The Order of Things: Activity-Centred Information Access

Matthew Chalmers, Kerry Rodden & Dominique Brodbeck

Ubilab, UBS, Bahnhofstr. 45, 8021 Zürich, Switzerland

Matthew.Chalmers@ubs.com

Abstract

This paper focuses on the representation and access of Web-based information, and how to make such a representation adapt to the activities or interests of individuals within a community of users. The heterogeneous mix of information on the Web restricts the coverage of traditional indexing techniques and so limits the power of search engines. In contrast to traditional methods, and in a way that extends collaborative filtering approaches, the path model centres representation on usage histories rather than content analysis. By putting activity at the centre of representation and not the periphery, the path model concentrates on the reader not the author and the browser not the site. We describe metrics of similarity based on the path model, and their application in a URL recommender tool and in visualising sets of URLs.

Keywords: heterogeneous data, activity, indexing, collaborative filtering, information retrieval, access and visualization.

1 Introduction

As Tim Berners-Lee pointed out with regard to search engines for the World-Wide Web [Berners-Lee], "they are notorious in their ineffectiveness. [...] A Web indexer has to read a page of hypertext and try to deduce the sorts of questions for which the page might provide the answer." More generally, the indexer cannot tell in advance the context within which the page might be useful, or which pages, other than those directly linked, might be useful. We look at some earlier approaches to this problem before proposing an alternative, the path model.

1.1 Analysis of Web Access

Analysing a web site to find patterns and structures within it is one approach to establishing the context of use of a page. [Pirolli], [Pitkow97] and [Yan] looked at web sites and server logs, classified visitors, and automatically suggested new links, not in the original link topology, to related information. The authors found problems with the openness of site connectivity, for example cached links in proxies and browsers, links from search engines, and bookmarks. Not using the order of accesses beyond transitions over pairwise links was also described as a weakness. By creating new links not in the original topology, they modify the graph structure and usage patterns of the site, and thus outdate their own analyses. They have problems tracking individual users accurately without excessive invasion of privacy.

Working from the browser side is more promising. A recent example is the WBI system that "monitors the web activity of a user, classifies web pages according to the subject, searches the web for additional pages related to subjects the user has browsed, and adds links to subsequently accessed pages that suggest additional pages of interest" [Barrett]. Letizia [Lieberman] is a similar system that records the URLs chosen by a user and adaptively maintains a user profile based on word frequencies in accessed pages. Each time a user moves to a new page, Letizia searches outward from the page's contained links and looks for nearby pages that match the user profile well. The author noted that the document similarity metric Letizia employs, while commonly used in information retrieval, was relatively unreliable. Both systems are constrained by this difficult clustering and categorisation, and are limited to textual data.

A strength of both systems is their passive operation, avoiding query languages. In everyday life we are able to move through information, to understand what we hear and see, and to speak and act in the world, without requiring formal plans or descriptions of behaviour [Nardi, Suchman]. We do not, and generally can not, describe our work in coefficients and classifications valid for all situations and times: we just do it. We often, however, know what we like when we see it. This is the basis of collaborative or social filtering [Goldberg]. People collaborate to help one another perform filtering by recording their reactions to documents they read, for example actively replying to a Usenet article or simply labelling it as "good" or "bad". These annotations can be accessed by others' filters, and one can ask for information such as "documents that Smith, Jones, or O'Brien liked."

This kind of approach can be taken slightly further to allow questions such as "show me information that I have not seen before, that people with interests like me liked," for example in [Shardanand]. Personalised recommendations can be made through an iterative process of being presented with objects and rating them, leading to a profile of interest. This is then compared with the profiles of other people, and the most similar profiles may then offer good recommendations. Many groups have applied these techniques to Web access, for example those in [Resnick].

Two key aspects of this approach are highlighted here. Firstly, relevance of information is defined from the point of view of a particular set of people, rather than using a single *a priori* categorisation shared by all but not tailored to any person or situation. Collaborative filtering allows us to weave together selected individuals' categorisations as the situation demands.

Secondly, assessment of interest or value deals only with subjective ratings. This allows complex, heterogeneous data to be handled because we do not have to measure similarity based on the content of objects. Instead we indirectly express subjective aspects of meaning and use e.g. the particular people using the information, their shared body of knowledge, their accepted styles of expression, history such as reviews and discussions, and opinion of the author [Foucault]. Collaborative filtering reminds us that the symbols inside a document -- the contained words, images and links -- are not its meaning as such, but instead are part of a mesh of subtler relations and associations.

Note, however, that by building up profiles purely from discrete, manually-entered ratings of individual objects, there is no association -- temporal or otherwise -- between ratings. Consequently there is no context that might be used to tailor recommendations to the current activities of the user, a lack the path model tries to address.

1.2 Introducing the Path Model

Our earlier work has been in making layouts or maps of complex information [Chalmers96a] and presenting and interacting with such maps in 3D [Chalmers96b] and 2D [Brodbeck]. Our tools were based only on object content analysis, for example word frequency coefficients in textual documents, and classifications of financial trade data based on attributes such as currency and volume. We would often find interesting and useful patterns within information sets, but we sometimes found that the rigid formal classifications underlying similarity metrics clashed with the subtlety of information work. The 'ideal' coefficients and classifications used to produce visualisations would depend on the situation: who was present at the time, what data, plans and activities were involved, and so forth.

A turn in our work came from study of Hillier's 'space syntax', a theory of architectural and urban form [Hillier]. Space syntax puts movement and visibility at the centre of city structure and development. It uses people's paths through the city as expressions of their activities, interests, and associations. Hillier deliberately avoids a priori categorisation of structures, instead relying on building up statistics of movement and activity by the public. Analysis is in terms of the configuration of buildings and streets rather than the content or functionality of individual elements. He emphasises the importance of considering the extended paths people actually take rather than short steps between city elements.

Hillier's approach is resonant with *As We May Think* [Bush], where Bush wrote of human 'trail blazers' constructing paths through bodies of information, thus representing similarities and associations. These paths were not mere pairwise links but could span any number of objects. His notion of trail blazing as a profession reinforces the view that paths are objects of utility and value in themselves. Most work that claims roots in *As We May Think* ignores the value of paths as compound objects, instead focusing on pairwise links and the great number of possible connections one can then make. Bush understood that the value of a path is greater than the sum of its pairwise parts, as human knowledge of the significance of the entire path is involved in constructing the particular sequence of links. Here we place this point of view at the centre of information representation. We differ, however, from Bush's reliance on active construction of paths. While we do not exclude active trail blazers, we initially focus more on paths built up as a passive by-product of everyday activity.

In the path model, we treat information objects like Hillier's urban spaces. As we move through information using browsers and other information access tools, we implicitly express our interests or activities. If the names or identifiers of the information objects we move through are continually logged then we build up a *path* representing past activity. A path can be considered as a string composed of symbols from a very large alphabet (the set of object identifiers), that is gradually appended with new symbols as we move. The paths of those who move similarly to oneself are a likely source of new and interesting information to access. If one contributes to a configuration of paths, shared by a community, then one helps to build up a continually evolving set of paths and, consequently, their relative similarities.

A person may, for example, take the most recent accesses in their path -- i.e. a contiguous substring or segment at the end of the full path -- and find similar segments earlier in that path and in the paths of others. Two example similarity metrics are given in the next section. Objects that consistently arise within those segments but that we have not recently visited can be presented as recommendations, thus affording future movement and triggering adaptation through further logging, reconfiguration, and presentation. This differs from collaborative filtering, where the situation or context of a user's current activity is generally not considered when making recommendations. Here, the path offers such a context.

Paths represent users' motion through information, and this may involve more than the links inserted by page authors. By working from the browser side we address the openness of site connectivity, for example in links from search engines, caches in proxies and browsers, and URLs in mail messages and bookmarks. A path need not necessarily employ authored links at all. We see a shift from the author to the reader, or user, based on the independence of the name space of URLs from the 'official' authored links. While activity will of course be influenced by such links, it is not constrained by them.

The path model is a general approach to information representation. With this as a basis, we developed Java applications that we describe in the following sections. We describe a recommender and browser for URLs, and then a system for visualising the set of URLs that make up a path or set of paths.

2 Path-Based Recommendations

The recommender constructs paths by logging changes to a web browser's history file. Users can turn logging off and on at will, but when on, each detected access adds a timestamp and URL to a chronologically ordered log file: the path file.

The recommender runs a sliding window, of default length 10, from the beginning to the end of the path. It periodically looks for extensions to the path file, and then slides its window on further. We use a 'stop list' of regular expressions describing URLs that we wish to filter out e.g. the home pages of search engines, advert sites and, being used more for small buttons and icons, GIF files. Each window specifies a contiguous segment of the path and the set of URLs it contains. The last (most recent) path segment is then compared against all earlier segments to find a ranked list of recommendations. At present we are assessing two recommendation techniques.

In the first technique, a set of neighbour segments is obtained by finding the segments whose overlap (i.e. size of the intersection of URL sets) with the last segment is greatest. Currently we look for the 10 neighbours with highest overlap, subject to a threshold of 2. From these neighbour segments we tally and rank the URLs that do not occur in the last path segment, thus forming a recommendation list.

The second technique uses a matrix C of URL co-occurrence, rather as was used by Hinrich Schütze in [Schütze] for words inside documents. Each entry c_{ij} shows how often URL i occurred within the same segment as URL j. We collect the URLs that co-occur with each member of the last segment's URL set, then normalise by weighting each co-occurrence by $c_{ij} / |c_i|$, where c_i is the i^{th} row of C and $|c_i|$ is the root of the sum of the squares over the row. Then we tally and rank to make the recommendation list. As will be used in the next section, C can also serve in a similarity metric between any pair of URLs i and j, for example as a scalar product of rows: $c_i . c_j / |c_i||c_j|$.

We currently use window step sizes of 1 for co-occurrence and 10 (i.e. non-overlapping) for segments. Small step sizes give more accurate co-occurrence statistics, but the segment method suffers because high-overlap neighbours tend to be very close in the path, making recommendations less useful. In this case increasing the neighbour set size may also bring in more distant and interesting URLs.

The recommender is quite small on screen, showing just a list of URLs for each recommendation technique. The two URL lists are used inside a second tool, the segment browser, that may demonstrate the recommendation process more clearly. It is shown (in pieces) in Figure 1a, 1b and 1c. This tool can track web activity just like the recommender, but its primary purpose is to let one select any segment to be the 'target' for recommendations.

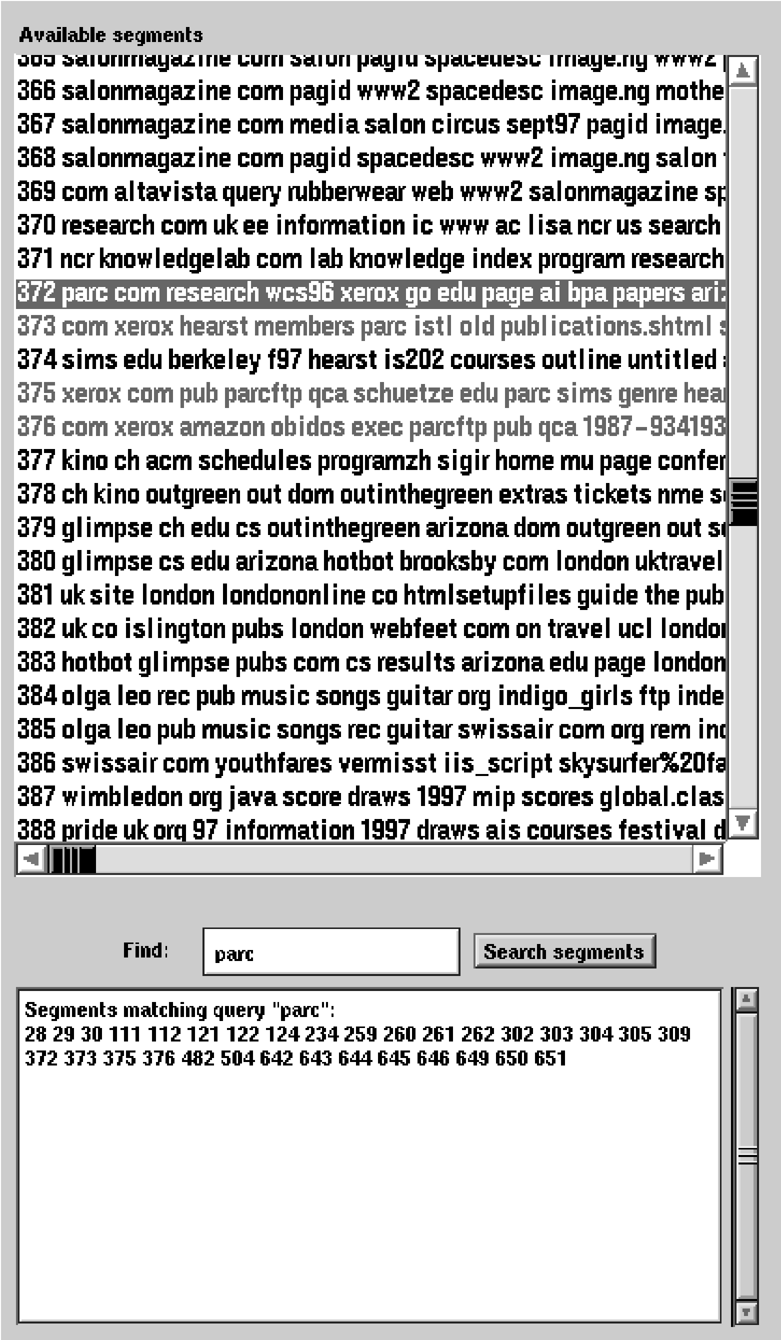


Figure 1a. The left section of the segment browser.

Here we see a list of 651 segments making up the concatenated paths of two of the authors, collected over ten weeks in Summer 1997. Each segment is represented by its most frequently occurring URL components and title words. Below this is a panel for searching for a string within the URLs. Matches are shown in light red e.g. segment 373. A segment selected with the mouse has its contained URLs and timespan shown in a panel in the centre of the browser, e.g. segment 372.

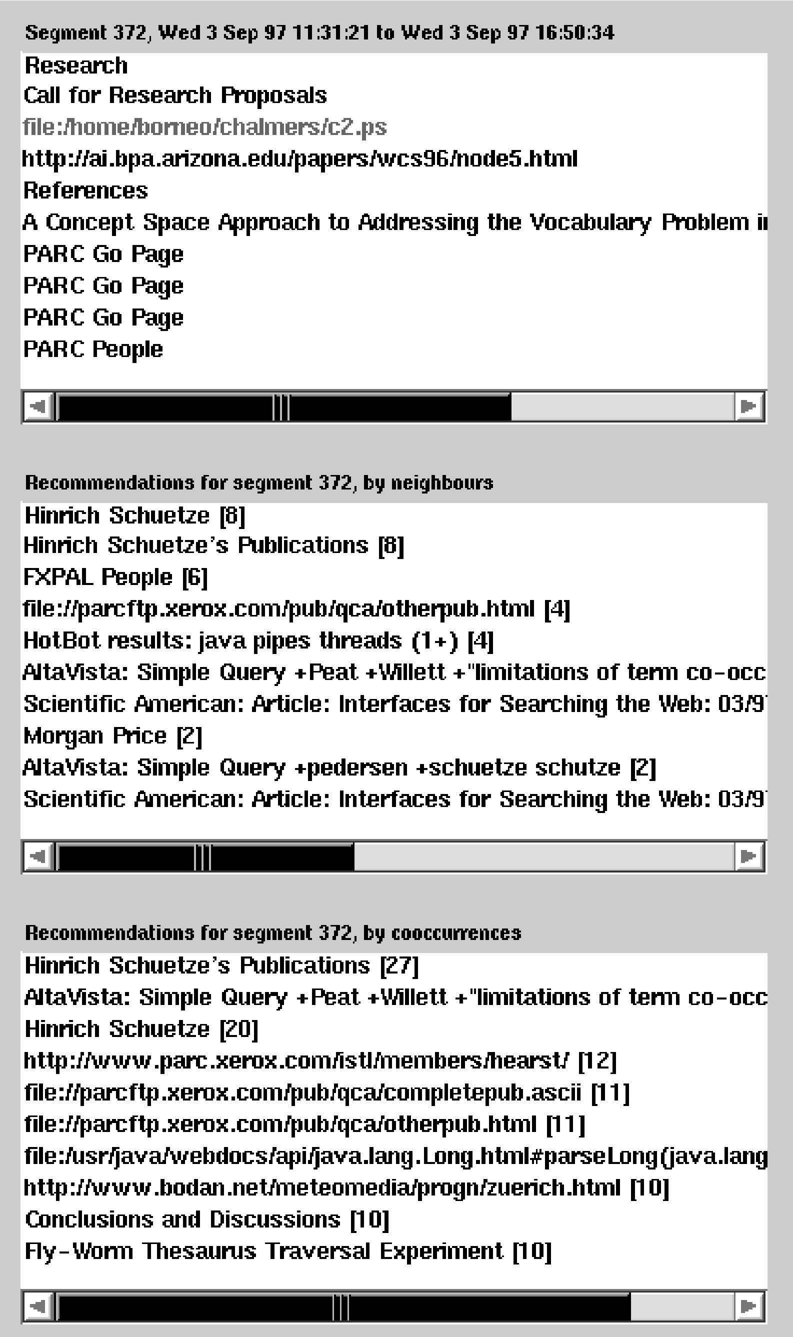


Figure 1b. The centre section of the segment browser.

Each displayed URL is checked to assess its accessibility and to obtain its title, if any; inaccessible URLs are shown in light grey e.g. the file c2.ps in Segment 372. Titles replace URLs, although one can still see the original URL by pressing the right mouse button over a title. With a left click on a URL or title, the segment browser sends a request to the web browser, to access the associated URL, and thus feeding back into the ongoing logging and recommendation process. The selected segment in the figure (#372) shows where one author had just accessed the top page for Xerox PARC (with title "PARC Go Page" repeated due to redirections), and then the "PARC People" page. The two URL recommendation lists are also shown for the selected segment. The URLs given as recommendations include the home pages of PARC's Hinrich Schütze, pages with his publications, a Scientific American article by Marti Hearst (formerly of PARC) on "Interfaces for Searching the Web", Marti Hearst's old PARC home page, and an introductory page at Fuji Xerox

Palo Alto Labs, a lab connected with PARC.

Also present in the recommendation lists is a page from a Java search and a page for weather forecasts (from www.bodan.net) for Zürich, that we look at regularly. While we may see it as thematically incongruous, so far it has co-occurred often enough to be recommended. As our data sets grow and our statistics build up, we observe such anomalies being gradually smoothed out. The neighbours of a selected segment are shown in the right of the segment browser, as in Figure 1c. The URLs and timespan of any neighbour can also be displayed upon mouse selection. Selecting any URL in the segment browser triggers display in the web browser.

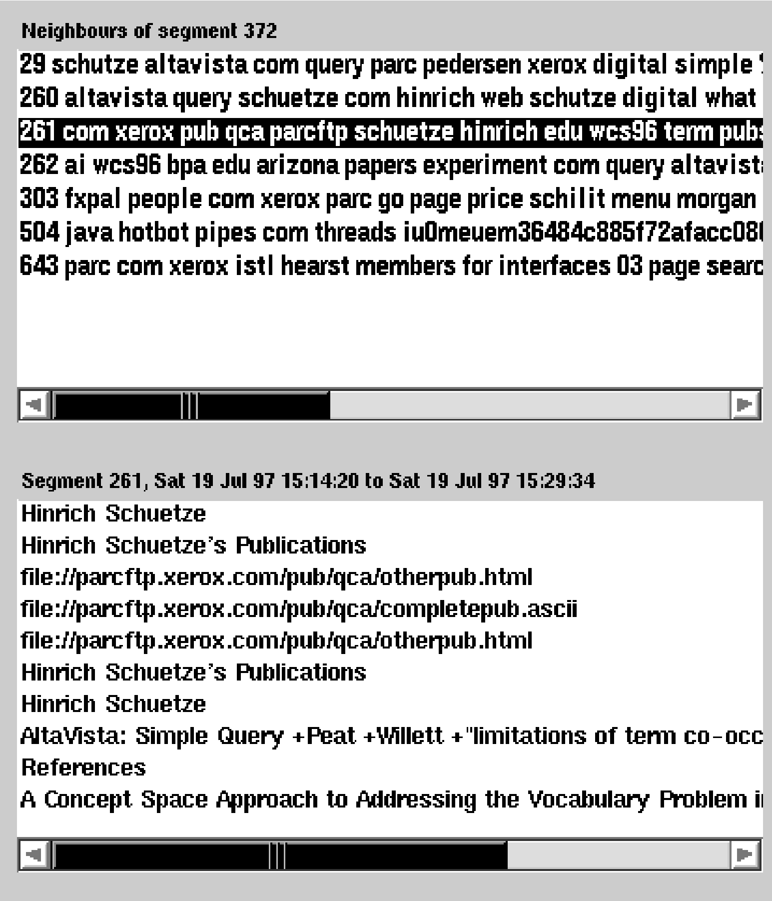


Figure 1c. The right section of the segment browser.

Recommendations for our selected segment in Figure 1c use seven neighbours, the first five neighbours being from one person, the last two from another. Selecting a neighbour with the mouse triggers display of its contained URLs, as with segment 261 here.

So far, our experience has been that using neighbours occasionally struggles against having small or null neighbour sets. When a segment required two or more shared URLs to be a neighbour of another, 75% of our example 651 segments had neighbours and the average neighbour set size was 2.4. With a threshold of three shared URLs, 51% had neighbours, and the average neighbour set size was 1.2. We use both techniques at this stage because the proportion of segments with neighbours is increasing as our paths grow, and because the neighbour method offers more specific recommendations than the co-occurrence method. The former produces its tally only from those segments that closely match the target segment, while the latter's tally involves statistics over the entire data set.

The recommender system is rather like an automatic bookmark system, reminding us of places we have been before. However, when we concatenate our path files -- after allowing an opportunity for a little judicious editing of any URLs that we may prefer not to show to others -- recommendations can include URLs that the user has never visited but that appear to be relevant given the current context. We gain the benefit of the path-building activities of others. Note that the recommendations may not be the pages to suit the general web-browsing population, as recommendations reflect the specific activities and interests of those whose paths are involved. They are the pages that are of most interest to *us*.

Since paths are associated with people, paths begin to externalise the community's use of and associations within information. Just as the patterns of mutual visibility in the physical world are strongly linked to mutual awareness and social behaviour, similarities in the way people have moved and are moving through information can make people perceptible to others. (Note, however, that here we do not have inherent symmetry or reciprocity as is normal in physical visibility on a street, or audibility in conversation.) This raises issues of privacy and invasiveness, but can enhance interpersonal communication and community [Bellotti]. At present we do not feel invaded upon because of our individual control over when logging is done, editing of the path file before combination, and our choice of whom we combine our paths with.

3 Visualising Path Components

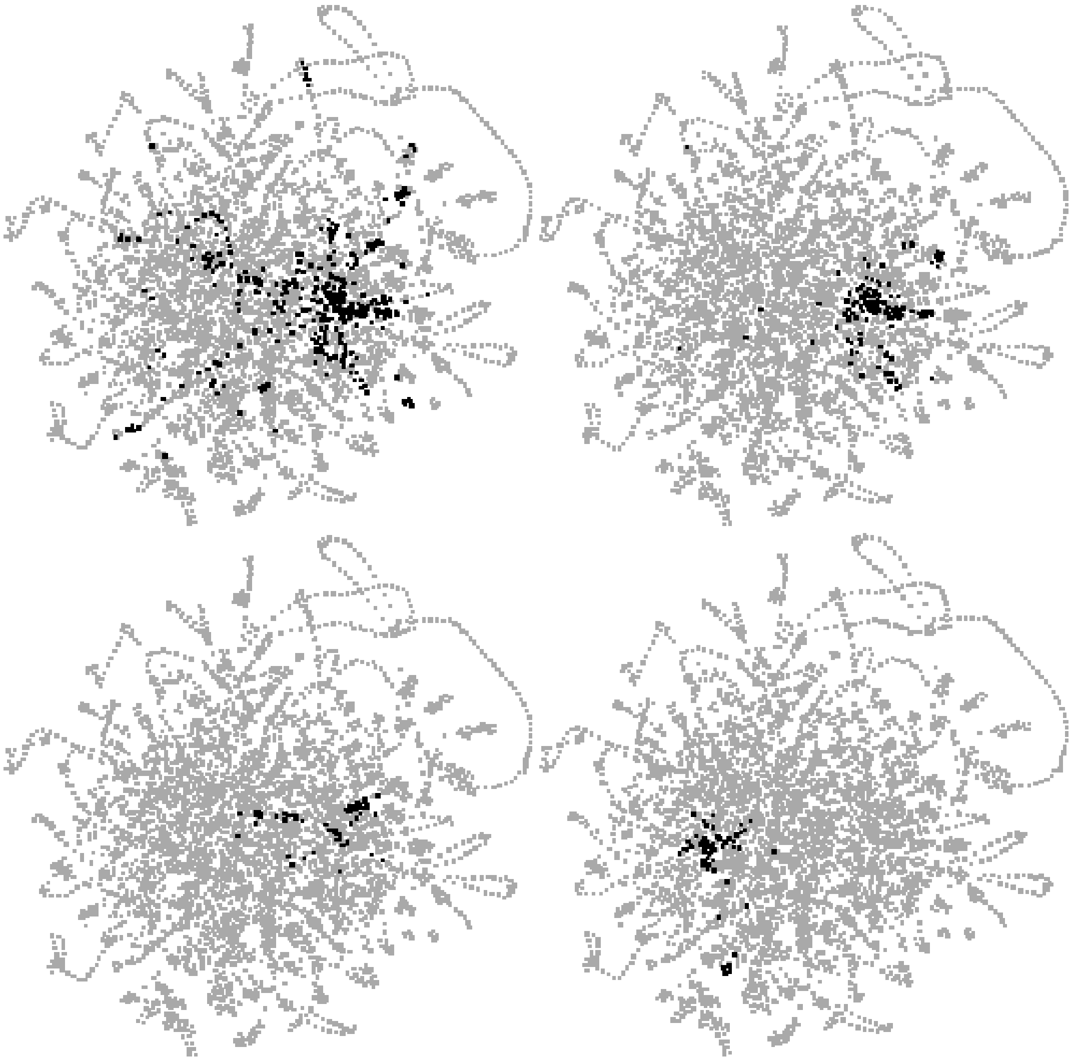
The recommender tool presents the URLs that appear to be most relevant to the most recently logged activity. The segment browser allows one to find the neighbours and recommendations for any path segment. It is however difficult to gain an impression of the overall configuration i.e. the 'big picture' of how the path components fit together. This is the traditional remit of graphics and visualisation, and so we made layouts of path components using the algorithm of [Chalmers96a] and the visualisation tool of [Brodbeck]. Other work, for example [Mukherjea], has concentrated on visualising the graph structure of a section of the Web. This has usually been a site, and also use has not been a fundamental aspect of representation. In contrast, we now describe the use the path model in a visualisation based on the activity of a particular set of people.

Our layouts rely on a metric of similarity being available for arbitrary pairs of objects. Here we use the scalar product of rows of the co-occurrence matrix, as described earlier. The layout algorithm initially places each object randomly in a three-dimensional space and then iteratively attempts to make the distances between all pairs of objects in this space correspond to their similarities. Objects push and pull on each other, iteratively moving closer to distant similar objects and further from close but dissimilar objects. An additional 'gravity' force pushes objects towards a ground plane. After a number of iterations, the layout process tends toward a stable set of positions, showing patterns and clusters within the modelled set of objects. Layouts are displayed in our visualisation tool where searches, selections, and other inquiries can be carried out.

The layout process pulls together URLs to form a chain in the layout corresponding to the path sequence. If a URL has only one occurrence, then the only co-occurrences will be very close by in the path. As the sliding window steps along the path URL by URL, the two URLs on either side of the URL in question will co-occur *len - 2* times, where *len* is the length of the window. The URLs two positions away will co-occur *len - 3* times, and so forth. The two URLs on either side of our singly-occurring URL in the path will be the closest in the layout. Sequences of singly-occurring URLs form a continuous chain, while each multiply-occurring URL causes the chain to loop back and cross over itself. URLs near to these recurrences are thus associated with each other, even though they may be from different periods of time. Brought close together, they begin to form clusters.

Figure 2, below, shows six searches in an example layout, displayed in our visualisation tool. Nearer the centre there are many crossings and interweavings, but looping sequences of singly-occurring URLs can be clearly seen, for example at top right there is a long curving chain representing a set of class files downloaded for an applet. The figure somewhat replicates a panel in the tool that displays the last five searches in miniature to aid comparison, but we show the searches in more detail here. (The tool's controls for searching and selection are also not shown or discussed here -- see [Brodbeck] for details.)

Searches for substrings allows us to begin to explore the layout e.g. to investigate clusters, and to see how particular strings relate to layout patterns. We are planning to extend searching by supporting queries based on person and time period. This would allow answers to such questions as "where am I now?", "who else has worked in this area before?", and "what was happening around time *t?*" The layout offers a common reference structure within which successive queries, selections and other activities take place.



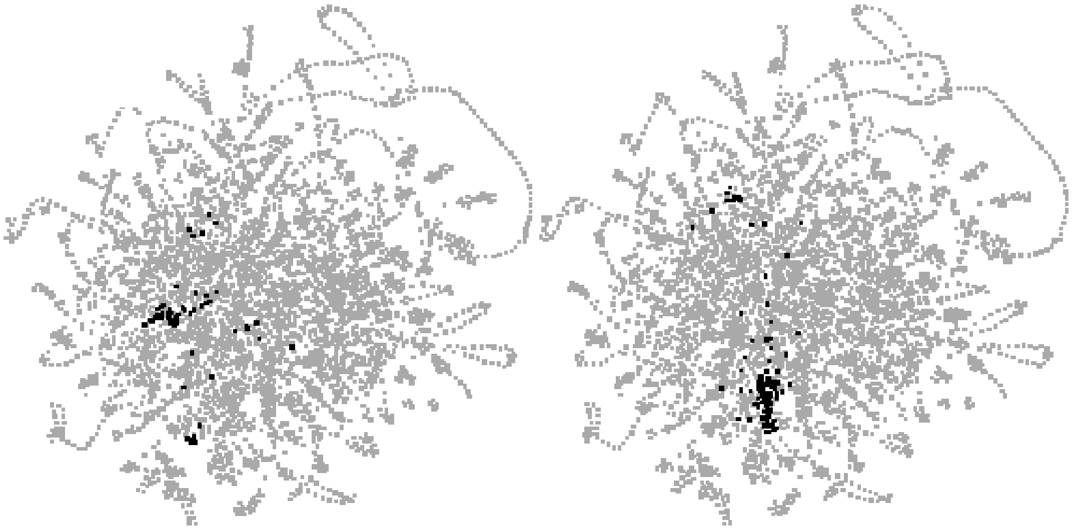


Figure 2. A layout of 4558 URLs, with matches for each of six searches highlighted: 'java', 'jdk', 'jgl', 'ubs', 'ubilab' and 'salon'.

Figure 2 shows URLs from the combination of the authors' three paths, showing roughly ten weeks' activity. In each view, a search for a text string has highlighted matching URLs in black, with the distributions of matches (and their overlap) accentuating patterns prolonged or recurring web activity e.g. reading Java documentation *(java, jdk, jgl),* work on our own web site *(ubs, ubilab)* and reading the *Salon* weekly web magazine.

Scale is a concern, both in layout time and in memory usage. The main clusters of Figure 2 took two days on a Silicon Graphics O2 to appear, and some fine detail took several days more. This may be in part due to the initial random placement, and we are looking at preclustering and exclusion of particular singly-accessed URLs as a way to improve this. Also, the quadratic size of *C* will at some time restrict us. We are currently investigating incremental, statistical approaches with lower order storage, and developing a second prototype that uses a relational database so as to support prolonged and concurrent use of recommendation and visualisation tools. Other future work is described in the next section.

4 Ongoing and Future Work

A number of our Ubilab colleagues have recently begun logging their web usage. Collecting more data will allow us to see how different combinations of people's paths affect the 'point of view' on the Web. As an example, when looking for information on databases one might restrict the choice of paths to those colleagues and friends one knows to be interested or experienced in that topic.

We also wish to extend logging to other tools, such as editors and mail readers, to better reflect the way that we interweave tools in everyday use. It may appear a trivial observation that we work with an ensemble of tools, but there has been little work that takes a holistic view of system use and adaptation. For example, a URL of interest may first be seen in a mail message or a document in a word processor but at present our system cannot recommend this source if the URL is accessed again. We should log all we can, interweaving logs from different tools using timestamps. Also, we wish to log words, names and URLs from within documents, handling all the 'symbols' we can, just as we currently handle URLs. We would avoid the narrow focus on content by using sliding windows that, like our current URL windows, span more than one document, and by similarly applying collaboratively-gained statistics on the context and frequency of use of these semiotic elements.

Lastly, we are obtaining usage logs for databases and trading systems from within UBS, to assess where else we may profitably apply the path model. We are looking for usage data from the 'communities' whose work is centred on such databases, where we can better perform larger-scale experimentation and evaluation.

5 Conclusion

By putting activity at the core of information representation, we have developed prototype tools for accessing Web information. Extending collaborative filtering to take better account of movement and context, the path model uses statistical consistencies in everyday Web use to build up associations and similarities, irrespective of content. Implementation issues such as algorithmic scale have to be addressed, and we need wider and longer-term use for deeper evaluation and feedback, but this early work has demonstrated the basic practicality of the approach and given us a platform for future development.

Taking a wider view, we may see beyond the traditional approaches to interaction, evaluation and analysis that treat each information access tool, each type of information, and each object in isolation. Accepting this complexity as natural lets us address issues such as the subjectivity of interpretation and the difficulty of formally describing information use. It also points out connections between information representation, interactive system design, and wider discourse on structure, language and human communication. We suggest that this wider view may help us build systems that better show, fit and support user activity.

Acknowledgments

Thanks to Chris Charlton, Steve Freeman, Hans-Peter Frei, David James and Lisa Tweedie for their comments on early drafts of this paper.

Kerry Rodden's current address is: Cambridge University Computer Laboratory, New Museums Site, Pembroke Street, Cambridge CB2 3QG, United Kingdom.

Lastly, we point out that shortly before submission, a colleague pointed out *www.alexa.com,* where a collaborative filtering tool apparently based on transitions between URLs is available. The work in this paper was performed independently, before we knew of Alexa, and we have not yet found out any details of their approach.

References

[Barrett] Barrett, R., Maglio, P. & Kellem, D. How to Personalize the Web, *Proc. CHI 97,* ACM, 1997, 75-82.

[Bellotti] Bellotti, V. & Sellen, A. Design for Privacy in Ubiquitous Computing Environments. *Proc. 3rd European Conf. on Computer Supported Cooperative Work, (ECSCW 93),* G. de Michelis, C. Simone and K. Schmidt (Eds.), Kluwer, 1993, 77-92.

[Berners-Lee] Berners-Lee, T. World-Wide Computer, *Comm. ACM,* 40(2), February 1997, 57-58.

[Brodbeck] Brodbeck, D., Chalmers, M., Lunzer, A. & Cotture, P. Domesticating Bead: Adapting an Information Visualization System to a Financial Institution, *Proc. IEEE Information Visualization 97,* Phoenix, October 1997, 73-80.

[Bush] Bush, V. As We May Think, *Atlantic Monthly,* July 1945. Reprinted in *ACM Interactions* 3(2), March 1996, 37-46.

[Chalmers96a] Chalmers, M. A Linear Iteration Time Layout Algorithm for Visualising High-Dimensional Data. *Proc. IEEE Visualization 96,* Oct.-Nov. 1996, 127-132.

[Chalmers96b] Chalmers, M., Ingram, R. & Pfranger, C. Adding Imageability Features to Information Displays. *Proc. ACM UIST 96,* Seattle, Nov. 1996, 33-39.

[Foucault] Foucault, M. *Archaeology of Knowledge,* trans. A.M. Sheridan Smith, Routledge, 1989.

[Goldberg] Goldberg, D., et al. Using Collaborative Filtering to Weave an Information Tapestry, *Comm. ACM 35(12),* December 1992, 61-70.

[Hillier] Hillier, B. *Space is the Machine,* Cambridge University Press, 1996.

[Lieberman] Lieberman, H. Autonomous Interface Agents, *Proc. CHI 97,* ACM, 1997, 67-74.

[Mukherjea] Mukherjea, S. & Foley, J. Visualizing the World-Wide Web with the Navigational View Builder, *Proc. 3rd Intl. World-Wide Web Conf.,* Darmstadt, April 1995.

[Nardi] Nardi, B. (ed.) *Context & Consciousness: Activity Theory and Human-Computer Interaction,* MIT Press, 1996.

[Pirolli] Pirolli, P., Pitkow, J. & Rao, R. Silk from a Sow's Ear: Extracting Usable Structures from the Web, *Proc. CHI 96,* ACM, 118-125.

[Pitkow97] Pitkow, J. & Pirolli, P. Life, Death and Lawfulness on the Electronic Frontier, *Proc. CHI 97,* ACM, 1997, 383-390.

[Resnick] Resnick, P. & Varian, H. (eds.) Special Issue of *Comm. ACM* on Recommender Systems, 40(3), March 1997.

[Schütze] Schütze, H. Dimensions of Meaning, *Proc. Supercomputing 92,* Minneapolis, 1992, 787-796.

[Shardanand] Shardanand, U. & Maes, P. Social Information Filtering: Algorithms for Automating "Word of Mouth," *Proc. CHI 95,* ACM, 1995, 210-217.

[Suchman] Suchman, L. *Plans and Situated Actions: The Problem of Human Machine Communication,* Cambridge University Press, 1987.

[Yan] Yan, T., Jacobsen, M., Garcia-Molina, H. & Dayal, U. From User Access Patterns to Dynamic Hypertext Linking, *Proc. 5th Intl. World-Wide Web Conf.,* Paris, May 1996.

Vitae

Matthew Chalmers has a B.Sc. in Computer Science from the University of Edinburgh, and a Ph.D.

in Computer Science from the University of East Anglia. After an internship at Xerox PARC he worked for almost five years at Rank Xerox EuroPARC in Cambridge, U.K. He joined Ubilab in 1995. His main interests are in information visualisation and retrieval systems, and in the theories of knowledge and interpretation that underlie them.

Kerry Rodden is a Ph.D. student at the University of Cambridge Computer Laboratory, and worked at Ubilab as an intern in the summer of 1997. She has a B.Sc. in Computer Science from the University of Strathclyde, Glasgow, and as an undergraduate she spent three months at CERN, Geneva. She is interested in user interaction with information systems.

Dominique Brodbeck received his Ph.D. in Physics from the University of Basel, Switzerland. He then worked at the IBM Almaden Research Center in San Jose, California for four years during which he built up a center of competence for scientific visualization and later shifted his focus to information visualization for data mining systems. His main interests lie in information visualization and design, and visual computing and interactive media in general.