with opposing editorial viewpoints side-by-side, and the ability to have the system mark updates on certain news stories.

Rating Potential Features

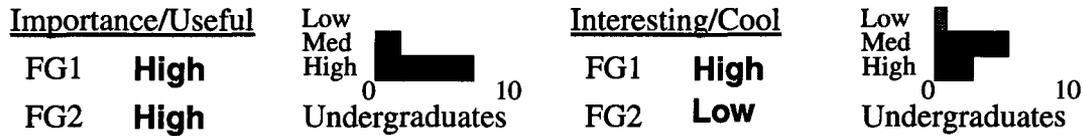
The following ten potential features for PeerGlass were described to each focus group of graduate students. It was assumed that each feature would not be difficult to implement and the result would be easy to use. The students were asked to compare the set of features and provide a High, Medium, or Low rating for each feature. The features were first rated in the important/useful category based on whether the feature helps users understand their user model and interact with their newspaper. Next, the focus groups rated how interesting/cool a feature would be to see or use a few times. A noticeable ceiling effect occurred during the meeting of the second focus group and they were asked to find a couple of items to move from the Medium category to the Low category on the interesting/cool scale so that the Low category would not be empty.

Each of the undergraduate evaluators was asked to rate these ten features on the same two scales. They were encouraged to distribute their ratings across categories and were specifically told not to mark everything Medium. Each undergraduate submitted a separate set of ratings but they were allowed to discuss the issues before reaching their own conclusion.

Confirmations and Contradictions:

Each model plane holds onto rankings of user preferences. In some cases, the ranking of the same item may be consistent across planes; in other cases there are contradictions. For instance, a user has explicitly stated a preference

for news on Bosnia, but does not seem to read any of the articles presented on Bosnia (therefore, the implicit rating of Bosnia is very low). Imagine there is a way to show this contradiction or confirmation to the user.

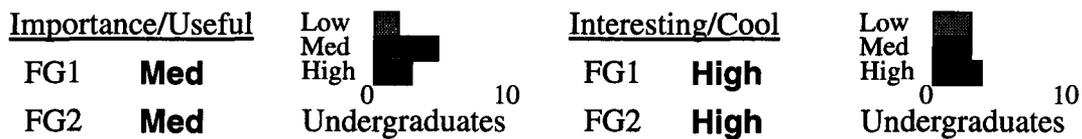


Focus Group #1 felt that exploring one’s user model was most critical when the user is not getting the news that the user expected. Knowing where contradictions are occurring would help diagnose problems. Focus Group #2 thought that finding the contradictions in the system would be useful information for learning about themselves (how their actions may differ from their stated interests).

Both groups thought that discovering contradictions between one’s explicit and implicit preferences may lead to interesting discoveries about themselves. The groups differed on how “interesting” they considered this knowledge.

Looking at Other People’s User Models:

Assume users could see someone else’s, or an entire community’s user model super-imposed with their own. Likewise, everyone has the option to mark parts of their own user model ‘public’ or ‘private’ so that only the public items are visible to voyeurs.



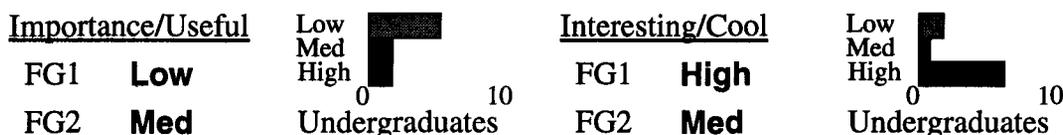
Focus Group #1 felt that this would be a good way to get a sense of who was similar to them, and also to find out other items that were not in their own user model but could be added. Focus Group #2 thought it is useful for finding out information about other people and learning about what other people’s models contain so that they could get ideas about information to add to their own models. They regarded this as an important way to add a new spin to their own model. They were particularly interested in viewing the models of communities to get a better sense of the focus of the community.

Both groups thought it would be fun to see other people’s models for the sake of curiosity. Focus Group #1 felt it would be cool to see how different people share the same interests in a community-building way[14]. Similarly, Focus Group #2 wanted to

be able to see what communities are interested in to find out about local trends. Additionally, this group wanted to see the user models of celebrities to find out their interests.

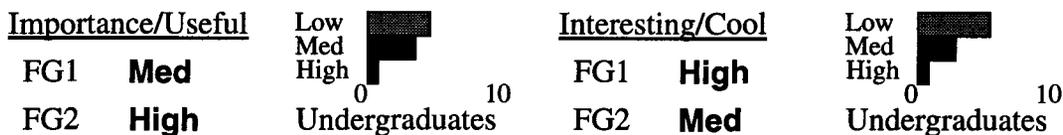
An Animation of Scoring:

During the filtering process, the features of each article are compared to the rankings on each plane. When articles match the items on the plane, the articles accumulate points. Articles that surpass a threshold for points make it into the personalized newspaper. What if there were an option to watch an article get filtered? This would be an animation where the article comes in off the wire, it is compared to each plane, the hits on the first plane light up, a score is accumulated, (this is repeated for each subsequent plane), then the score is compared to a threshold, and either the article makes it into the newspaper or into a discard pile. Perhaps the animation would exist on a larger scale so that someone could look at a whole swarm of articles getting filtered. Such an animation would aim at providing an increased understanding of the filtering process.



Showing Matches Between an Article and the Model Planes:

The totem poles show the strongest matches between the article and the user model. Imagine there were some way to look at the model planes and see all of the matches between the article and the user model. Perhaps the matching items on each plane would be red or noted in some distinctive way. The main difference between this feature and the previous feature, is that this one is not animated.



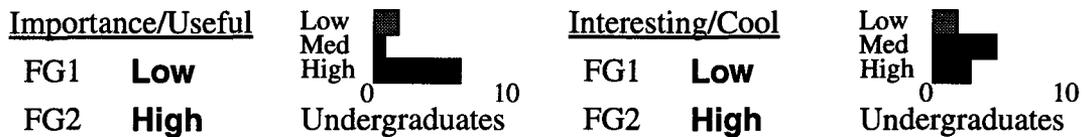
Both groups saw the need to ask the system, “Why did you give me this article?” and thought that the information shown on the totem pole would be insufficient for diagnosing how an undesirable article had slipped through the filter. By seeing all of the matches, it might be easier to understand the reasons why an article made it into the paper. They did not predict the need to investigate every article, just the ones that seemed confusing about why they appeared in the newspaper.

Focus Group #2 thought the animation would not have provided enough control for users who are investigating their user model. The fact that the user can control the speed at which they explore the model planes in the non-animated version earned a positive response. An added suggestion included the ability to take old news and run it through an animation for a recently modified model to test out the new model and see if it produced the expected results.

Both groups thought that the animation would be a fun and useful way to learn about how the filtering is done but also predicted the animation would get dull after the first few viewings.

Creating an Interpolated/Merged User Model:

Imagine some way to merge/interpolate the models of the other people. There are two reasons for doing this: First, merging their models would form a community out of any group of people. If only one printed newspaper were being delivered to a household, a merged user model of the residents would be helpful. Two, using a merged user model as the basis for a new user model may be helpful. For instance, new users could start off with a sports section model that is a combination of their sibling's sports section and their officemate's. (Note this assumes that users could interpolate between other people's models; not merge their own model with someone else's.).

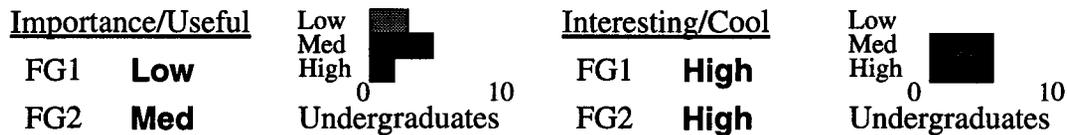


Focus Group #2 thought this would be very useful for building communities while Focus Group #1 found the reasons for merging models to be non-essential for common use.

Focus Group #1 thought that merging models would not be interesting past the first few times if at all. On the other hand, Focus Group #2 assumed that the merging could involve some type of sophisticated morphing and would be fun to watch. A fun example they proposed was merging Rush Limbaugh and Bill Clinton to find their common interests.

Visualizing How the Model Has Changed Over Time:

Both the explicit and implicit preferences of people change over time. Imagine some way to visualize these changes. Perhaps the older preferences would appear increasingly faint in the background and there would be a history trail leading up to the foreground (the current preferences). Perhaps the models would be time-sliced onto a set of planes.

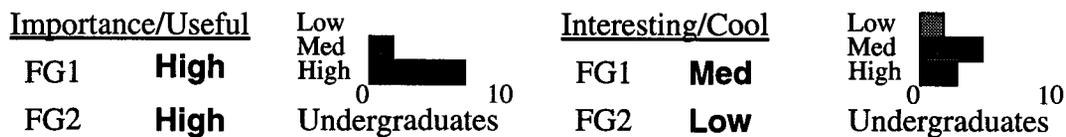


Focus Group #1 thought this feature did not have much value with regard to ensuring that the current paper was an accurate representation of their model. Likewise, Focus Group #2 thought it would provide some insight into how a person's interests have changed over time but would not be a necessary component for understanding how the newspaper is personalized. They felt this feature did not deserve a low rating because it is one of the aspects that distinguish an on-line personalized newspaper from a traditional paper.

Focus Group #2 thought it would be very interesting to see how someone has changed over time. Focus Group #1 agreed but were also interested in reflecting on how the focus of news and the world has changed over time.

Seeing the Articles that were Excluded:

The current newspaper does not give the user any clue about the news that is excluded. What if there were some way to sift through those articles that were excluded?



Focus Group #1 thought it would be an essential option that would help them gain trust in the filtering over time. Similarly, Focus Group #2 suggested this feature helps provide information about how successfully their model is working. An added feature would be a feedback option for the user to mark an article from the discard pile and tell the system

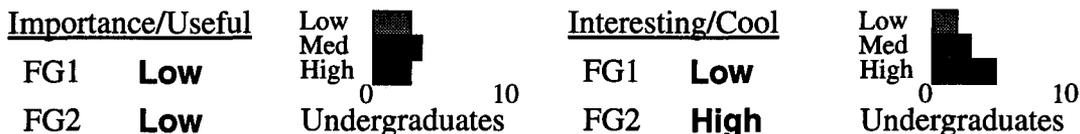
that it belonged in the included pile. This “programming by demonstration / learning from example” work was part of [4]. Relevance feedback was also explored by [9].

Focus Group #1 felt this feature may reveal interesting information about the filtering but neither group thought seeing the excluded articles would be particularly captivating.

During their discussion someone in Focus Group #2 mentioned an additional feature overheard in a talk by Negroponte: a slider for serendipity vs. specificity. The focus was a serendipity factor that could be cranked on the weekends when the reader may be more interested in a variety of news. Focus group #2 gave this a high rating on both scales.

A History of Everything the User Has Read:

Assuming that the system keeps track of every article the user has read, the system would generate summaries of how often certain Topics, Sources, etc. appeared in selected articles. Not only would there be an explicit list of the actual articles, but charts and graphs of what has been read for the past week/month/year. A variation of this feature is currently used by other user modelling programs as a reply to the user’s question: Why did you suggest this?

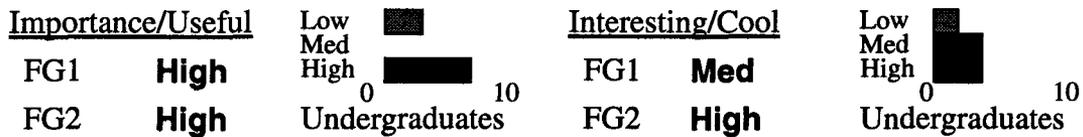


Focus Group #1 thought it was not useful information and did not help them ensure the accuracy of their current user model. Moreover, there is a privacy issue regarding the potential for other parties (hackers or lawyers) to gain access to this very personal information. Focus Group #2 thought this feature did not serve much of a useful purpose but instead satisfied some curiosities about their past actions. They mentioned that historical summaries could be important if they had to pay for news and could use the history information to help them budget their time/money.

Focus Group #2 gave this a high rating on the interesting/cool scale because this information could provide insight into how they have been spending their time with news. They also mentioned that looking at historical summaries would be a cool way to look back at one’s life. Conversely, Focus Group #1 suggested that it is not very interesting to read long lists even if there were charts and graphs to summarize the information.

Direct Manipulation of Model Items [on the Model Planes]:

What if there were some way to alter preferences by selecting an item off a model plane and changing it directly? Without this feature, changing preferences would require a dialog box or similar interface. As a response to any change made, there would be an option for loading a new newspaper on the fly.



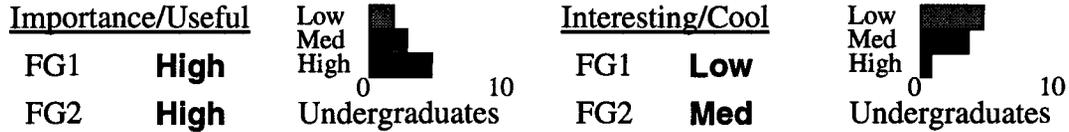
Both groups felt direct manipulation is an essential component to the user interface and should not be overlooked. Focus Group #1 wanted controls to crank up any preference, throw it away, or add new preferences. Focus Group #2 noted that using a dialog box interface would be an added level of abstraction. If users had a good mental model of how the Rolodex of model planes worked, then they should be able to affect that model directly and not be sidetracked through another user interface mechanism.

Focus Group #2 thought being able to directly manipulate the Rolodex (pushing items around on the planes and pulling items on and off) would be very cool. Conversely, Focus Group #1 thought that watching the newspaper change as the result of an action would be interesting, but just watching an item be directly manipulated on a model plane would be rather mundane.

Explicit Control Over the Implicit Information:

In the current system it is assumed that the user has control over the explicit choices for Topic, Type, Nature, and Source. The user should also have the ability to assign weights to the various model planes (e.g., perhaps the explicit preferences should be weighted twice as heavily as the implicit preferences). Assume that the user modelling program is in tune with the user's behavior; it has been drawing accurate conclusions about the user's preferences for reading news. Also assume the system has been building up this model through numerous observations over a long time. Should the user

have the ability to explicitly override the ratings that the user modelling system has derived through long-term observation of the user's behavior?



Both groups agreed that there needs to be a way to tell the user model that the user does not agree with its inferences. Each group cited examples where an override option seemed necessary to correct misconceptions the system may have drawn. A common example involved correcting inferences that were drawn when the user lived in California, but are now false because the user moved to Massachusetts. Such a change would affect many of the Sources and Topics that a user prefers. Although it was assumed the system would adapt to this demographic change, the algorithm may take a while to make such a drastic change to the user's news.

Focus Group #1 believed that the users must have complete control of their user model. They felt that allowing the user to make changes that override the system's inferences would be a useful way for people to experiment with their user model, learn how it works, and gain trust in the system.

Focus Group #2 found the issue of overriding the system to be very controversial. The point was made that the representation of the implicit information should be a coherent representation [of the system's implicitly derived model] and should not be contaminated with the addition of explicit influences on top of it. It was suggested that implicit items that are being explicitly changed by the user should be somehow distinctive (perhaps in a different font).

Focus Group #1 did not think that making changes to the user model would be fascinating to watch. Focus Group #2 suggested that it would be interesting to correct the system but would be too "academic" for on-lookers to find cool.

Both focus groups and a strong majority of the undergraduate evaluators (at least 7 of 10) agreed that the following features would be very important/useful:

Confirmations and Contradictions

Seeing the Articles that were Excluded

Direct Manipulation of Model Items [on the Model Planes]

The focus groups differed from the undergraduates; the focus groups had a strong preference for *Explicit Control Over the Implicit Information*. While the undergraduate evaluators thought that *Creating an Interpolated/Merged User Model* would be a more useful feature.

Both focus groups and a strong majority of the undergraduate evaluators agreed that *Visualizing How the Model Has Changed Over Time* would be very interesting/cool. The focus groups also had a strong preference for *Looking at Other People's User Models* that the undergraduates did not share. Furthermore, the undergraduates gave their highest interesting/cool ratings to *An Animation of Scoring* and their lowest to *Showing Matches Between an Article and the Model Planes*.

Evaluation: New Designs

Since the most confusing model planes were the Implicit plane and the Implicit Communities planes, re-designing these planes was the agenda for a meeting of each focus group. The new designs proposed in these meetings were mocked-up using Diagram!2 on a NeXT workstation. Each undergraduate evaluator was asked to rank the new designs on three independent scales: clarity, extensibility, and aesthetic design. An explanation of each scale was provided. The image files of plane designs were kept on-line for the undergraduates to review at their leisure. Evaluators provided individual ratings but they were allowed to discuss the designs and ask questions at any time.

Another product of the focus group meetings was a set of suggestions for new planes. These appear at the end of this section.

Improving the Implicit Communities Plane

The Implicit Communities plane is based on the idea that there are people who have common reading interests. The system forms clusters of readers who have common interests into communities and labels each community with a triplet (three items that describe the common interests of its members). There is a way to measure the distance between “you” and each community. The triplets that appear near the word “you” are closer to the user’s interests than those farther away. Note the information conveyed in each design should be the same.

Each undergraduate participant rated the designs on the following scales:

The Scale of Clarity:

By glancing at the plane, can you immediately tell which communities are close to you and which are far away?

(1) Confusing vs. (5) Clear

The Scale of Extensibility:

Currently, there are only a few communities that are close to you in the representation. What if 5-10 communities were close to you; would the layout remain clear or get jumbled? Would there be too many triplets bunched up for you to make sense of the plane?

(1) Likely to get muddled vs. (5) Likely to remain clear

The Wave with a Fisheye View was ahead of its competition on each scale and in the cumulative scoring.

Improving the Implicit Plane

The Implicit plane displays inferences that the system has made about the user's preferences by observing which articles the user reads (i.e. implicit feedback). There are two axes: preference and confidence. Confidence grows when users confirm their preferences by reading (or ignoring) the same item over time. If there is a lot of contradiction in the data or there are very few data points, then the system computes a low confidence level for that inference. This is important because the system draws numerous inferences that are so weak (low confidence) that a PeerGlass user should not be distracted by them. Note the information conveyed in each design should be the same.

Each undergraduate participant rated the designs on the following scales:

The Scale of Clarity:

By glancing at the plane, can you immediately tell which items are high preference, high confidence? (In this example, "Hillary Clinton" and "Poetry" are high confidence and high preference, while "Movies" is medium confidence and high preference.)

(1) Confusing vs. (5) Clear

The Scale of Extensibility:

Currently, there are only a few items on each grid. What if there were 30 more items? In most cases, you would want to pay attention to the "high confidence, high preference" items; and ignore the "low confidence" items. (There are probably at least 100 low confidence items in a normal model.) When looking at a grid design, consider where the pools of low confidence items would form. Would they get in the way of the high confidence items?

It is safe to assume that there is a Z-axis so that you can zoom farther into the plane to see each item. The items that are dark would be farther away than the items that are light-colored.

(1) Likely to get muddled vs. (5) Likely to remain clear

The Scale of Aesthetic Design:

Do you think the design idea is clean or elegant? or is the design clunky and not a pleasure to look at?

(1) Ugly vs. (5)Crisp

The first six designs re-define how information is distributed across planes. Each new design assumes that there is one plane for Topic, one plane for Nature, one plane for Type, and another for Source. On each of these planes, both the implicit and the explicit information is listed side-by-side to allow for comparison. Also, there is now a preference axis for explicit information (previously all explicitly declared items shared equal preference).

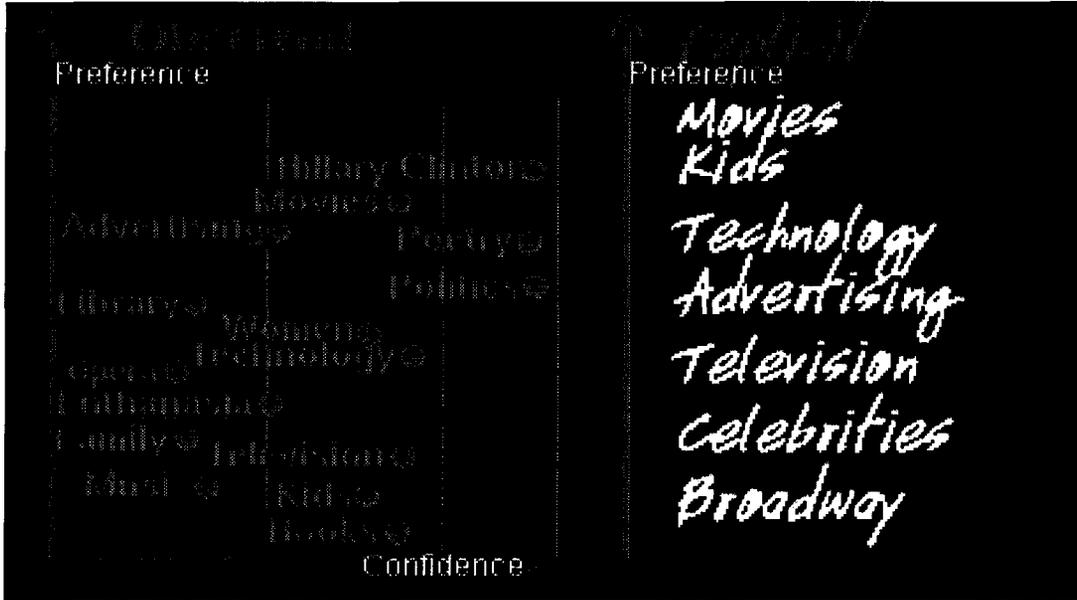
The first design, *Gridless*, puts the implicit information on the X & Y axes. Variations in color and size emphasize the confidence scale. Dots are used to pinpoint the position of an item on the grid but actual gridlines are not shown.

Clarity 33 Extensibility 21 Aesthetic Design 26 Cumulative Total 80



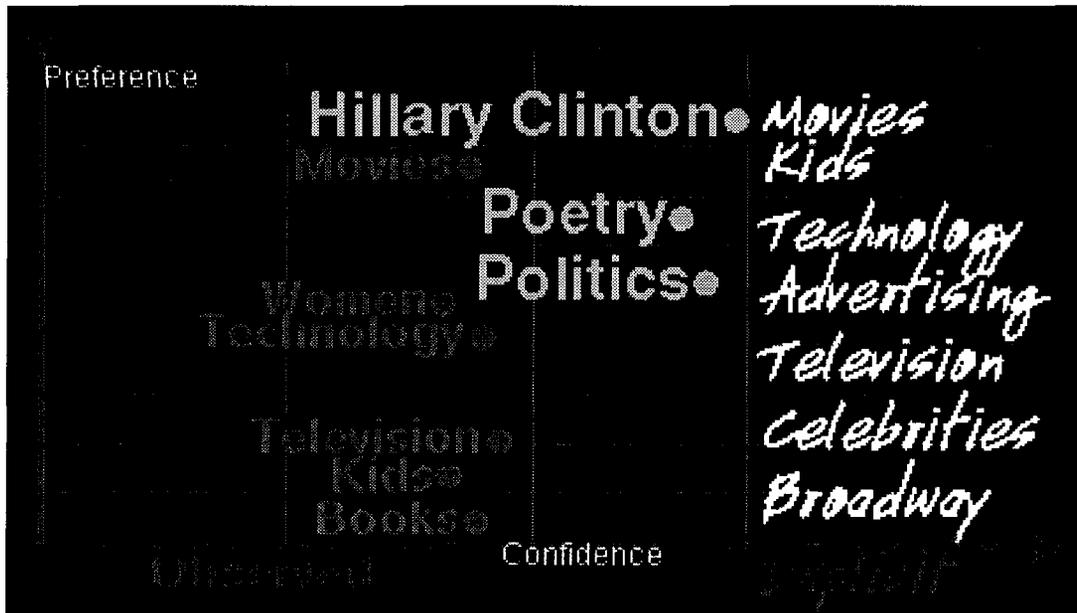
Same Font is a cut-and-dry approach with no changes in color or size. Additionally, gridlines appear between the axes.

Clarity **32** Extensibility **25** Aesthetic Design **29.5** Cumulative Total **86.5**



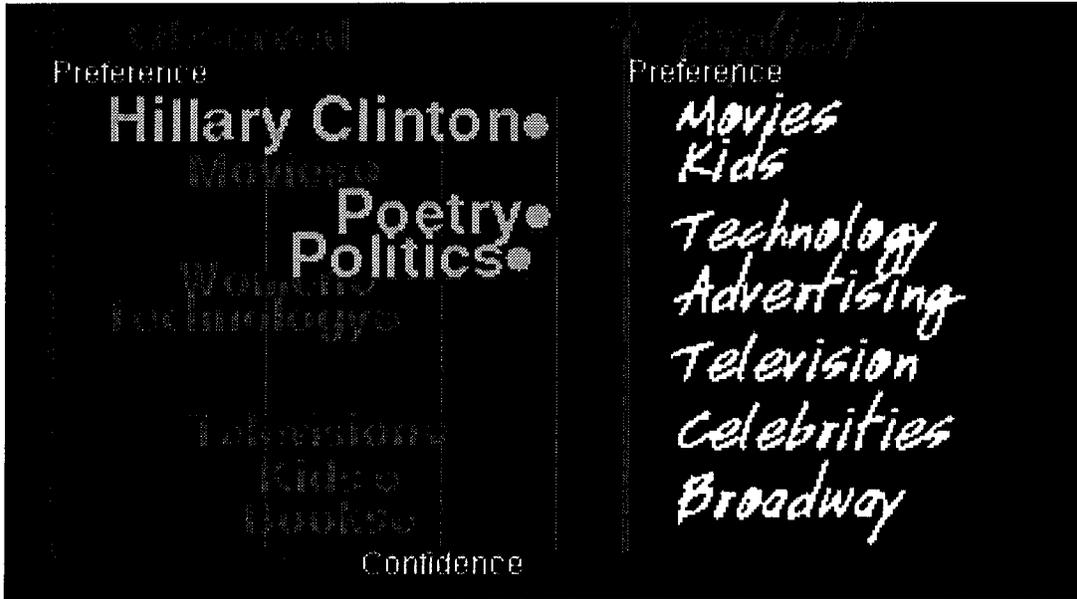
Stretched is a combination of the previous two designs. There are gridlines, plus variations in size and color. The additional feature is a stretching of the X-axis to allow for more of the plane to be consumed by the grid.

Clarity **32** Extensibility **24** Aesthetic Design **29** Cumulative Total **85**



Standard is the same as *Gridless* with the addition of gridlines.

Clarity **32** Extensibility **19** Aesthetic Design **28** Cumulative Total **79**



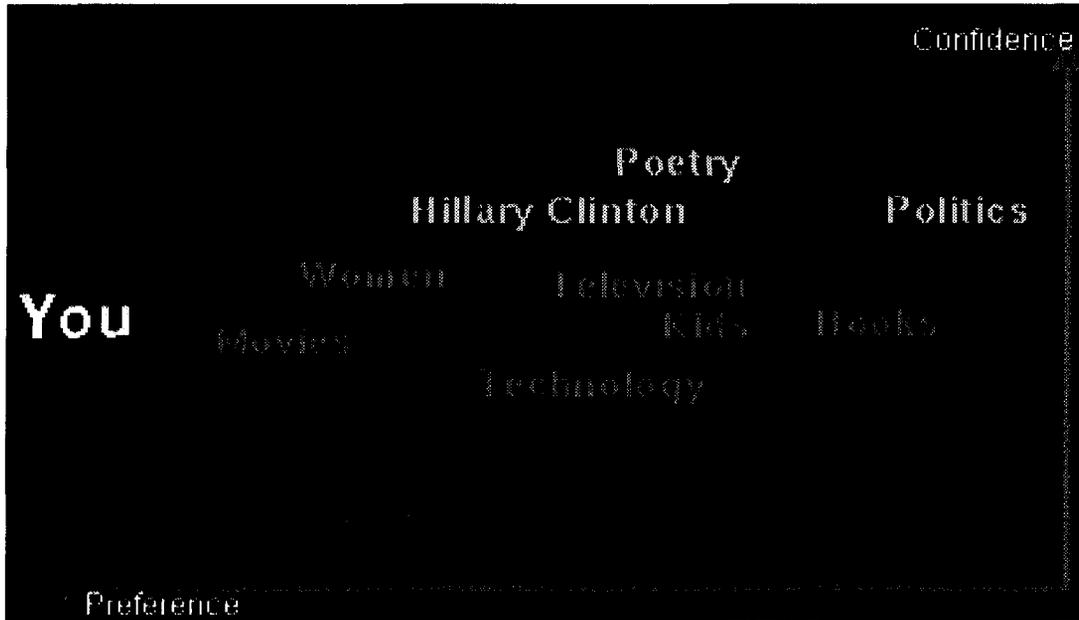
X is Recency is based on *Standard* but adds a new axis to the design. Items that have appeared more recently in the news are distinguished from items that have not. Hopefully this will segment the items on the plane to draw more attention to the current news. The Z-axis is labelled, “confidence,” as the size and color variations are an attempt to artificially convey depth.

Clarity **29.5** Extensibility **20.5** Aesthetic Design **23** Cumulative Total **73**



The *Topic Wave* is very similar in appearance to *The Wave with a Fisheye View* shown above. The axes have been altered to match the new information. High preference items appear on the far left, closer to “you”. Confidence is on the Y-axis, but this leads to a strange flaw where a high confidence, high preference item could be farther from “you”, then a high preference, medium confidence item.

Clarity **37** Extensibility **37.5** Aesthetic Design **36** Cumulative Total **110.5**



The next six designs follow the original breakdown of planes. That is, the implicit information for Topic, Source, Type, and Nature all appear on a single plane. These designs try to reduce the dizzying clutter of the current implementation.

Confidence Up keeps the higher confidence items near the top of the gridlines.

Preference is represented by color brightness so that low preference items are dim.

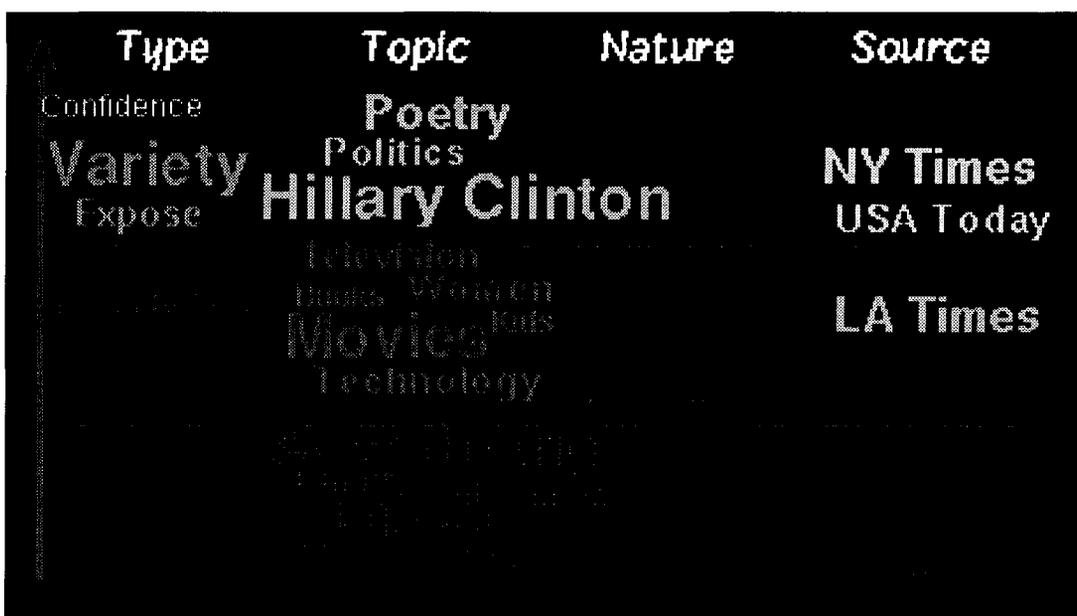
Clarity **32** Extensibility **25** Aesthetic Design **29.5** Cumulative Total **86.5**



Preference is Size makes the more preferred items increasingly larger than others.

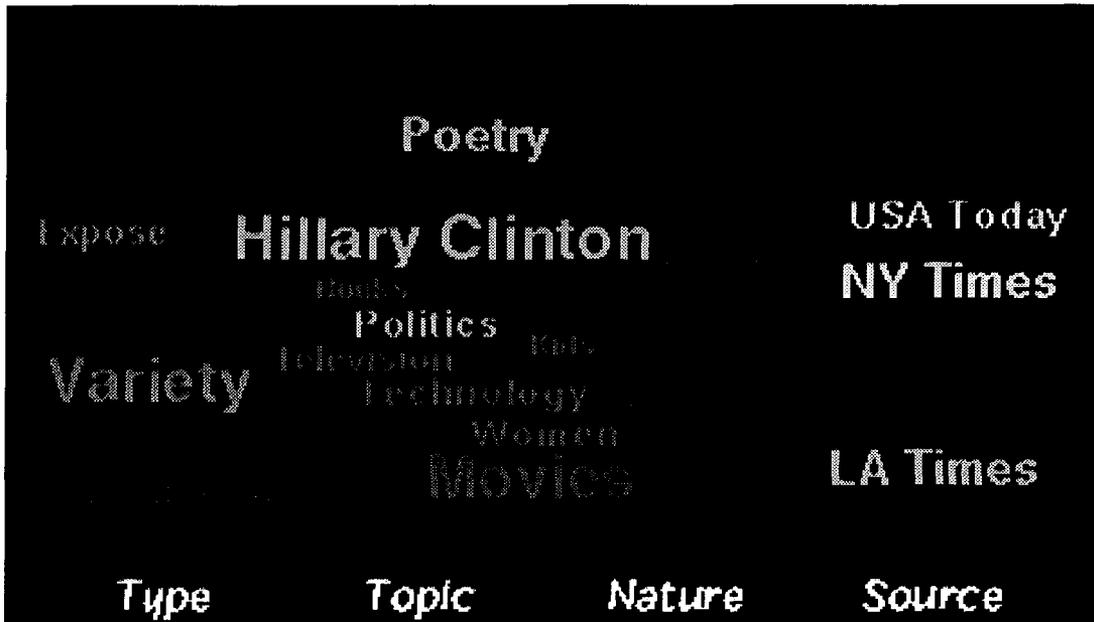
Confidence is on the Y-axis (with gridlines) but color brightness keeps the low confidence items dim. Moreover, the X-axis is used to sort the items into the categories of Type, Topic, Nature, and Source with a dark, color-coded background behind each.

Clarity **36** Extensibility **24** Aesthetic Design **30** Cumulative Total **90**



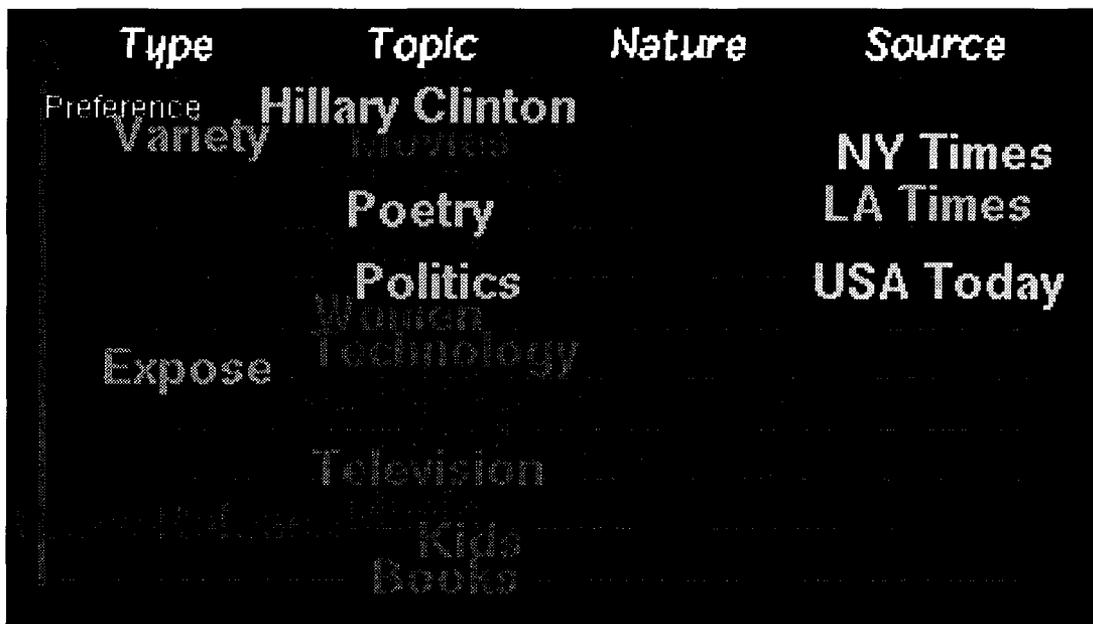
Position Meaningless maintains that size is preference while color brightness is confidence. Although Topics are roughly grouped by category, position is meaningless.

Clarity **18.5** Extensibility **23** Aesthetic Design **27** Cumulative Total **68.5**



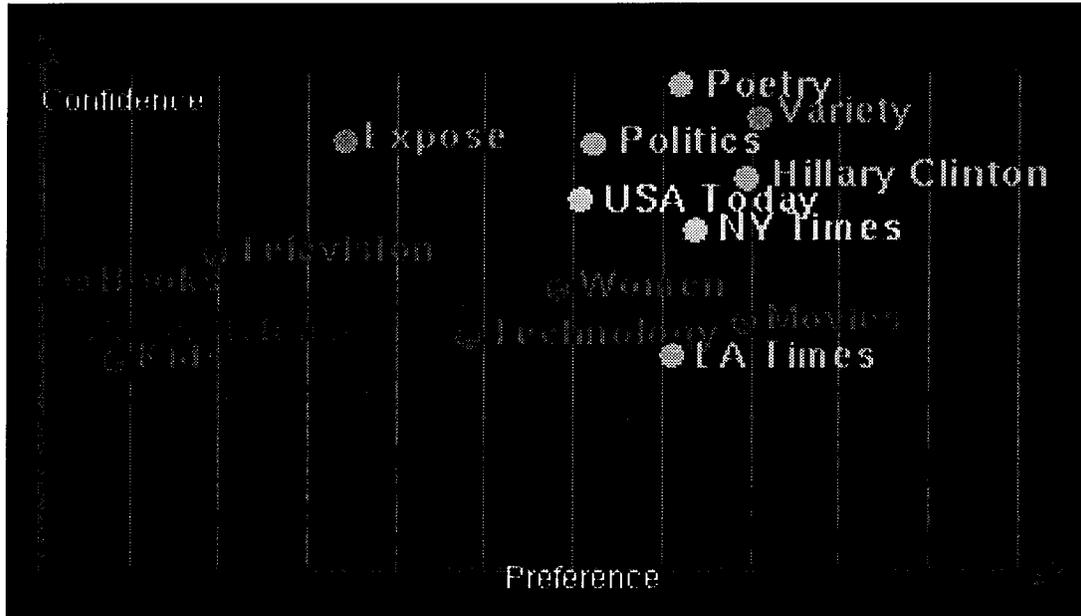
Preference Up uses the color-coded background, gridlines, and sorting-by-categories shown in *Preference is Size* but all of the words appear the same size. The color brightness represents confidence.

Clarity **37** Extensibility **25.5** Aesthetic Design **31** Cumulative Total **93.5**



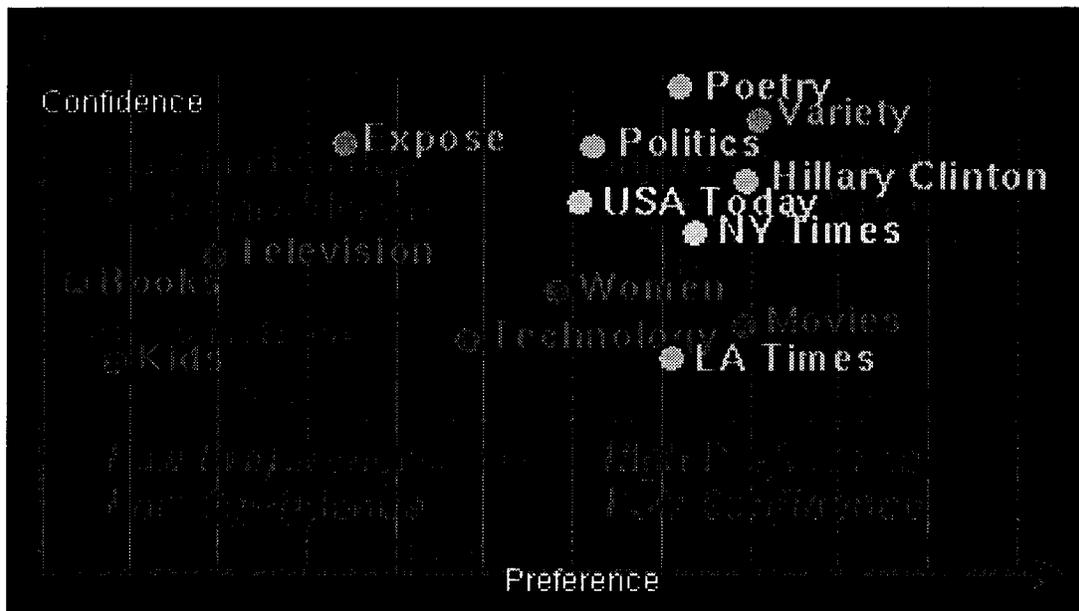
Standard Axes uses color brightness for confidence but keeps all of the items the same size. Confidence is the Y-axis and preference is on the X. Gridlines are shown too.

Clarity **31** Extensibility **23** Aesthetic Design **25** Cumulative Total **79**



Standard Axes w/ Quadrants is the same as the previous design except there are labels marking each quadrant. For instance, the upper-left quadrant is labelled “Low Preference, High Confidence”.

Clarity **26.5** Extensibility **18** Aesthetic Design **23.5** Cumulative Total **68**



The top five new plane designs according the undergraduate evaluators are:

Topic Wave (cumulative total 110.5)

Preference Up (cumulative total 93.5)

Preference is Size (cumulative total 90)

Same Font (cumulative total 86.5)

Stretched (cumulative total 85)

These planes (in varying orders) also held the top spots on all three scales. To help determine which design aspects users prefer, they were asked to mark what they like and dislike from a list of design aspects. For a few of these design aspects, at least six of the ten evaluators agreed:

Good Design: labelling axes; varying word size; varying shades of color

Bad Design: dots marking grid location; X-axis as “recency”

Another issue the undergraduate evaluators considered was the breakdown of planes. Some of the designs suggested new distributions of model planes in the Rolodex. The users were asked to rank three different breakdowns.

Their favorite way to separate the information is to have the implicit and explicit information about Topics on one plane; likewise for a Nature plane, a Source plane, a Type plane, and then, as usual, an Implicit Communities plane.

Second, they preferred the currently implemented breakdown: one plane for implicit Topic/Type/Nature/Source; one plane for explicit Topic/Type/Nature; one plane for explicit Source; and one plane for Implicit Communities.

A third method was derived from the current implementation. In this case, the Implicit plane was divided into four separate planes, one for each category. The resulting breakdown included three more planes per Rolodex than the current implementation. This method was strongly disliked by seven of the ten undergraduates.

The irony of these ratings is that the *Topic Wave*, which earned the highest ratings of any of the Implicit plane designs, could only appear using the third and least-liked breakdown of model planes. It is doubtful that users realized this contradiction while marking their opinions. The high preference for the *Topic Wave* probably had to do with the fact it contained the least information of any of the designs and therefore seemed clearer. Another reason why the *Topic Wave* drew so much attention is that it appears visually similar to the highly-ranked community design, *Wave with a Fisheye View*.

Although the *Topic Wave* design is appealing, the representation is somewhat confusing. The wave for communities shows the distance between “you” and various communities. This is not quite the same as showing preference as a distance from “you”. Another problem is that the design is shown with two axes at right angles, plotting information along the curves of the wave is misleading. Considering these problems, it should not be assumed that the *Topic Wave* is the best design. Each of the top five designs for Implicit planes should be iterated for further improvement and then re-evaluated.

One user identified the third breakdown of model planes as having “too many” planes. This is surprising because a key strength of the Rolodex is its extensibility for adding new planes. This user’s opinion must be explored further because extensibility is the primary advantage of the Rolodex metaphor, yet if a consensus of users are opposed to numerous planes then it is necessary to find a more useful navigation metaphor[20].

Suggestions for New Planes

One of the graduate student focus groups made suggestions for new planes:

A plane for collaborative filtering showing the reader whether an article has been pre-approved by a community. [10][13][8]

Additional planes for implicit communities that are layered according to a physical metric. For instance, there could be separate planes for communities on a local level, national level, or international level. The design representation should be similar for each layer. A way to seamlessly zoom through the layers would be a bonus. [18]

A plane for representing explicit, demographic community influences (e.g, gender, age, or residence).

A “timely interests” plane that allows users to set a start/stop date on a set of explicit interests. A significant population of newsreaders work on limited-timeframe projects that create a need for different news based on their current project. A further suggestion would be to have an integrated calendar program that implicitly added explicit interests (with a limited timeframe) based on the user’s appointments. [11]

New designs and new planes are accommodated by PeerGlass’ flexible architecture. Details of the current implementation and how to extend it are described in the last section of this document.

Retrospective

During the evaluation of PeerGlass it became clear that although many of the features were easy to use and some were easy to understand, certain design changes are necessary to clarify the representation of the user's model. These changes will make the visualization of the user's model easier to understand and bring PeerGlass closer to achieving its goal of facilitating trust between the user and the system. None of these changes affect the flexibility of the PeerGlass architecture to visualize user models for any domain of personalized information.

Rolodex

The metaphor of a Rolodex was intended to convey that a growing set of information could be attached to the user model. The analogy to a physical object helped clarify the user interface because most people know that to access any part of a Rolodex, just spin it to the desired place. Using an abstract design or uncommon metaphor would require more explanation on how to access the information. The choice of metaphor emphasized that the order was circular so no single plane would seem "in front" or more important. Consequently, it does not make sense to weight the planes differently, although this customization is necessary for improving the user's control over the model. Also, the circular design and dial-based interface prevent random access to a desired plane. Even with the arrow keys, it may take multiple key presses to reach a particular plane.

The assumption that the Rolodex was a good design choice because of its flexibility for additional planes may be misguided. The response of the undergraduate evaluators suggested that users are not interested in having more than a few model planes on their Rolodex. Another false assumption was that being able to see the "off-sides" of the Rolodex would be an interesting preview of the neighboring planes (Figures 7-10 show this best). Many users reported that this information was confusing and distracting.

Adding a "gravity" feature is necessary before more planes can be added. When there are only three or four planes it is possible to get cameras lined up with a clean view of each plane. If more planes are added, the camera angles become intractable. The gravity feature would simulate a physical Rolodex where the rest of the planes fall away behind the current plane after the spinning has concluded. This option would also avoid the "off-sides" problem.

From an implementation perspective, limiting the Rolodex to such a small piece of screen real estate was a bad design decision. Having the Rolodex in a separate window (still with the totem poles of the currently selected articles visible) would allow for more information to be represented in a large, clear manner and would be more captivating for the user. The Rolodex would also be more fun if the hardware were fast enough to allow for smooth spinning.

The value of the Rolodex is severely limited until the designs of the model planes are more coherent and reasoning is shown on them. The user study confirms that providing the user with a view of the Rolodex showing all the matches for a certain article and the scoring results would be very helpful for understanding how an article was selected. A canned, animated demonstration of a sample article being compared to each plane would also be a very valuable training tool but should not be considered a necessary feature.

Model Planes

The evaluators of model planes have suggested a number of clearer designs. Changing the design of a model plane requires simply changing the layout algorithm for the plane. Nevertheless, there are logistical problems with model planes that go beyond the layout routines. The choice to keep the Rolodex limited to one-third of the screen real estate restricted the model planes to such a small space that it was detrimental to the quantity of information shown and the clarity of that information.

To prevent model planes from becoming too full of information, it is necessary to add depth to the representation. The more important and useful information should always be presented upfront without the less important information cluttering the view. Combining rotation and zooming functionality could also be a very provocative way for seeing the “behind the scenes” reasoning that is being applied to the user modelling data. Zooming deeper for information was not part of the current implementation of PeerGlass for technical reasons. It did not seem programmatically possible at the time to allow the users the ability to rotate the Rolodex and zoom into a plane. A suggestion for achieving this is presented in the *Implementation Details* section of this document. Other impediments to the design of model planes included hardware that was too slow to handle smooth fonts, texture maps, images, curved 3D shapes, or even anti-aliasing. The current implementation attempts to use depth as an axis in some of the model plane designs, however, all 3D information is lost when, to maximize legibility, the camera focuses on just one plane head-on.

The model plane concept is still valid even when the design of planes are altered and the information stored on the planes is changed. User models will always be complex sets of statistics and somehow the models have to be presented to users in understandable segments. Even if the user model is not limited to words, it is necessary to still divide the information across multiple planes. Representing the information on planes provides the user with a sense of their user model containing a tangible amount of information. Still, it is important that the quantity of planes is not overwhelming.

Totem Poles

Many users asked what a totem pole offered that could not simply appear as a list. The answer is that the totem poles distribute the information onto four sides, thereby displaying as much as four times the information that could be contained in a list. Moreover, no matter how they are viewed, totem poles are fun. Nevertheless, the trade-off between screen real estate and totem poles is controversial. In the event that enough space exists to have a sidebar and a column of totem poles beside the articles, then it makes sense to keep as many totem poles as possible. If the space is not available, then a sidebar with a single totem pole is a reasonable approach (Figures 18 and 21).

The small size and screen space of each totem pole restricts the quantity of information that can legibly appear. The layout of the words on each side of the totem pole is non-optimal. Most words are legible (especially when the arrow keys are used for navigation) but the angles for the Topics and Types should be improved. The size of totem poles, the variety of words, and the slowness of the hardware prevented 3D icons from being used in place of words. The intended use of icons was the inspiration for calling these widgets “totem poles”. In retrospect, using icons may only confuse users because very few icons exist to convey specific information about Topics, and even fewer icons exist that could represent the Type or Nature of an article.

Using totem poles as browsing tools restricts the ability of the widget to provide an answer to the question, “Why was this article selected for my newspaper?” To use the totem pole for browsing would require the addition of an exhaustive list of the article’s features (independent of whether they match the user’s model). Such a list would impede the user’s ability to determine why an article was selected. Additionally, showing more reasoning data on a totem pole, not just words, may prove a benefit to answering the user’s inquiry, but such additions should not be allowed to clutter the content of the totem pole.

Bibliography

Users and Their User Models

- [1] S Bull, H Pain, and Paul Brna. Student Modelling Beyond Domain Knowledge. *Proceedings of the Fourth International Conference on User Modelling*, pg 197. August 1994.
- [2] R Cook and J Kay. The Justified User Model: A Viewable, Explained User Model. *Proceedings of the Fourth International Conference on User Modelling*, pg 145. August 1994.
- [3] J Orwant. Apprising the User of User Models: Doppelganger's Interface. *Proceedings of the Fourth International Conference on User Modelling*, pg 151. August 1994.
- [4] B Sheth. A Learning Approach to Personalized Information Filtering. Master's Thesis, MIT Media Lab, 1994.
- [5] A Sutcliffe and A Old. Do Users Know They Have User Models? Some Experiences in the Practice of User Modelling. *Proceedings of the Second IFIP Conference on Human-Computer Interaction: INTERACT 87*, pg 35.

User Modelling Systems

- [6] R Kass and T Finin. A General User Modelling Facility. *Proceedings of CHI 88*, pg 145.
- [7] J Orwant. Doppelganger Goes to School: Machine Learning for User Modelling. Master's Thesis, MIT Media Lab, 1993.

Feedback, Inferences, and Collaborative Filtering

- [8] P Chesnais and J Sheena. The Fishwrap Personalized News System. To appear in the Second IEEE International Workshop on Community Networking. May 1995.
- [9] T Erickson and G Salomon. Designing a Desktop Information System: Observations and Issues. *Proceedings of CHI 91*, pg 49.
- [10] D Goldberg, D Nichols, B Oki, and D Terry. Using Collaborative Filtering to Weave an Information Tapestry. *CACM*, pg 61. December 1992.
- [11] M Marx. Toward Effective Conversational Messaging. Master's Thesis, MIT Media Lab, 1995.

- [12] E Rich. Users are Individuals: Individualizing User Models. *Intelligent Information Systems: Progress and Prospects*, Editor R. Davies, Halsted Press. 1986.
- [13] U Shardanand and P Maes. Social Information Filtering: Algorithms for Automating “Word of Mouth”. *Proceedings of CHI 95*, pg 210.

Audiences and User Communities

- [14] J Donath. Visual Who. MIT Media Lab Technical Report. January 1995.
- [15] Times Mirror Center for the People & the Press. *Technology in the American Household*. Times Mirror Technical Report. May 1994.

Navigating Information in 3D Space

- [16] J Abrams. Muriel Cooper’s Visible Wisdom. *I.D. Magazine*, pg 48. September/October 1994.
- [17] E Rennison. Galaxy of News: An Approach to Visualizing and Understanding Expansive News Landscapes. *Proceedings of UIST 94*, pg 3.
- [18] D Small, S Ishizaki and M Cooper. Typographic Space. *Proceedings of CHI 94* (Video Program), pg 437.
- [19] L Strausfeld. Financial Viewpoints: Using Point-of-View to Enable Understanding of Information. *Proceedings of CHI 95* (Short Paper), pg 208.

Usability and User-Centered Design Issues

- [20] A Marcus. Human Communications Issues in Advanced UIS. *CACM*, pg 101. April 1993.
- [21] D Norman. How Might People Interact with Agents. *CACM*, pg 68. July 1994.
- [22] D Wildman. Getting the Most from Paired-User Usability Testing. To appear in *INTERACTIONS Magazine*. June or July 1995.

Note that some of the citations above belong in more than one category of this bibliography. The categories reflect how the citation is primarily referenced in this document.

Implementation Details

Basic Flow of the System

The Main() routine of PeerGlass does the following:

- Initialize main window
- Set up fonts
- Set up the root of the Rolodex (it has a different root switch for each section)
- Scan all article files by loading them into article reference structures
- Set up the struct of all model items (draws from the user modelling data files)
- Select articles by comparing the set of articles to the model for each section
- Build a set of model planes for each section
- Build the newspaper (just sets up the Motif widgets)
- Initialize the current article and section indices
- Display the current news by filling in the six article and headline widgets
- Set the initial state of the newspaper
- Initialize the 3D viewer for the Rolodex (including the camera angles)
- Set up all of the actions and callbacks for the newspaper
- Show the main window
- Launch the event loop

List of possible events:

To hide the article pop-up, click the article pop-up's close button or the article pop-up's background.

To hide the articles and display the model planes for the current section, click the Examine Profile button.

To exit the program, click the Quit button.

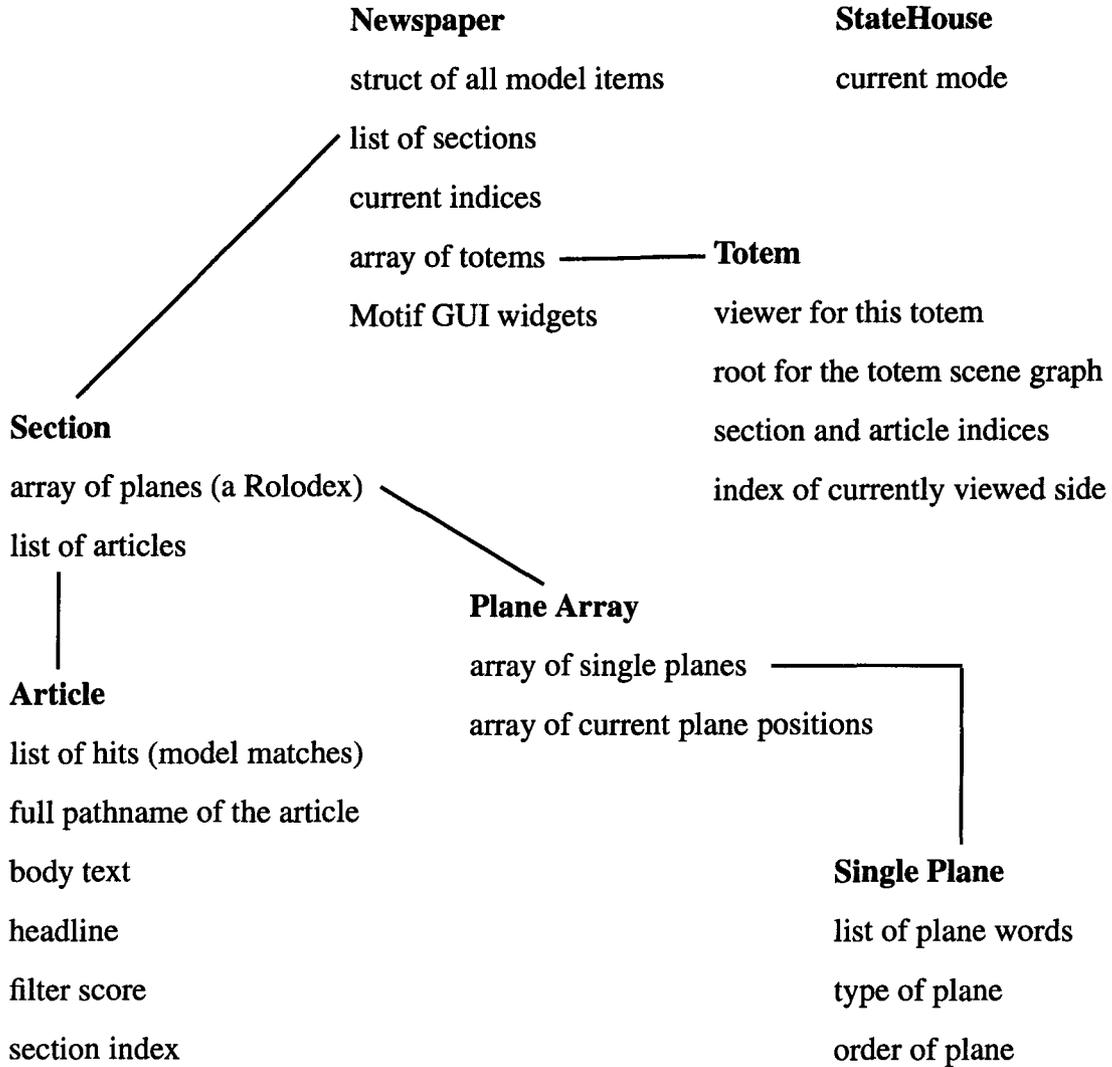
To open the article pop-up, click on the body of an article or its headline (above the article or above its totem).

To display the articles of a section click the button for that section.

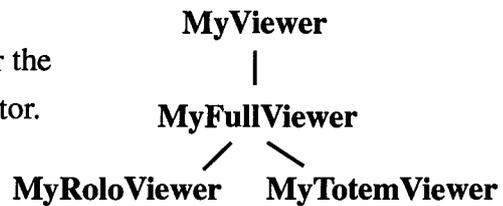
Both the mouse and keyboard input options for the 3D viewers can be found in MyTotemViewer::processEvent(XAnyEvent *xe) and MyRoloViewer::processEvent(XAnyEvent *xe).

Overview of Classes

Class names appear in boldface. Only the most influential objects in each class appear in this diagram.



These viewers are modifications of the code for the MyExaminerViewer provided with Open Inventor.



Article Format

PeerGlass is ready to accept new articles. Without text-understanding routines that can automatically tag more than the Topic of articles, there is not much point to visualizing user models. Still, hooking a wirefeed to PeerGlass is rather easy. First, the #define for the ARTICLE_DIR_PATH should be updated. Another important note is that currently the scan_articles() routine is looking for filenames starting with a digit. The format of each article must be:

```
<SectionName>  
STARTTAGS  
<category> : <tag>  
ENDTAGS  
<Headline>  
<article body>
```

There can be 0 to MAX_HIT_NUM <category> : <tag> pairs. Examples of these pairs include:

```
USA_today : page_one  
boston_globe : sectn_front  
nature : liberal  
nature : scandalous  
type : interview  
type : variety  
topic : football  
topic : shoes
```

Note there must be one space on each side of the colon. A last point to consider is where to set the threshold on the filter to adjust for an optimal set of articles. The thresholding is done at the bottom of Newspaper::filter_score().

User Model Format

PeerGlass needs mechanisms for collecting the user's feedback both implicitly and explicitly. There are no hooks currently in place for these mechanisms or their algorithms but they could be attached to the action table and set of callbacks [in main() of pg.c++] to record the user's implicit actions and explicit feedback. Reasoning algorithms need to be written to interpret this data and draw inferences (about the individual and about relevant communities). For immediate updates to the newspaper it is necessary to change the major_model_struct and re-filter the articles for the chosen section by using a single-section variation of the Newspaper::select_articles() routine.

As long as there is feedback, user input, and new inferences it is necessary to update the user modelling data files (in MODEL_PATH_DIR). Each line of the user modelling data file must begin with a code for the model plane (assigned in news.h). Subsequently, the format of each line in the file is determined by the switch statement in Newspaper::setup_model_struct . This switch parses the rest of the line from the user modelling data file according to the line parsing routine for the type of plane. Parsed data is put into the fields of a model_data_struct and added to the major_model_struct. The available fields are listed in news.h. For example, the currently implemented Implicit plane keeps its data in the form:

4 implicit topic florida 4.2

that is modeltypeID, typetag, category, label, weight, and confidence. While explicit preferences are currently stored as:

3 explicit topic boston

which is modeltypeID, typetag, category, and label. Each model plane has its own set of relevant features. The flexibility of the Rolodex accommodates new additions.

Adding New Model Planes

Each model plane requires a layout algorithm and a position in all of the switch statements for model planes. Building a good-looking layout routine that can tolerate being seen at all positions on the Rolodex can be frustrating. Open Inventor comes with the SceneViewer application that loads iv files and supports direct manipulation of scene objects. This is the best way to figure out translations, rotations, and lighting. Also the Noodle tool is a modelling program that exports iv files but is very buggy. These tools are found in /usr/share/src/Inventor/demos/ . The routine to export the current scene graph to iv format is in MyRoloVwr.c++ . It is also helpful to setDecoration(TRUE) in main() of pg.c++ to gain access to the ExaminerViewer controls while working with a new model plane. Lastly, pressing 'c' at any time while the cursor is in one of PeerGlass' viewers will activate a get_cam() routine that will print all of the variables for the current camera to standard out.

The new plane must also be added to the user modelling data files (described above) and be assigned a #define in planes.h. Besides the layout routine for the new plane, the following routines need to be aware of the new plane:

Newspaper::collect_hits() compares the features of the article and the model.

Newspaper::setup_model_struct() reads the user modelling data files and parses their contents to be added to the proper slots in the major_model_struct.

SinglePlane::make_plane() fills in the descriptive text for the model plane.

SinglePlane::make_words() calls the layout routine and attaches the node it returns.

Totem::do_type_words(), do_topic_words(), and do_nature_words() all switch on the type of model plane while gathering the hits to attach to the totem pole.

Navigating the Model Planes

A more ideal method for navigating model planes was discussed during one of the focus group meetings. The rotator dial should remain as is, but the camera angles should be altered to avoid seeing parts of planes that are not the current plane. One way around this is to implement “gravity” for the planes by installing a true Rolodex simulation where only one plane is visible at a time and the rest sit in a stack behind the current plane.

The focus group also suggested adding a zoom slider that allows the user to zoom into the current plane. This slider could trigger a switch from an ExaminerViewer to a WalkViewer at the right time. Users should be able to use the mouse to pan or zoom the WalkViewer. The range of possible camera positions should be bounded so that the user does not get lost in 3D space. Returning to the higher level view (and the ExaminerViewer) could be accomplished with a keystroke or use of the rotator dial.

Description of User Interaction

The StateHouse::set_mode() routine manages three states corresponding to whether the user is viewing: a set of articles, one particular article, or the Rolodex. Based on the user’s state, the text at the top and bottom of the screen changes. It is possible for the user to move freely from any one state to any other state with a single mouseclick.

User Input Features

Currently there is no interface for adding explicit information. Some interface (a dialog box at the very least) should support users selecting their explicit interests with regard to Source, Type, Nature, Topic, and demographic communities. Also, the user should be able to assign preference weights to these interests. The planes of the Rolodex require a similar feature for specifying weights to customize which planes should be considered more important during the filtering process. Explicit feedback devices should also be attached to each article, see Figure 13. In the event that users are allowed a direct manipulation interface to the model planes, there must be some way to distinguish what is selectable and what cannot be altered by the user.

Enhancing the Newspaper

Two enhancements to the newspaper are required: an automatic layout system and a better way to browse through articles. An automatic layout system would have to adjust the size of the `art_grids` widgets in the `Newspaper::display_news()` routine. To improve article browsing the sidebar concept should be re-instated. It is actually possible for a sidebar to peacefully co-exist with more than one totem pole if PeerGlass had a wider piece of screen real estate. The sidebar feature is shown in Figure 18. Additionally, the pop-up window for individual articles must be fixed to better handle long text. Also, the `RoloViewer` for the model planes should probably appear as a pop-up window outside of the main newspaper window.

Upgrade Issues

Upgrading to the next version of Open Inventor should not break the code for PeerGlass. The demo will continue to execute without being recompiled. The 4,000 lines that are specific to PeerGlass should recompile easily as long as the improvements to the Inventor scene object classes are reverse-compatible. The 9,000 lines of code that are viewer-related and SGI-authored should be updated from `/usr/share/src/Inventor/samples/viewers`. Since the viewer files were modified for PeerGlass' use it is necessary to insert the same modifications in the upgraded viewer files. Each modification to the current files is noted with a "WAS HERE" comment. To ease the upgrading task, changes to the Inventor scene object classes are described in the upgrade's README files and in the Inventor on-line man pages. Note that PeerGlass uses the 4Dwm window manager.