Visualizing Digital Library Search Results with Categorical and Hierarchical Axes

Ben Shneiderman, David Feldman, Anne Rose, and Xavier Ferré Grau*

Human-Computer Interaction Laboratory
Department of computer Science and *Institute for Systems Research
University of Maryland, College Park, MD 20742
http://www.cs.umd.edu/hcil
Email: ben@cs.umd.edu

ABSTRACT

Digital library search results are usually shown as a textual list, with 10-20 items per page. Viewing several thousand search results at once on a two-dimensional display with continuous variables is a promising alternative. Since these displays can overwhelm some users, we created a simplified two-dimensional display that uses categorical and hierarchical axes, called *hieraxes*. Users appreciate the meaningful and limited number of terms on each hieraxis. At each grid point of the display we show a cluster of color-coded dots or a bar chart. Users see the entire result set and can then click on labels to move down a level in the hierarchy. Handling broad hierarchies and arranging for imposed hierarchies led to additional design innovations. We applied hieraxes to a digital video library of science topics used by middle school teachers, a legal information system, and a technical library using the ACM Computing Classification System. Feedback from usability testing with 32 subjects revealed strengths and weaknesses.

KEYWORDS: Digital libraries, graphical user interfaces, information visualization, hierarchy, categorical axes, hieraxes

INTRODUCTION

Digital library researchers face numerous challenges in helping users find what they want. Among them are designing meaningful overviews so that patterns can be easily recognized, creating comprehensible interfaces to specify what they want, and providing effective displays of search results. The visual information seeking mantra: overview first, zoom and filter, then details on demand provides a good starting point, but interpreting the mantra has produced a remarkably diverse set of research and commercial interfaces (Shneiderman, 1998; Card, Mackinlay and Shneiderman, 1999; http://otal.umd.edu/Olive). Within this body of work, two important strategies are (1) two-dimensional visualizations and (2) browsers for hierarchical data sets.

Two-dimensional visualizations with meaningful axes have been an effective visualization strategy because thousands of items can be shown at once. Typical axes make use of continuous variables (for example, relevance or publication date) or categorical variables (for example, language or name of journal). The labels on such axes are cues to searchers and may contain valuable information if they are ordinal variables. Two-dimensional visualizations with non-meaningful axes have been tried in systems such as BEAD (Chalmers and Chitson, 1992), BIRD (Kim and Korfhage, 1994), and ThemeScape (Wise, Thomas, Pennock, Lantrip, Pottier, Schur, and Crow, 1995) (commercial version is at http://www.cartia.com). Clusters of documents in a two-dimensional space can be meaningful, but users can make mistakes in assuming that distances between clusters have strong significance.

Hierarchical browsers are common and often successful, such as Yahoo! Hierarchies have the potential to reduce complexity by organizing related information into comprehensible structures. Some hierarchies, such as the Dewey Decimal System, are fairly straightforward trees. Others, such as the National Library of Medicine's Medical Subject Headings (MeSH), the Universal Medical Language System (UMLS) (Pratt, 1997), and the ACM Computing Classification System are partially hierarchical, multi-hierarchical (organized simultaneously into multiple hierarchies), or otherwise complicated. Hierarchies are typically shown by textual lists of one level at a time, by node-link diagrams, cone trees (Hearst and Karadi, 1997) or by an outliner in which levels and branches can be expanded and contracted (Nation, Plaisant, Marchionini, and Komlodi, 1997).

In this paper we apply hierarchical axes, also referred to as *hieraxes*, to combine the power of a hierarchical browse tool with a two-dimensional visualization. This combination preserves visual overviews and enables users to rapidly comprehend search results. We focus on search results that may include 100 to 10,000 items, but we anticipate that hieraxes may be useful for viewing much larger sets of

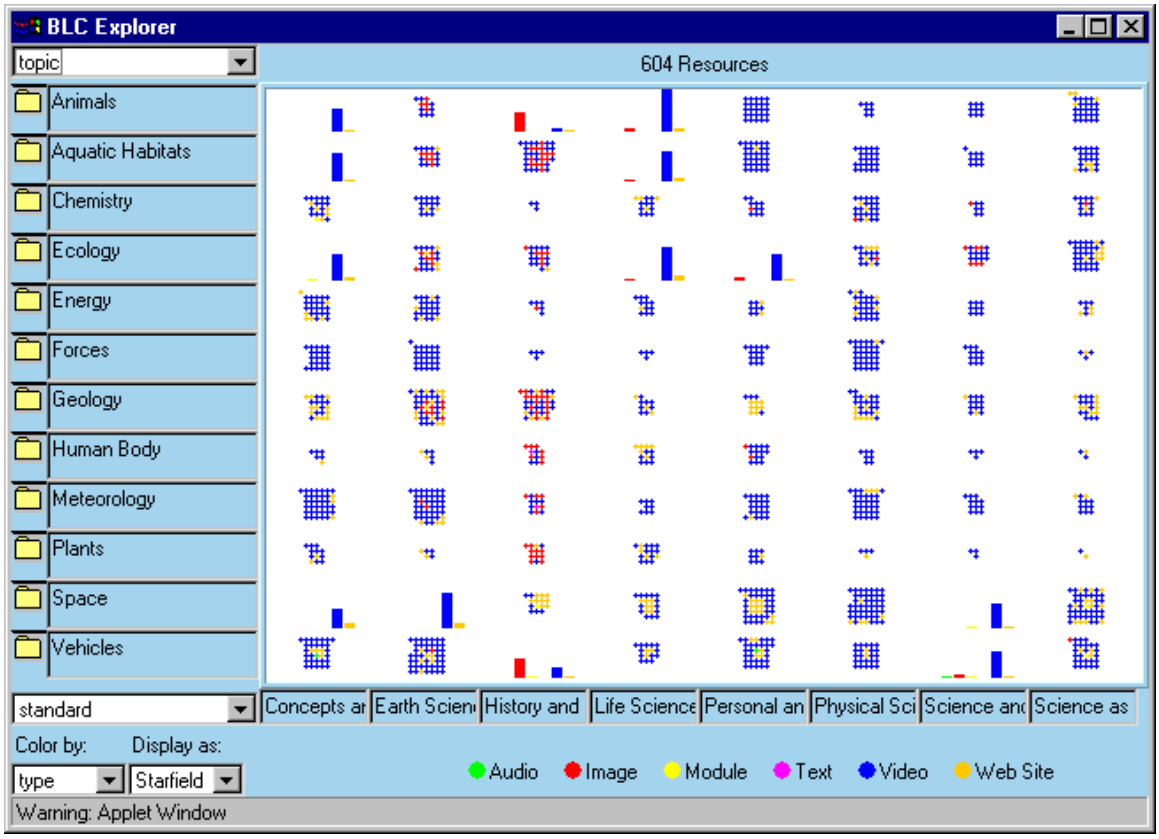

Figure 1: Early version of GRIDL, displaying a single hierarchical y-axis at the top level

items or entire digital libraries. We describe our initial work on a digital video library for middle school teachers, a legal information system, and a computer science library. Then we discuss design alternatives for hieraxes and report on a usability study. We call our interface GRIDL, a Graphical Interface for Digital Libraries, to convey its grid-like design.

PREVIOUS RESEARCH

Two-dimensional displays with meaningful axes have been used successfully to show document spaces, such as in Envision (Fox, Hix, Nowell, Brueni, Wake, Heath, Lenwood, and Rao, 1993; Nowell, France, Hix, Heath, Lenwood, and Fox, 1996) and FilmFinder (Ahlberg and Shneiderman, 1993). FilmFinder's x-axis was fixed as the year of a film's production, and its y-axis was the film's popularity (number of video rentals, on a scale of 1-9). Each spot on the graph was color-coded by film type (drama, action, musical, etc.). The commercial expansion of this tool (Ahlberg and Wistrand, 1995), Spotfire (http://www.spotfire.com) allows any spreadsheet data set to be visualized. Spotfire allows both continuous and categorical variables to be put on either axis or used for color and size coding (continuous variables are color-coded using a spectrum, categorical variables with a set of discrete colors). Shape and rotation are available options, but these codings are less easily perceived. Changes to the axes or codings can be made in a few seconds.

Our experience and Spotfire's commercial success shows that it is powerful, but that some training is needed. Moreover, some users find the rich display and extensive controls (including control panels, sliders, and zooming) confusing. Additionally, the potential clutter of color-coded, size-coded, overlapping spots requires users to plan their visualizations carefully. Labels and lines connecting related items add further challenges for novice users.

Tree structures are commonly presented as indented outlines that can be scrolled and opened or closed to expose or hide levels – as in Microsoft's Windows Explorer, Allen's "Dewey GUI" (Allen, 1995a) or WebTOC (Nation et al., 1997). Trees represented as node-link diagrams are useful, but as they grow to include thousands of nodes and many levels, layout and navigation problems become serious. Treemaps ensure a fixed-size presentation, but are complex, and many users require 15-20 minutes of training (Johnson and Shneiderman, 1991). Allen (1995b) described two "facet space" browsers for exploring multi-hierarchical (multi-dimensionally hierarchical) data sets. However, these tools did not allow users to view their data along categorical axes. On an axis, a hierarchy must be represented one-dimensionally, so we chose the metaphor of opening and closing folders to move up and down the hierarchy, displaying one level at a time.

THE BALTIMORE LEARNING COMMUNITY

The need for hieraxes arose in designing user interfaces for the Baltimore Learning Community. This digital library consists of approximately 2000 educational resources, including digital video, websites, and images, for a community of thirty middle school teachers. In addition to several categorical and continuous fields, each resource was also indexed according to a three-level curriculum hierarchy (subject/topic/subtopic) to make searching easier. Teachers needed to be able to find resources of interest quickly, without the burden of complex controls. In response to usability testing with 14 teachers (Rose, Ding, Marchionini, Beale Jr., and Nolet, 1998) we created the first version of GRIDL (then called Dotfire, since it was a variant of Spotfire) that supports limited hierarchies (Figure 1).

GRIDL uses categorical variables to organize search results on a two-dimensional grid. Categorical axis variables reduce the number of XY regions and increase overlap (i.e., many data points are mapped to the same XY region). In GRIDL, displaying each region as a cluster of dots (one for each data point in the region) solves the overlap problem. If there are more than 49 items in a region, a bar chart showing the frequency of each type of object replaces the cluster. Users can click on a cluster or bar chart to reveal a list of resource titles. Selecting an item in this list provides a more detailed description. GRIDL supports basic hierarchical axes. In Figure 1, the y-axis shows science topics (the second level of the curriculum hierarchy). Users can refine their searches by clicking on the folder icon to the left of an axis value, which expands the selected category and filters the display. For example, clicking on the "Animals" folder changes the y-axis to display only Animal subtopics (Birds, Mammals, etc.) and filters the result set so only Animal resources are displayed (Figure 2). Users can click on the "up" icon at the top of the axis to return to the science topic level.

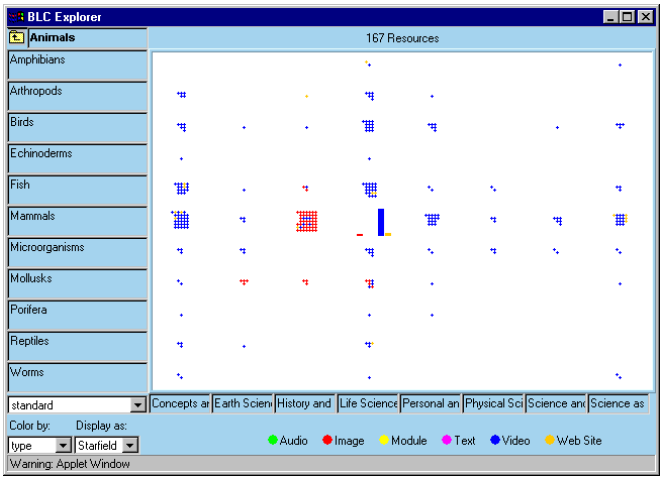

Figure 2: Expanding the "Animals" Category on the y-axis in the early version of GRIDL

However, the early version of GRIDL only handled the three-level curriculum hierarchy, and does not generalize to other hierarchies. It displays the immediate supercategory of the user's current location, but not the complete browse history. And it does not handle axis variables with large value sets (more than 15). Even with a relatively small set, all the values are added to the axis, thereby forcing the potential truncation of long names (Figure 3).

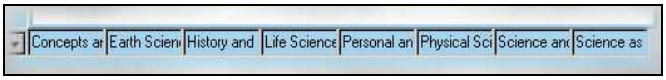

Figure 3: An Overcrowded Axis

LEGAL INFORMATION DIGITAL LIBRARY

Our interest in hieraxes grew in conducting research for WestGroup Publishers. Their digital library provides access to 12 million online cases, numerous statutes, and other valuable resources for the legal community. However, searching this digital library requires careful thought; a search result set could easily include tens of thousands of items, and users have pressing demands for comprehensiveness in their legal research. Finding a few relevant cases is rarely sufficient. Many search situations require review of every potentially relevant case and an understanding of the range of existing precedents.

WestGroup maintains an elaborate hierarchy of terms called the Key Number system. As this hierarchy is used as a topical index of cases and headnotes, its inclusion in our legal research system was crucial. The first three levels (473 items) are meaningful and familiar to many lawyers, but still complex. The entire 85,000-item hierarchy is many more levels deep, extremely complex, and specialists become familiar only with branches relevant to their work. A portion of the hierarchy is displayed below with the total number of branches for each item shown in parentheses:

- Persons (5)
- Natural Persons (16)
  - …
  - Drugs and Narcotics
  - Drugs and Druggists in General (31)
  - Narcotics and Dangerous Drugs (145)
    - …
    - Search under warrant
    - In general
    - Issuance, requisites, and validity of warrants in general (2)
    - Affidavits, complaints, and evidence for issuance of warrants (5)
      - In general
      - Probable cause in general
      - Error or ambiguity; omissions
      - Informants
      - In general
      - Personal knowledge
      - Reliability; corroboration
      - Confidential/unnamed informants
      - …
    - Execution and return of warrants in general (5)

        - Disposition of property
        - …
    - Food
    - …
- Property (4)
- Contracts (4)
- Torts (0 at level 2,16 directly at level 3)
- Crimes (0 at level 2, 56 directly at level 3)
- Remedies (13)
- Government (6)

Both depth and breadth are variable: a particular item may have as few as 2 branches or over 100, and the tree is as shallow as 2 levels in some places and as deep as 7 in others. Items are not always ordered alphabetically or numerically.

Other hierarchies relevant to legal research are geographical region and court district. In addition, some ordinal non-hierarchical fields (such as dates, relevance, numbers of citations, and names) can have hierarchies imposed on them to improve ease of use (see Imposed Hierarchies below).

A scenario for a legal information seeker might deal with a case on "stolen furniture". The search might produce a 738 items in the result set. Displaying these on a grid with year along the x-axis and Key Number along the y-axis reveals that most of the cases found were under Criminal Law (Figure 4a). Another way to look at the result set would be to put the years on the y-axis and then move down the hierarchy on the x-axis (Figure 4b).

In trying to use GRIDL with this data set, we realized that more comprehensive hierarchy support would be necessary, and began development on the general purpose hieraxis widget. We made some modifications to GRIDL itself, and then wrote the hieraxis widgets in Java, including an API to allow them to work with diverse applications.

HIERARCHICAL AXES: HIERAXES

In response to the challenge of serving legal information searchers, we developed a general purpose hierarchical axis widget for GRIDL and other applications, termed a hieraxis. Our goal was to support hieraxes on both x and y axes in a convenient manner that preserved screen space for viewing result sets, and to provide full-featured widgets for applications which meet the following criteria:

- Includes a two-dimensional visualization for which it is desirable that the user be able to change axis variables
- Data must include some categorical fields
- Data set must be large enough to necessitate some sort of management strategy

Categorical Zooming

Visualizations have been tried with both continuous and categorical/discrete axis variables. Though zooming on one or both axes has been a familiar concept for some time with continuous variables, it has not been used with categorical variables. Partly this is because "zooming" generally applies to continuous spaces. However, drilling down into a hierarchy is, in essence, categorical zooming. Such a process is already familiar to users in browse tools such as Yahoo, but not in conjunction with visualizations. We applied hieraxes to a two-dimensional visualization, explored design alternatives, and conducted a usability test (Figure 4a and b show the legal information system Key Topics and Figure 4c and 4d show the ACM Computing Classification System).

Browse History: Maintaining Context

As mentioned above, one shortcoming of GRIDL's original hierarchy support was the potential to get lost in a confusing hierarchy (Chimera and Shneiderman, 1994). To alleviate this, we added a simple browse history to the hieraxes. For each axis, users see their current location in the hierarchy in a drop-down list and as an indented series of labeled folder

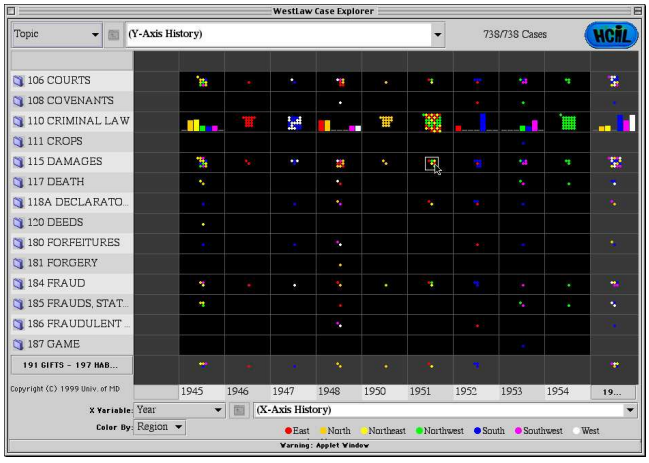

Figure 4a: GRIDL with hieraxis widgets for legal information, showing hierarchical data on the y-axis

Figure 4b: GRIDL showing legal information after two category expansions on the x-axis, 110 Criminal Law and 110XX(B) Course and Conduct of Trial in General

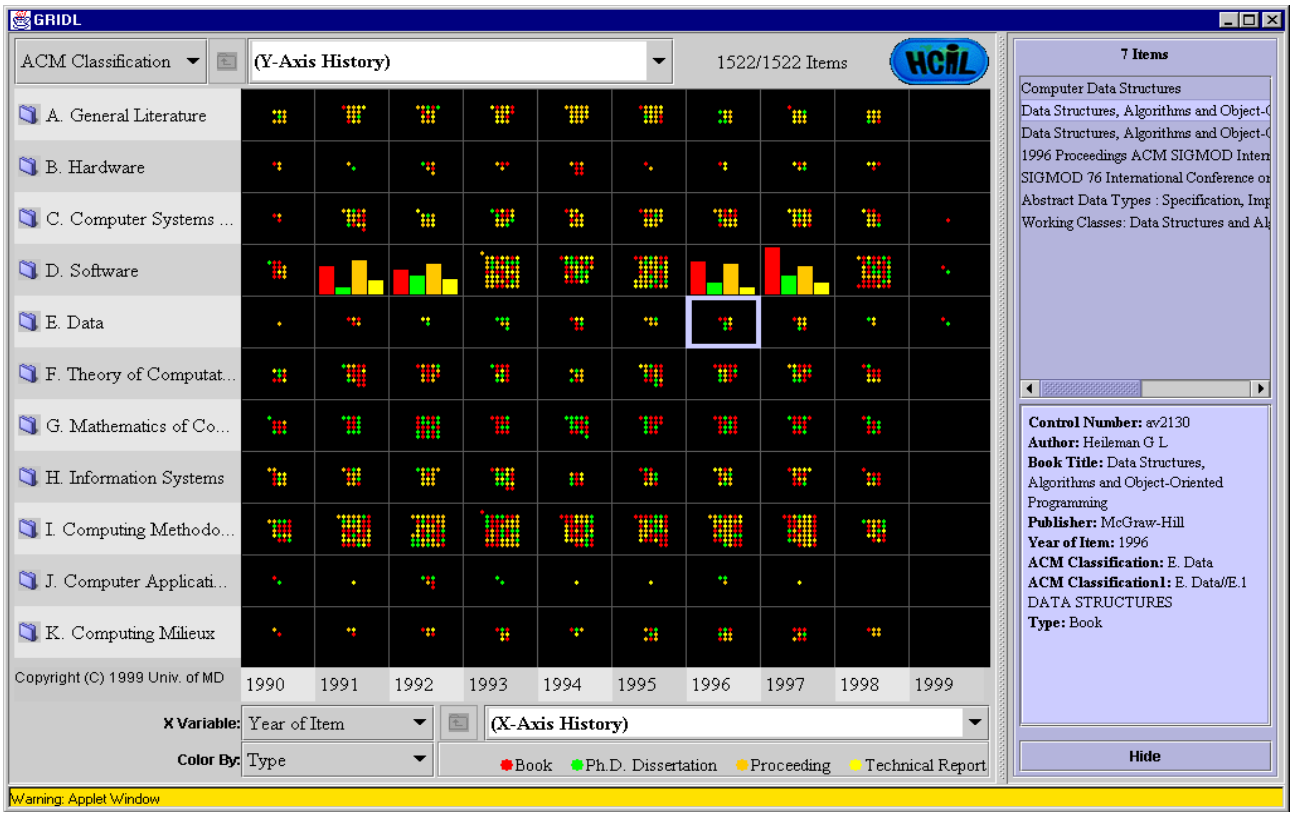

Figure 4c: GRIDL with the ACM Computing Classification System, after user clicks on a cluster

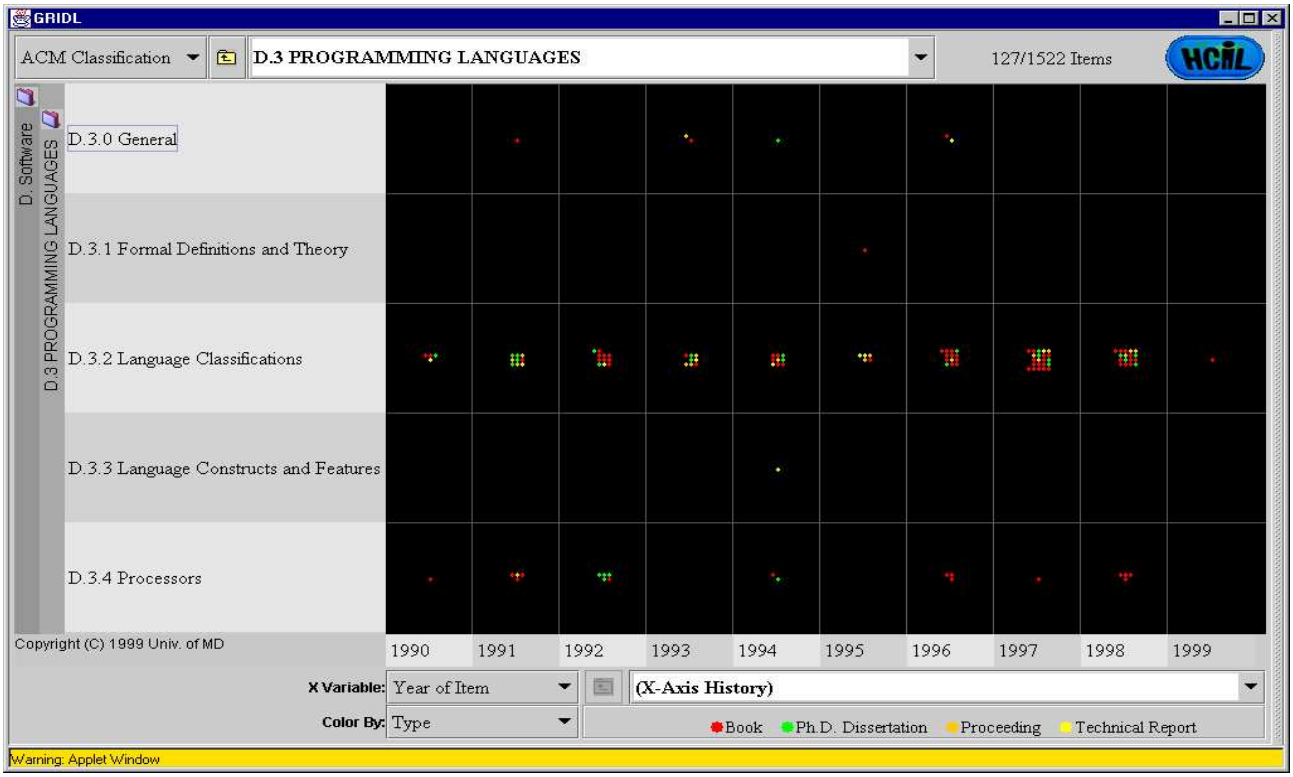

Figure 4d: GRIDL with the ACM Computing Classification System after two category expansions on the x-axis, D. Software and D.3 Programming Languages

icons. Users can zoom out to any supercategory of their current location by selecting from the drop-down list or clicking on a folder (see Figure 4). Users can also zoom out one level by clicking on a standard "Up Level" icon near the browse history drop-down.

Any feature that allows users to view a subset of the entire visualization risks confusing them, causing users to lose their place. The browse history can help users remain aware of their location in the data set despite the fact that they may not be viewing the entire data set. By starting at the top, making all data exclusions user-driven, and providing an easy-to-use history we hope to reduce disorientation.

Hieraxes and Filtering

Some visualization tools have dynamic axes, for which the variable can be changed rapidly. With hieraxes, application developers can employ a powerful data filtering mechanism that allows a visualization to serve simultaneously as a search tool and results display.

If users filter such a display using categorical zooming, then select a new axis variable, the new variable's values will be displayed, but the displayed data will not increase; whatever data has been excluded by previous operations will continue to be excluded. Through successive category expansions (exclusions by zooming) and re-categorizations (changes of axis variable), users can graphically locate the information they desire. In this way we can continue to blur the unnecessary line between query and results. However, this approach might eventually need to be coupled with a more sophisticated search history, since the user's actions can no longer be represented accurately as a linear series of category expansions.

GRIDL does not employ this filtering mechanism, as such functionality is somewhat outside its scope: if users select a new axis variable, the axis returns to the entire data set.

Overcrowded Axes (Broad Hierarchies)

Often, categorical axis data is extensive: with hundreds or even thousands of discrete items. If the data is large enough, no labeling scheme can possibly enumerate all data items or distinguish among regions of the visualization (see Figure 3). We considered several solutions and implemented some of them (Figure 5).

Tool Tips (Figure 5a). For slightly crowded axes, a solution akin to Microsoft's tool tips is effective. Each axis label can be truncated to fit in the available space; moving the mouse pointer over a label reveals its full text, temporarily obscuring adjacent labels. However, if space becomes even more limited and no more than a few letters of a label can be displayed, users will not have adequate information to determine which labels interest them. If the axis becomes crowded enough, space may not be available for even a single letter.

A y-axis variant on the tool tip approach involves showing all tool tips at once when users mouse over the axis, or temporarily expanding the axis width to accommodate the labels. GRIDL implements standard tool tips on the x-axis, and temporary axis expansion on the y-axis.

Additional solutions exist when axes are only slightly overcrowded. Angling the axis labels, labeling in two or three staggered rows, and similar strategies can increase the number of readable labels significantly. But, as with tool tips, these techniques are ineffective when the number of items becomes too large, and users are once again faced with overcrowding.

Imposed Hierarchies (Figure 5b). There may be an appropriate hierarchical grouping – alphabetical, chronological, numerical, etc. – that can be imposed on the data before transfer to an axis, creating a hierarchy on the fly. If the grouping can be done in a meaningful manner, the groups will be comprehensible to users, who can then expand a group as they would a category in a semantic hierarchy. In particular, categorical but ordinal variables work well with this method: fields like date, age, rank, etc. which can easily be ordered, and therefore grouped, in a meaningful way (see Figure 5b).

Paging (Figure 5c). Another possibility is *paging*. If a display has room for 16 columns and the axis field has 80 items, the first 15 columns can be displayed as usual, with the 16th representing the remaining 65 items. The user will be able to see from the visualization how much data is contained in those 65 columns, and can move to the next "page" of data by clicking on a folder or arrow icon next to the 16th item in the x-axis. The distinction between the first 15 columns and the 16th column can be further highlighted by a different background color. If users are on a page other than the first or last, the first and last columns will contain the contents of all previous and remaining pages, respectively. This strategy has the advantage of maintaining a display of all data at all times, but a potential disadvantage of user disorientation due to too much paging. GRIDL implements paging.

Scroll/Zoom (Figure 5d). A fourth strategy is a combination scroll/zoom technique, using range (double-box) sliders as Spotfire does. All items are placed on the axis, but compressed together with labels minimized or removed when the entire display is visible. Users can zoom in on an area of the axis by decreasing the range of the slider and scroll along the axis by sliding the slider. At any time they can easily return to the full visualization by increasing the slider's range to its maximum. This strategy minimizes disorientation by maintaining users' connection with the full visualization and tightly linking the potentially confusing scrolling with zooming. However, with somewhat smaller data sets (16-100 items) this method is awkward as both scrolling and zooming become "chunky" and confusing.

Implementation Strategy. Our primary design for handling overcrowded axes is paging, supplemented by tool tips/axis expansion. However, paging through

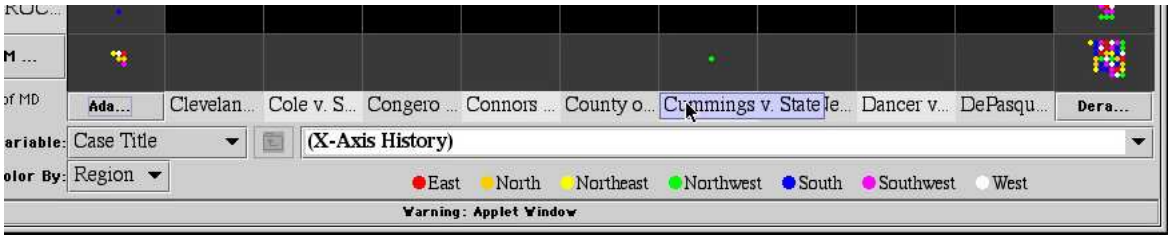

Figure 5a: Managing Overcrowded Axes: Tool Tips

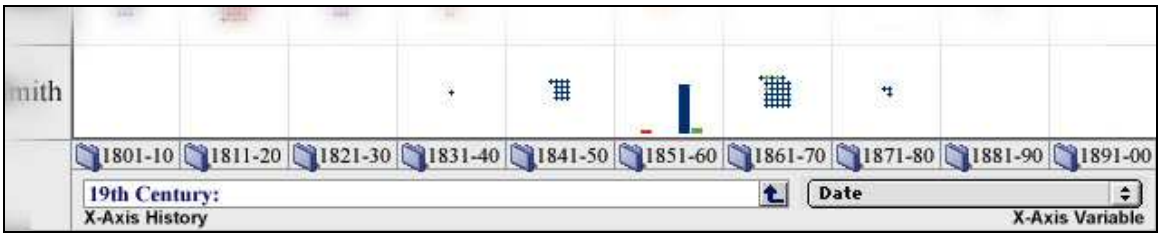

Figure 5b: Managing Overcrowded Axes: Imposed Hierarchies (Sketch)

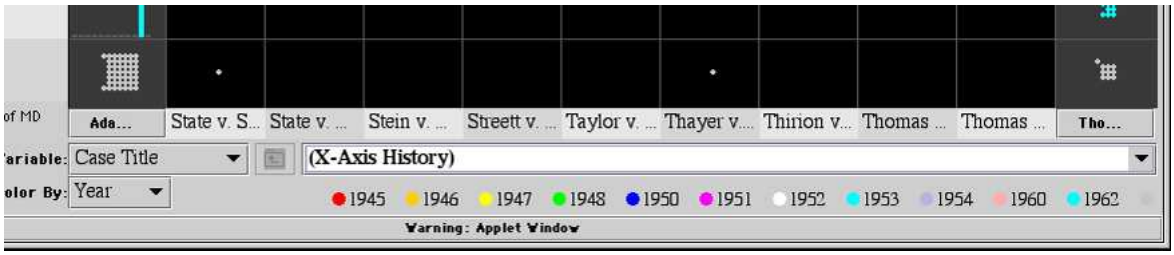

Figure 5c: Managing Overcrowded Axes: Paging

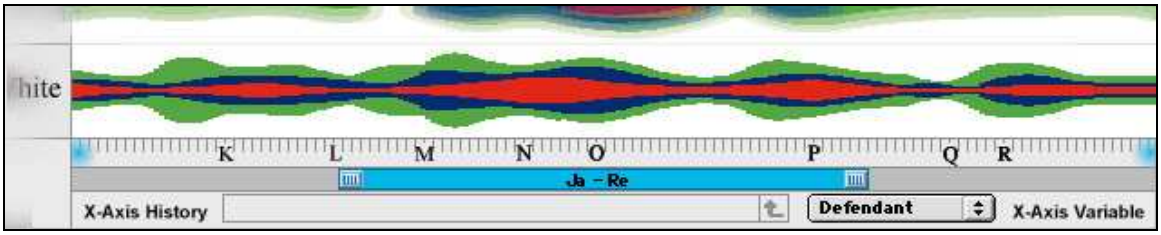

Figure 5d: Managing Overcrowded Axes: Scroll/Zoom (Sketch)

thousands of items is tedious. The other two strategies seem more compelling when the number of items is greater than a few hundred. Imposed hierarchies and scroll/zoom can speed the process of locating a known item (see Figure 5). Users could choose among imposed hierarchies, paging, and scroll/zoom by selecting from a pull-down menu.

ACM COMPUTING CLASSIFICATION SYSTEM

Searching for articles related to a specific computing field can yield hundreds or thousands of records in big digital libraries, such as the ACM Digital Library, IEEE Computer Society Digital Library or INSPEC. GRIDL may be useful in this domain, by showing the data distributed according to the ACM Computing Classification System. This classification system contains 345 categories organized in a 3-level hierarchy, and it is widely known in the field.

The 11 first level links were easily accommodated, and all second and third levels can fit in the axis without paging.

We used GRIDL with bibliographic data taken from the database of the Computer Science Department Library at the University of Maryland. The library uses the ACM Computing Classification System for keyword searches.

The general-purpose design of the hieraxis widget allowed us to make the adaptation to this new domain in a short period of time, and with a low programming effort. Some new features were added, such as showing the database entry for the selected item in the description panel (lower right), and providing visual highlights about how data shown in the grid and the description panel are related (see Figure 4c).

USABILITY TEST

We tested GRIDL for the ACM Computing Classification System in two phases.

In the first set of tests we had 8 subjects, all of them computer science graduate students. After getting a 2-minute introduction to GRIDL, they were asked to perform 8 simple tasks using hieraxes.

Two tasks implied searching for a specific cluster in the grid, while the other 6 tasks implied comparison of clusters. Three tasks required distinguishing different color-coded items. Six tasks implied measuring the number of items in clusters.

Overall, test participants showed a good understanding of the concept of categorical zooming, and they were able to easily navigate through the hierarchy to find information. 7 participants accomplished 6 of the 8 tasks, and the remaining participant accomplished 5 of the 8 tasks.

Two participants suggested that more status information would be desirable, in order to reduce user disorientation. Two participants said it was helpful to get the number of items in a cell by clicking on them, they suggested using tool tips to give this information when moving the pointer to a cell. Two participants found temporary y-axis expansion annoying. Other improvements suggested by participants are:

- To make bar graph scale explicit.
- To provide more status information about the current position in the hierarchy.
- To order dots in a cell in a meaningful way.

We added vertical text labels with the categories (see figure 4d). This improvement was aimed at reducing user disorientation about their current position in the hierarchy.

We carried out a second set of tests with 24 different participants with greater diversity in their educational background: 10 participants had a library science background, 8 had a of computer science background, and 6 had other backgrounds.

After the 2-minute introduction, they were asked to perform 6 simple tasks, and to provide comments and suggestions about the tool. Two tasks required searching for a specific cluster in the grid, while the other 4 tasks required comparison of clusters and counting the number of items in clusters.

Seventy-one percent of the participants were able to complete the 6 tasks successfully, 8% completed 5 tasks, 13% completed 4 tasks, 4% completed 3 tasks, and 4% completed just 2 tasks.

The two main problems found by the participants were:

- "Clickability" of categories: 89% of participants clicked at least once on a category that could not be further expanded.
- Temporary-axis expansion: 49% of participants found it disturbing or annoying.

Participants with a library science background provided the following comments:

- 70% of them showed a high appreciation for the tool when asked about their impression.
- 50% of them compared the tool with other systems, stating that the kind of view that GRIDL offers cannot be obtained in other systems for exploring the contents of a library.

CONCLUSIONS AND FUTURE DIRECTIONS

Hieraxes, in combination with a grid display, offer a simple approach to browsing search result sets. Users can see an overview by color-coded dots or bar charts arranged in a grid and organized by familiar labeled categories. They can probe further by zooming in on desired categories or switching to another hierarchical variable. Our early success with middle school teachers has encouraged our development of a general-purpose tool. GRIDL was refined by applying it to the legal information Key Topics and the ACM Computing Classification System. Usability tests gave us insights for further developments and the encouragement to consider using GRIDL to visualize a

complete library.

Although the basic idea of hieraxes is appealing to many users, not every user is visually oriented. For example, some legal researchers will prefer to see 20 titles of cases at a time, rather than 2000 color-coded dots. For digital libraries that do not have existing hierarchical categorizations, the imposed hierarchies may provide some benefit, but paging through long lists of authors or other unique (or relatively unique variables) could be tedious. Rapid visual exploration may be useful for some users and tasks, but it might undermine the willingness of users to reconsider their initial search terms. These and the following issues will be examined in future work:

Making Sure the User doesn't Get Lost

Though much of our effort has focused on avoiding user disorientation, it may continue to be a problem. Prominent indications of the user's location, carefully chosen icons, and animation may help with this, as well as a more extensive and intuitive search/browse history.

Multiple-Category Expansion

In many cases, users may want to expand several (adjacent) categories at once. This can be accomplished by allowing users to band-select several categories at once, expanding them onto the axis together. This could prove especially useful for ordinal axis fields and well-constructed hierarchies in which proximity is meaningful.

Category Expansion through Direct User-Visualization Interaction

Forcing the user to click on the axes in order to expand the visualization is comprehensible but a more direct alternative is to allow users to select portions of the visualization itself to expand, on one or both axes (depending on the shape of the selection).

Three-Dimensional Hieraxes

Our current work extends the early GRIDL functionality by including robust x- and y-hieraxes, with support for greater depth and breadth, allowing users to visualize complex, multi-hierarchical data sets. However, addition of a third dimension, through an additional hieraxis or another method, might yield an even more powerful tool.

ACKNOWLEDGMENTS

We appreciate the support of WestGroup Publishers and the U.S. Dept of Education Technology Grant (#R303A50051) to the Baltimore City Public Schools. We thank Robert B. Allen for his thoughtful comments on a draft of this paper.

REFERENCES

Ahlberg, Christopher and Shneiderman, Ben, Visual information seeking: Tight coupling of dynamic query filters with starfield displays, *Proc. CHI'94 Conference: Human Factors in Computing Systems*, ACM, New York, (1994), 313-321 + color plates.

Ahlberg, Christopher and Wistrand, Erik, IVEE: An information visualization & exploration environment, *Proc. IEEE Information Visualization '95,* IEEE Computer Press, Los Alamitos, CA (1995), 66-73.

Allen, Robert B., Two digital library interfaces that exploit hierarchical structure, *Proc. DAGS95: Electronic Publishing and the Information Superhighway* (1995).

Allen, R. B., Retrieval from facet spaces, *Electronic Publishing*, (1995), 247-257, 8.

Card, Stuart, Mackinlay, Jock, and Shneiderman, Ben, *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann Publ., San Francisco (1999).

Chalmers, M. and Chitson, P, BEAD: Exploration in information visualization, *Proc. ACM SIGIR Conference*, ACM, New York (1992), 330-337.

Chimera, R. and Shneiderman, B., An exploratory evaluation of three interfaces for browsing large hierarchical tables of contents, *ACM Transactions on Information Systems 12*, 4 (October 1994), 383-406.

Fox, Edward A., Hix, Deborah, Nowell, Lucy T., Brueni, Dennis J., Wake, William C., Heath, S. Lenwood, and Rao, Durgesh, Users, user interfaces, and objects: Envision, a digital library, *Journal of the American Society for Information Science 44*, 8 (1993), 480-491.

Hearst, Marti and Karadi, Chandhu, Cat-a-Cone: An interactive interface for specifying searchers and viewing retrieval results using a large category hierarchy, *Proc. 20th Annual International ACM SIGIR Conference*, ACM, New York (1997), 246 - 255.

Johnson, Brian, and Shneiderman, Ben, Tree-maps: A space-filling approach to the visualization of hierarchical information structures, *Proc. IEEE Visualization'91*, IEEE, Piscataway, NJ (1991), 284–291.

Kim, H. and Korfhage, R., BIRD: Browsing Interface for the Retrieval of Documents, *Proc. of IEEE Symposium on Visual Language'94,* IEEE (1994), 176-177.

Nation, David A., Plaisant, Catherine, Marchionini, Gary, and Komlodi, Anita, Visualizing websites using a hierarchical table of contents browser: WebTOC, *Proc. 3rd Conference on Human Factors and the Web*, (June 1997). http://www.uswest.com/web-conference/proceedings/nation.html

Nowell, Lucy Terry, France, Robert K., Hix, Deborah, Heath, Lenwood S., and Fox, Edward A. Visualizing search results: Some alternatives to query-document similarity, *Proc. 19th Annual International ACM SIGIR Conference,* ACM, New York (August 1996), 67-75.

Pratt, Wanda, Dynamic organization of search results using the UMLS, *Proc. American Medical Informatics Association Symposium* (1997).

Rose, A., Ding, W., Marchionini, G., Beale Jr., J., Nolet, V., Building an electronic learning community: From design to implementation, *Proceedings of CHI'98: Human Factors in Computing Systems*, ACM, New York (1998), 203-210.

Shneiderman, Ben, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison Wesley Longman Publishers, Reading, MA (1998), Chapter 15.

Wise, James A., Thomas, James, J., Pennock, Kelly, Lantrip, David, Pottier, Marc, Schur, Anne, and Crow, Vern, Visualizing the non-visual: Spatial analysis and interaction with information from text documents, *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 51-58.