

NetLens: Iterative Exploration of Content-Actor Network Data

Hyunmo Kang, Catherine Plaisant, Bongshin Lee*, Benjamin B. Bederson*
Human-Computer Interaction Laboratory
Computer Science Department*
University of Maryland Institute for Advanced Computer Studies
College Park, MD 20742
{kang, plaisant, bongshin, bederson}@cs.umd.edu

ABSTRACT

Networks have remained a challenge for information retrieval and visualization because of the rich set of tasks that users want to accomplish. This paper offers an abstract Content-Actor network data model, a classification of tasks, and a tool to support them. The NetLens interface was designed around the abstract Content-Actor network data model to allow users to pose a series of elementary queries and iteratively refine visual overviews and sorted lists. This enables the support of complex queries that are traditionally hard to specify. NetLens is general and scalable in that it applies to any dataset that can be represented with our abstract data model. This paper describes NetLens applying a subset of the ACM Digital Library consisting of about 4,000 papers from the CHI conference written by about 6,000 authors. In addition, we are now working on a collection of half a million emails, and a legal cases dataset.

CR categories: H.5.2 [Information Interfaces and Presentations]: User Interfaces – graphical user interfaces (GUI), interaction styles, prototyping.

Keywords: Human-Computer Interaction, information visualization, network visualization, content-actor network data, iterative query refinement, incremental data exploration, user interfaces, digital library, piccolo.

1. INTRODUCTION

Most visualization research on understanding relationships in large datasets implicitly assumes that a node link diagram is appropriate. In fact, nearly all the submissions to the InfoVis 2004 contest - that required solutions to the analysis of 10 years of InfoVis publications and their authors [12] – used displays of node-link diagrams. We have produced many such visual displays ourselves but we believe that while node-link diagrams have their place, they don't scale up well and too often produce cluttered overviews with few readable labels, and often have difficulties supporting even the simplest tasks such as reviewing the papers that cite a selected paper.

In this paper, we took a completely different approach in designing NetLens, which was inspired by our submission to that contest (PaperLens[15]), by using multiple simple coordinated views of ordered lists and histogram overviews. NetLens represents a general and scalable design that uses a “Content-Actor” model of information. Examples of Content-Actor pairs of interest to the visual analytics community include scientific publications and authors, emails and people, legal cases and courts, intelligence reports and countries or groups, products and companies. In all those examples, both the content and actors consist of networked data such as reports citing other reports,

authors having advisors or co-authors, products replacing or integrating other products, etc.

NetLens shows paired networks of content and actors in coordinated views. It supports complex queries that are traditionally difficult to specify by allowing users to pose a series of elementary queries and iteratively refine them with visual overviews and sorted lists.

After reviewing the related work, this paper first describes the NetLens interface using a subset of the ACM Digital Library. We show how users can easily determine which sub-fields are trendy and which are on the decline, find appropriate experts to review a paper or serve as an expert witness in patent litigation, or even determine relationships between groups based on an analysis of their publications. We then describe NetLens in more general terms and show how the Content-Actor data model is defined with its related tasks, and finally review the most important usability challenges of this novel approach and how we addressed them.

2. RELATED WORK

2.1 Network Data Visualization

Many interactive visualization systems have been developed to help users explore and analyze collections of networked documents or communities of people [11][9]. Several websites (e.g. [2], [27]) show a rich variety of network visualization examples. However, most of the network visualization efforts have focused on layout strategies that can be classified into node-link diagrams or matrix representations [8][10].

Visual interfaces to digital libraries show an alternative way of network visualization [6]. Butterfly [17] combines search and browsing to visualize citation graphs. Envision [19], a digital library augmented with a flexible user interface, facilitates examining very large datasets and helps users discover patterns in results of queries. GRIDL [23] visualizes thousands of search results at once on a two-dimensional display that use categorical and hierarchical axes.

2.2 Multiple Coordinated Views

Multiple coordinated views often improve user performance, enable discovery of unforeseen relationships, and offer unification of the desktop [18][3]. Spotfire [25] displays a single table in different types of views, such as scatter plots and bar charts. All the views are coordinated by brushing [4] to allow users to relate data points across views. Visage [21] provides richer data manipulation operations such as drill-down, drag, paint, and roll-up to enable users to compose a complex query, but all the views are coordinated based on a single centralized table so it can not represent the relationships across multiple tables. Snap-Together [18] enables users to create different types of coordinations such

as brushing, drill down, overview and detail view, and synchronized scrolling. Although some types of coordinations can represent joined tables, their relationships are limited only to one-to-one or one-to-many and thus are not appropriate for representing network relationships.

2.3 Iterative Query Refinement

Conventional search interfaces to online resources allow users to specify several query fields and initiate the query by pressing a button. While these systems are easy to interact with, it is often difficult to formulate or express queries effectively. Furthermore, users often learn what they need to ask and how to do it through the query process. Thus, iterative query refinement has been used in many information retrieval systems. For example, SIS [8] commits queries whenever users change any of the filtering widgets. VQuery [13] enables users to interactively refine query results, using Venn diagrams to provide dynamic query previews. PESTO [7] provides a query history mechanism enables users to reuse old queries.

3. NETLENS INTERFACE

While NetLens supports any dataset that can be represented with our abstract Content-Actor data model (see Section 4), we first illustrate the interface using a specific dataset and describing simple scenarios of use in this context. The dataset consists of a

subset of the ACM digital library and contains 4,073 papers from the CHI conference from 1982 to 2004 authored by 6,358 people. The NetLens display (Figure 1) is divided into two symmetric sections. On the left is the content – i.e. the papers for this dataset, and on the right are the actors – i.e. the authors of the papers. Each section includes several panels: overviews at the top (showing histograms which can show distributions of items over all the available attributes), filters on the right side, and lists of items in the lower area. In each section the different panels are tightly-coupled [18] so that any changes such as selection in one panel is immediately reflected in the other panels. All elements of the display are used to specify queries and to display results: for example, selecting a histogram bar filters the list of papers, and hiding certain papers from the list (e.g. the less cited ones) is reflected in the histogram.

Initially the two sections for papers and authors sections are not linked and are explored individually. The red oval at the center of the display indicates that nothing will flow in between, and users can change this and set the direction of the flow by right clicking on the red oval. When flow is allowed the oval is replaced by a directional arrow (Figure 2c).

We provide series of figures but recommend that you view the video demonstration submitted with the paper (also available at www.cs.umd.edu/hcil/netlens/VAST-video.html)

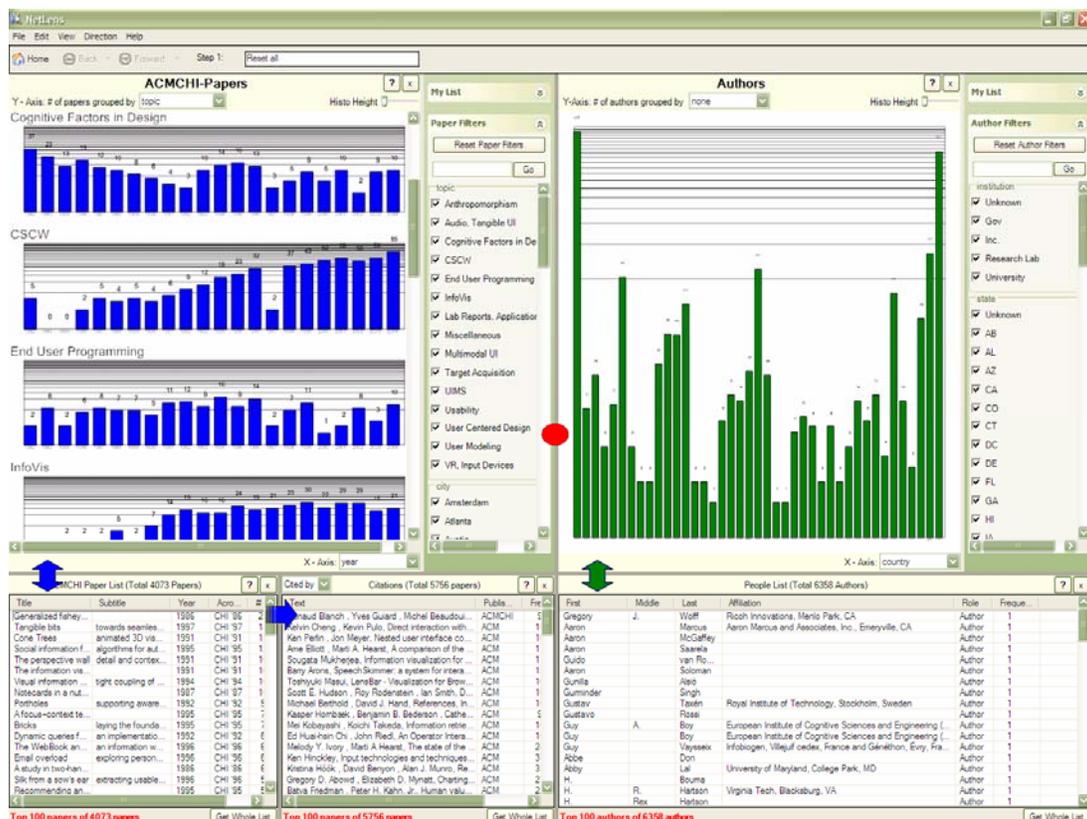


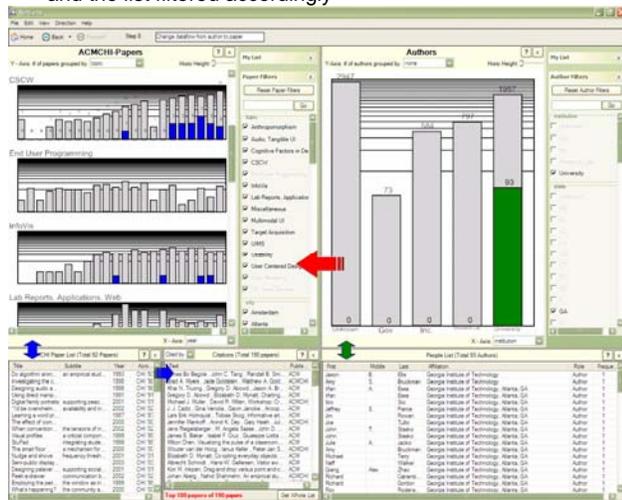
Figure 1. NetLens has two symmetric windows. The left is for Content (papers), the right for Actors (authors). Each side is further divided into panels; overview at the top, filters on the right, and lists at the bottom. Here the Content side has two lists to reflect the citation or reference relationship between papers. The paper overview panel shows the distribution of papers (in log-scale) over time, grouped by topics. Users can see which topics have seen their number of papers increase or decrease over 22 years. On the right side the overview of the authors shows the distribution of countries of origin (the scale is logarithmic)



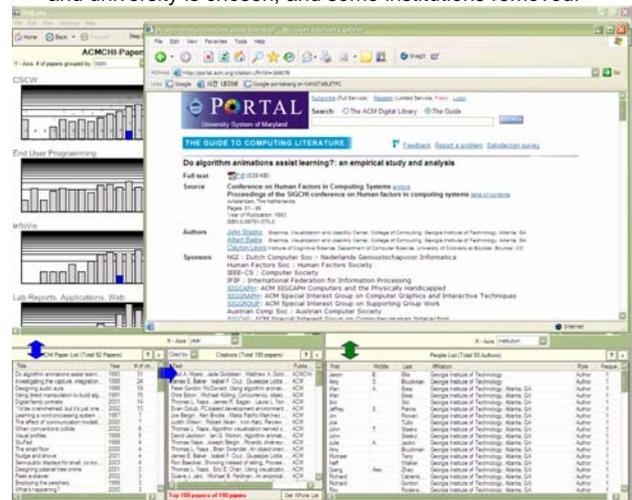
(a) In the overview of authors per state, Georgia was selected and the list filtered accordingly



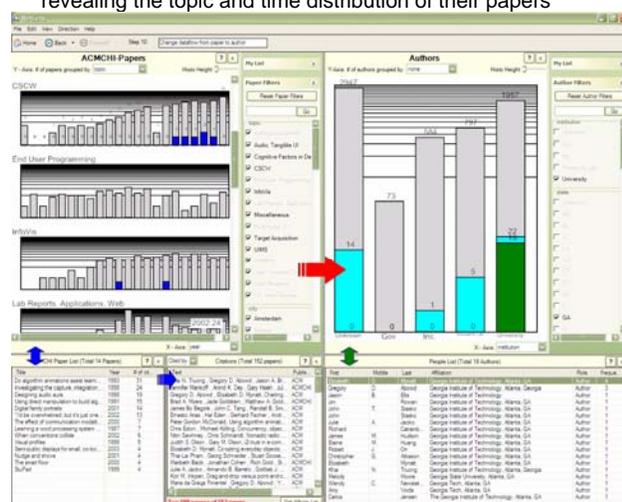
(b) the overview is changed to show distribution by institution and university is chosen, and some institutions removed.



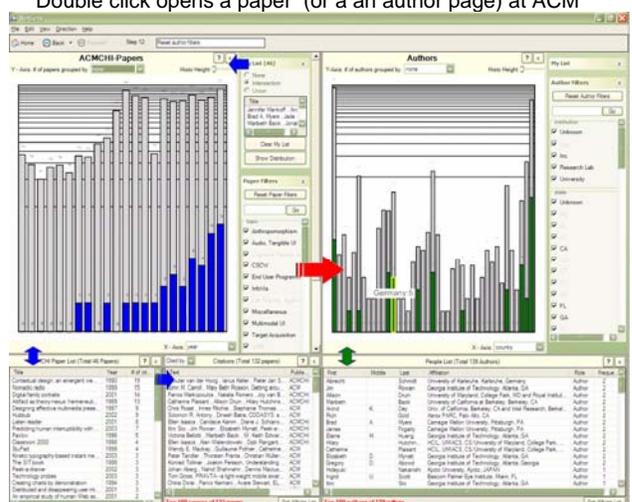
(c) The GeorgiaTech authors are sent to the paper side revealing the topic and time distribution of their papers



(d) Sorting by number of citations reveals influential papers. Double click opens a paper (or a an author page) at ACM



(e) Most influential papers were selected and sent to author side. The list of influential GT authors is sorted by number of papers.



(f) All citations of the influential papers are saved in "My list" and shown. Right side show country distribution of their authors.

Figure 2. Selected steps of the interaction of the scenario "Learn about the group of researchers"

Scenario: “Learn about a group of researchers”

Let’s imagine that John needs to learn more about a group of scientists from Georgia that is knowledgeable in HCI. He thinks of several questions such as: “What group is it?” “How active are they?”, “What topics are they working on?”, “Who leads with many influential papers?” and so on. He asks to see an overview of the authors by State by changing the attribute mapped to the X axis to “State”. He clicks on the Georgia bar and scans the list of names and affiliations (Figure 2a). Looks like Universities are dominating, so he sets the overview to show a distribution by institution and selects “Universities” which filters the list of authors to 125 authors now. He sorts and visually scans the affiliations. He can see many variants of the name Georgia Tech so that that must be the place he should be studying more closely. He removes other institutions from the list (Figure 2b).

Now he wonders what they worked on, so he sends the list of authors to the paper side by changing the red oval to a left arrow (Figure 2c). He sees that they have 92 papers. By scrolling the paper overview he sees lots of papers in CSCW and Multimodal User Interfaces. Papers are sorted by number of citations so the top of the list now shows the most influential papers. A double click on a paper access the ACM Digital Library so he can read the paper (Figure 2d). When a paper is selected, its citations are also highlighted and brought to the top of the list of citations. He can switch to references instead of citations.

Next John wonders who wrote the most influential papers. He selects the 14 most influential papers (cited more often) then changes the direction of the arrow. Only 21 authors are left (Figure 2e). He can order them by the number of (influential) paper. On the top is Beth Mynatt who was an author on 4 papers.

Finally, John decides to look at who was influenced by the work of Georgia Tech. He selects all the citations of the 14 most cited papers and save those citations in “My List” and ask to have that shown on the bars now. It updates the list of authors accordingly, showing the distribution of who cited the influential papers. John resets the author filters to see them all (instead of only the Georgia authors). He then switches to an overview by country, to see which countries have been following this work (Figure 2f).

Now that we have introduced the interface of NetLens with this scenario, we will describe the underlying data model and explain how and why we chose it in Section 4. In Section 5 we will come back the some of design challenges we encountered and how we addressed them.

4. NETLENS DATA MODEL

4.1 Data Analysis

Entity-Relationship models [24] of several common network data domains such as email archives, personal photo collections, digital libraries, etc. (Figure 3) show that network data sets are generally composed of small number of independent entities (usually two or three) and their relationships. The relationships can be further divided into *intra*-relationship (relationship within the same entities) and *inter*-relationship (relationship between the different entities).

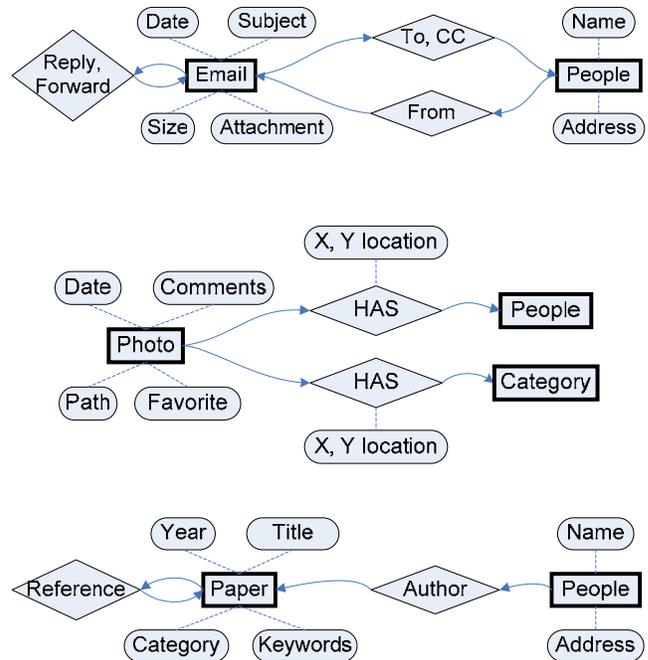


Figure 3 Entity-Relationship diagrams of three networks (email collections, photo collections, and scientific publications).

It is interesting to note that Entity-Relationship models have two properties that are useful to define an abstract network data model. First, the structure of the data model can be manipulated to suit the tasks and tools. An entity’s attributes can be separated from the entity to become a new independent entity. For example, one of the paper’s attributes, “Category”, can be separated from the paper entity and converted to a new entity so that they can have a “HAS” relationship. On the contrary, an independent entity can be merged into the related entity to represent an attribute. For example, the author entity could become an attribute of the paper entity.

Second, many network tasks can be described by following a path (an alternating sequence of entities and relationships) in the model. Suppose that you want to find out how many citations of the papers a person has written are not self-citations. You would have to search for papers written by the author, find all citations of those papers, find the authors of those citations, remove the original author, and filter the original set of citations to see only the papers written by the remaining authors. Finally, counting the papers in the resulting set yields the answer. The path for this search is People (a researcher) → Paper (written by the researcher) → Paper (citations of that paper) → People (authors of the citations) → Paper (written by those authors) → Paper (intersect two lists of papers).

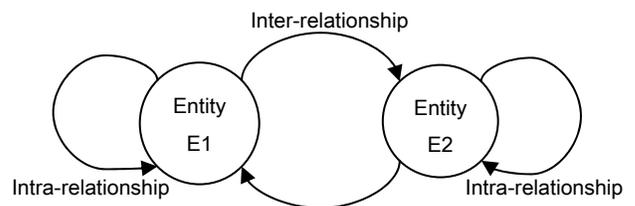


Figure 4 Abstract Content-Actors network data model

Based on these two properties of Entity-Relationship models, we designed an abstract data model as shown in Figure 4. The data model is composed of two symmetric entities, which we describe as Content and Actors respectively, and their *inter-* and *intra-*relationships. We use the terms content and actors to help in understanding and simplifying network data. While we understand that there will be some kinds of network data that are hard to map to this simple model, we are confident that it can be used as is for many practical datasets.

4.2 Task Analysis

Using the abstract Content-Actor data model as a basis, we examined the tasks that were collected from group interviews and discussions with our stakeholders as well as graph visualization user studies.

The network tasks we identified can be classified into two groups based on our abstract data model: Single-step and multi-step tasks. Single-step tasks can be defined as a query regarding the entity itself and its attributes, or one step *intra-* or *inter-*relationships. The following are examples of single-step tasks for a digital library dataset with paper and author entities:

- How many papers on “User Study” were published in 1998?
- Who are the authors of the papers on “Virtual Reality”, which were published at the CHI 99 conference?
- Which paper is most frequently cited by the papers published at the CHI 2004 conference?
- Which author is most frequently cited in the “InfoVis” topic?
- How many papers were published by UMD HCIL people?
- Who are the authors whose nationality is Korea?

Table 1 shows the classification of single-step tasks, which is based on our abstract Content-Actor data model. It shows that each entity can be used either for a search query or for a search result. In other words, papers can be searched either by papers or by people, just as authors can be searched by papers or by people. Due to the symmetric structure of the entities in our data model, the search results of a single-step task can be recursively used as an input to other single-step tasks, which makes it possible to accomplish more complex multi-step tasks by iterating over the table.

Single-step tasks can be linked together to form multi-step solutions to more complicated and high level network tasks. The following examples show some interesting high level tasks that people might ask about digital libraries.

- Evaluate individuals:
 - how many papers were self-referenced?
 - how frequently was each paper referenced by other papers?
- Identify communities:
 - what are the major paper topics published by UMD HCIL and who in this group has the most papers in that topic?
 - how do UMD HCIL’s research interests change over time and who in this group made that change?
- Find experts (to review papers or attend a workshop):
 - who wrote the most papers in the InfoVis topic and how many papers cited that author’s papers?
 - whose paper in the InfoVis area is most frequently referenced by other papers?
- Learning about a new topic (to find a good PhD topic):
 - which topic has growing publications and who contributed most to this topic in the last 3 years?
 - what are the other topics that the authors in the InfoVis area also are interested in?
- Where should I go on a sabbatical?
 - which country (or research group)’s authors most frequently reference my papers?

The above example tasks can be answered by iterating through Table 1 multiple times.

5. A GENERAL AND SCALABLE INTERFACE DESIGN

NetLens is a general interface design in that it applies to any dataset that can be represented with our abstract Content-Actors data model. This means that there must be just two entity types, each with attributes and relationships within and between those entities. For example, in conference publications, the two entity types are papers and people. Papers have attributes such as titles, abstracts and keywords. They cite and are cited by other papers and are authored by people. Those people have attributes such as institutions and fields of interest, and can be connected to papers through authorship relationships and possibly to other people through advising or committee memberships. This basic structure applies equally to collections of photos (with the primary entities being photos and objects within those photos), email collections and even legal cases.

Table 1: Sample single-step tasks (on digital libraries) classified by the combination of two entities and the way the entities are used (either search-by entities or result entities)

Search By Search Result	Entity 1 (Paper)		Entity 2 (People)	
	Entity 1’s Attributes	Entity 1 (Paper)	Entity 2’s Attributes	Entity 2 (People)
Entity 1 (Paper)	Search for papers by paper attributes such as year, keywords, title, conference, topics, etc.	Search for papers that “cite” or “are cited by” the specified papers.	Search for papers by people attributes such as author’s affiliation, institution, nationality, etc.	Search for papers written by the specified authors (either conjunctive or disjunctive).
Entity 2 (People)	Search for authors by paper attributes such as year, keywords, title, conference, topics, etc.	Search for authors of the selected papers.	Search for authors by people attributes such as author’s affiliation, institution, nationality, etc.	Search for academic advisors of the specified authors (either conjunctive or disjunctive).

An example of intra-relationship is reflected in the two lower panels of the Papers side, with the list of papers on the left and their references or citations on the right (Figure 1). NetLens uses whatever metadata (i.e., attributes) is available, creates filters and displays the results. The quality of the interface depends heavily on the richness and value of the metadata.

NetLens is also scalable because it applies to a standard relational database and the interface is built of common simple components such as histograms and lists. Together, they offer surprisingly rich support for real complex tasks. By avoiding visual overviews of the entire dataset that display a visual element for each entity instance, we avoid immediate problems of scalability.

6. INTERFACE CHALLENGES AND PROPOSED SOLUTIONS

In this section, some of the design challenges that were uncovered from our early usability testing are discussed along with the proposed solutions.

6.1 Dataflow Control

One of the most important issues in designing the NetLens interface was about how to control dataflow between and within the entity windows (*between-entity* and *within-entity* dataflow hereafter). The initial *between-entity* dataflow model was simple. We designed the entity windows to be tightly-coupled so that any data changes in one entity window were automatically transferred to the other. However, the tightly-coupled entity windows often resulted in a cyclic dataflow, and moreover, the automatic *between-entity* dataflow often caused unintended and unnecessary query refinement as well as performance degradation problems. Therefore, the *between-entity* dataflow model was revised to let users manually select the direction of dataflow (if any) by using an additional interface widget (red arrow) as shown in Figure 2c.

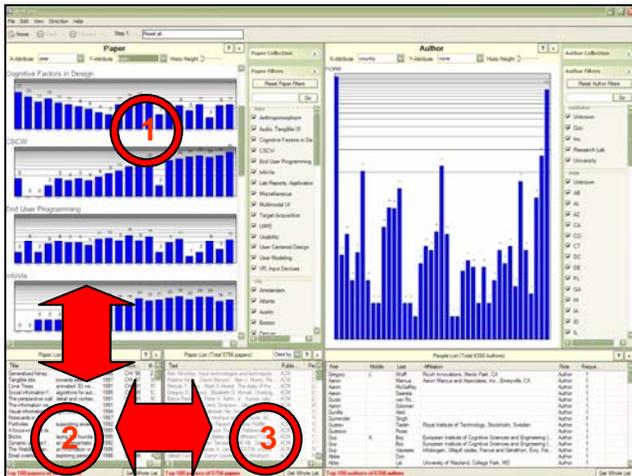


Figure 5. The *within-entity* dataflow among three sub-panels. Note that red numbered circles and arrows are not part of the actual interface.

On the other hand, all the sub-panels in the entity windows are tightly-coupled. For example, if some papers are filtered out from the overview panel in Figure 5, then the list of papers in the detail-on-demand panel will be updated (1→2), and the list of references in the intra-relationship panel will also be updated

accordingly (2→3). However, NetLens stops the *within-entity* dataflow and does not allow it to be propagated to the overview panel (3→1) in order to avoid cycles. Although the direct dataflow between 1 and 3 is not allowed in NetLens, users can make use of the “My List” feature for indirect dataflow between 1 and 3. Users can collect any entity instances and save them to “My List” while exploring data, and use this collection as a filter in the overview panel.

6.2 History and Integrated Help

Because users generally perform a sequence of interactions to accomplish a task, they sometimes get lost during the data exploration and ask themselves questions such as, “How did I get here?” or “What does the current filtered dataset mean?” To avoid such problems, we added two support mechanisms in NetLens: query history management, and integrated guidance.

The query history management system records every interaction users make in NetLens and automatically annotates each interaction to help users remember the interactions they have made before. By using the list of annotated interactions (Figure 6), users can go back and forth to confirm or correct their previous interactions. The annotations can be manually edited later if any interaction needs to be described more precisely. The history of interactions can be saved to a file and retrieved later for reuse.

The integrated guidance system helps users understand the meaning of the current dataset in each panel. Each panel has a “?” button located at the top right corner. Upon clicking that button, a yellow sticky-note-like window appears over the window to show what queries have been applied so far to the dataset in the panel.

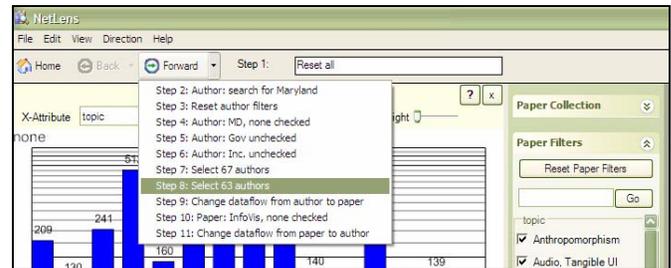


Figure 6 History management system shows the list of steps users have performed.

6.3 Multi-layered Interface

Although the NetLens interface is composed of two entity windows and six sub-panels (three panels per entity) to represent the whole abstract Content-Actor data model, users do not always need all the panels and may find the interface overwhelming, therefore we implemented a multi-layered interface [14][22] with 3 levels: a) overview only b) overview + lists, and c) overview + detail-on-demand + *intra-relationship* (e.g. references/citations). Users can also see a single entity to start.

6.4 Data Export and Integration

NetLens provides a rapid way to sift thru a lot of data and identify areas of interest. It can transfer any filtered or selected set of data to other applications through the windows clipboard (using copy and paste), internal graph class object, or even xml documents. To illustrate this capability we connected a graph browsing tool we

developed called TreePlus [16] to allow users to study the coauthorship network in more detail (Figure 7).

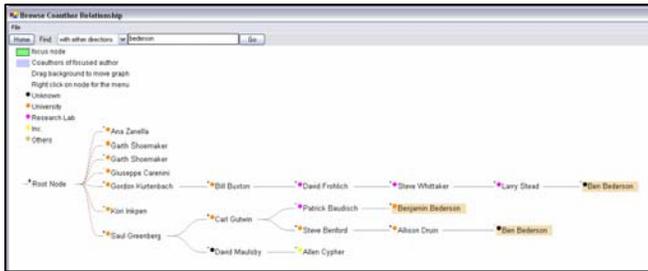


Figure 7. TreePlus shows the network of co-authorship network between Ben Bederson and Canadian authors who cite him.

7. SYSTEM ARCHITECTURE

NetLens was written in C# using the University of Maryland's open source Piccolo toolkit [5] for histogram visualization, connected to an MS Access database. One of the major benefits of using a relational database system in implementing NetLens was that it is relatively easy to construct a real data schema from our abstract Content-Actor data model. Performance and relatively small data storage size are two additional benefits we can get from the current relational database technologies.

8. FURTHER EVALUATION

Evaluating complex interfaces is a challenge, especially in the field of exploratory search [28] and Information Visualization [20]. In the case of NetLens three approaches can be taken:

- 1) Usability: evaluation could measure how usable the interface is, using classic usability measures such as speed of performance or error rates on simple imposed representative tasks.
- 2) Power: evaluating NetLens could consist of evaluating the range and complexity of queries that can be answered, and characterizing the queries that cannot be answered, then comparing the complexity of the user task compared to let's say SQL queries, either theoretically or empirically.
- 3) Generality: The evaluation can focus on the generality claim of the system by evaluating how easy it is to create new applications for new domains and data models.

We have not focused extensively on evaluation yet as we are finishing our first mature prototype but some progress has been made in all three approaches. The ACM Digital Library turns out to be a wonderful application domain to investigate because we found ourselves surrounded by potential users who are familiar with the data (and its limitations), have queries of their own, and often are mentioned in the data itself, titillating their interest and increasing their motivation to participate and provide feedback. Overall we have collected feedback from dozens of users who saw or interacted with our interface in the lab or at large government sponsored project meetings (intelligence analysts, researchers from industry and academia, software developers). They made us aware of many low level usability problems such as readability of the histogram labels, color choices, or needed improvements to the table displays. On the other hand, NetLens has always been well received. Users seem to show a sense of

relief from avoiding the traditional node link diagram displays, and greatly appreciated the simplicity of the interfaces while always acknowledging that the complex queries executed by NetLens would be very difficult if not impossible for them to specify with SQL.

An independent heuristic evaluation has just been conducted at the National Institute of Standard and Technology (NIST) in the context of the NIMD program. After video training the reviewers conducted six tasks using NetLens. The reviewers used both general HCI heuristics and visualization specific heuristics to categorize problems and issues. A report that collates the 5 individual reports will be available soon, as well as the individual reviews and video recordings of all the interactions.

To explore the generality of the NetLens design, we connected NetLens to two other datasets. Our first attempt was to explore the Enron email collection [26]. In about 3 days the first author and main developer of NetLens was able to design the new data model, produce data converters and customize NetLens to create the prototype. An estimated 200 lines of code was needed to modify the software. Most of the effort was focused on designing the data model and manipulating data. Our prototype shows emails on the left, and people on the right. An example iteration of queries is: Find the emails with a certain characteristics (e.g. rated as confidential), then find who received those emails, are there people who received significantly more emails like this, for those people, what were the characteristics of the emails they received (e.g. topic distribution), read the emails that concerned pricing issues.

A second application was developed to explore legal precedents. The database of legal cases collected by a team of researchers from the Dept of Government and Politics at the University of Maryland contains 2,780 federal judicial decisions from the period 1978 to 2005 concerning the legal issue known as "regulatory takings." Within a day, we designed the new data model, produced data converters and customized NetLens. An estimated less than a hundred of lines of code was needed. Again, most of the effort was focused on designing the data model and processing the data. The prototype shows on the left the cases, and on the right the courts that managed them.

9. CONCLUSION

Our analysis of Entity-Relationship models of common network datasets enables us to propose an abstract Content-Actor data model which we used in the design of NetLens. The NetLens interface allows users to explore network data incrementally and refine their queries iteratively so that they can accomplish complex network tasks that is usually composed of a sequence of low level tasks. NetLens is *general* in that it applies to any dataset that can be represented as a Content-Actor data model. Also, NetLens interface design is *scalable* because the interface is built of common simple components such as histograms and lists so that it can avoid visual overviews of the entire dataset that display a visual element for each entity instance.

This is just a beginning, but the utility of NetLens seems promising. Generalizing and simplifying common network datasets to the abstract Content-Actor data model and connecting them to NetLens might facilitate completion of required tasks more rapidly and reliably while offering rich support for real analytic tasks. We believe that there are great benefits in using a consistent and simple user interface to explore a variety of

content-actor datasets. Finally NetLens demonstrates that there exist powerful alternative and complementary techniques to the classic node-link diagrams to explore network data.

10. ACKNOWLEDGMENTS

We appreciate the efforts of Mary Czerwinski and George Robertson from Microsoft Research for their work on PaperLens, the motivating work for this project, and their topic analysis of CHI papers that we used in the current prototype. We also thank Doug Oard for suggesting the content actor terminology. This work was supported in part by ARDA #MDA-904-03-c-0408 Novel Intelligence from Massive Data Program, and Booz Allen Hamilton.

REFERENCES.

- [1] Amar, R., Eagan, J., and Stasko, J. Low-Level Components of Analytic Activity in Information Visualization. *InfoVis 2005, IEEE Symposium on Information Visualization* (Oct. 2005), 111-117.
- [2] An atlas of cyberspace: www.cybergeography.org/atlas
- [3] Baldonado, M.Q.W., Woodruff, A., and Kuchinsky, A. Guidelines for using multiple views in information visualization. In *Proceedings of Advanced Visual Interfaces Conference (AVI '00)*, ACM Press, 2000, 110-119.
- [4] Becker, R. and Cleveland, W. Brushing scatter plots, *Technometrics*, 29(2), 1987, 127-142.
- [5] Bederson, B.B., Grosjean, J., and Meyer, J. Toolkit Design for Interactive Structured Graphics. *IEEE Trans. Software Engineering*, 30(8), 2004, 535-546.
See also: Piccolo.Net, www.cs.umd.edu/hcil/piccolo
- [6] Börner, K. and Chen, C. Visual interfaces to digital libraries: Motivation, utilization, and socio-technical challenges. In *Visual Interfaces to Digital Libraries [JCDL 2002 Workshop]*, Springer-Verlag, 2002, 1-12.
- [7] Carey, J., Haas, L., Maganty, V., and Williams, J. PESTO: An Integrated Query/Browser for Object Databases. In *Proceedings of the International Conference on Very Large Data Bases (VLDB '96)*, Morgan Kaufmann Publishers Inc., 1996, 203-214.
- [8] Di Battista, G., Eades, P., Tamassia, R. and Tollis, I.G. *Graph Drawing: algorithms for the visualization of graphs*. Prentice-Hall, 1999.
- [9] Freeman, L. Visualizing Social Networks, *Journal of Social Structure*, 1, 2000.
- [10] Ghoniem, M., Fekete, J. D., and Castagliola, P. A Comparison of the Readability of Graphs using Node-Link and Matrix-Based Representations. *IEEE Symposium of Information Visualization 2004*, 17-24, 2004.
- [11] Herman, I., Melançon, G., and Marshall, M.S. Graph Visualization and Navigation in Information Visualization: a Survey, *IEEE Transactions on Visualization and Computer Graphics*, 6(1), 2000, 24-43.
- [12] Information Visualization 2004 contest: : The History of InfoVis:, www.cs.umd.edu/hcil/iv04contest
- [13] Jones, S. Graphical query specification and dynamic result previews for a digital library. In *Proceedings of the Symposium on User interface software and technology (UIST '98)*, ACM Press, 1998, 143-151.
- [14] Kang, H., Plaisant, C., and Shneiderman, B. New Approaches to Help Users Get Started with Visual Interfaces: Multi-Layered Interfaces and Integrated Initial Guidance *Proc. of the Digital Government Research Conference*, (March 2003), 141-146.
- [15] Lee, B., Czerwinski, M., Robertson, G.G and Bederson, B.B. Understanding Research Trends in Conferences Using PaperLens. *Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems*, (April 2005), 1969-1972.
- [16] Lee, B., Parr, C.S, Plaisant, C. and Bederson, B.B Visualizing Graphs as Trees: Plant a Seed and Watch it Grow, In *Proceedings of GD 2005*, Lecture Notes in Computer Science, Springer – Verlag.
- [17] Mackinlay, J.D., Rao, R., and Card, S.K. An organic user interface for searching citation links, In *Proceedings of the the SIGCHI Conference on Human Factors in Computing Systems*, 1995, 67-73.
- [18] North, C., and Shneiderman, B. Component-Based, User-Constructed, Multiple View Visualization, *Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems (CHI'01)*, (April 2001), 201-202.
- [19] Nowell, L.T., France, R.K., and Hix, D. Exploring Search Results with Envision. In *Extend Abstracts of the SIGCHI conference on Human factors in computing systems (CHI '97)*, ACM Press, 1997, 14-15.
- [20] Plaisant, C. The Challenge of Information Visualization Evaluation, in *Proceedings of the working conference on Advanced visual interfaces (AVI 2004)*, ACM Press, 2004 (109-116).
- [21] Roth, S.F., Lucas, P., Senn, J.A., Gomberg, C.C., Burks, M.B., Stroffolino, P.J., Kolojechick, J.A., and Dunmire, C. Visage: A User Interface Environment for Exploring Information, In *Proceedings of the Symposium on Information Visualization*, 1996, 3-12.
- [22] Shneiderman, B. Promoting Universal Usability with Multi-Layer Interface Design, *ACM Conference on Universal Usability*, ACM Press, New York (2003), 1-8.
- [23] Shneiderman, B., Feldman, D., Rose, A., and Grau, X.F. Visualizing digital library search results with categorical and hierarchical axes, In *Proceedings of the fifth ACM conference on Digital libraries*, ACM Press, 2000, 57-66.
- [24] Silberschatz, A., Korth, H. F., Sudarshan, S., *Database System Concepts*, 3rd ed., *McGraw Hill*, 1999
- [25] Spotfire: <http://www.spotfire.com>
- [26] The public Enron email dataset: www.cs.cmu.edu/~enron
- [27] Visual Complexity: www.visualcomplexity.com
- [28] White, R., Kules, B., Drucker, S., Schraefel, M. C., Supporting Exploratory Search, *Special Issues of the CACM*, April 2006 (to appear)