

# A Case for Usage Tracking to Relate Digital Objects

Elin Rønby Pedersen  
Google, Inc.

1600 Amphitheater Parkway  
Mountain View, CA 94043  
elinp@google.com

Jeanine Spence  
Microsoft

One Microsoft Way  
Redmond, WA 98052  
jeanines@microsoft.com

## ABSTRACT

This paper covers the evolution of the concept of Usage Tracking to automatically link digital objects such as documents. Extensive ethnographic studies of information work have revealed that establishing and maintaining relationships – between documents, between artifacts and between people – is at the core of information work. Focusing on just one aspect of this challenge, we looked for practical ways of relating digital documents. Leveraging the fieldwork, we designed a mechanism that captures the user’s activity across documents and reinterprets it as links between these documents. We implemented the mechanism as a running prototype to assess the feasibility of the concept, and in general gauge the opportunities to make better use of usage data – which are mostly gets ignored in today’s computing platforms.

Object to object relation building through usage data has three important advantages over most existing methods for automatically establishing relations: first, it is behaviorist, not relying on guesswork about the user’s intentions; second, it is media agnostic: text, images and sounds are all just objects and treated alike; it is the user’s handling of the objects that matter, and third, it is application agnostic: it does not rely on privileged access to specific applications.

## 1 INTRODUCTION

Advances in computing and information processing have provided a wealth of data to the information worker, but at the cost of massive information overload. As information technology brought us into this problem; can it also be used to solve it?

A true and tested way to mitigate the problem of overload is to organize and cluster the information. However, few people are good at keeping their files organized. Even for those who are there arises a conflict between the organization principle and the actual needs; for instance, a

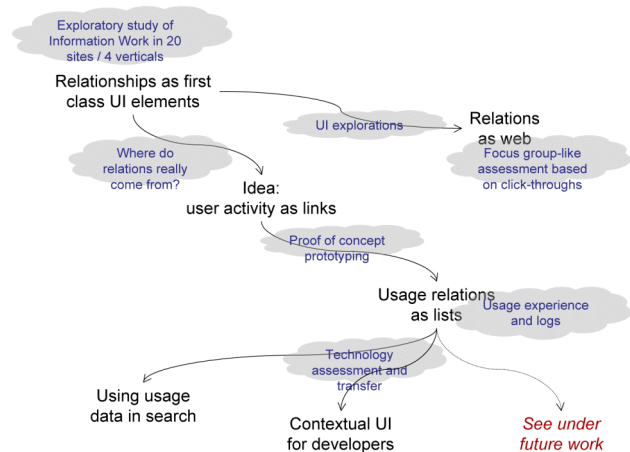


Figure 1: Overview of project: from an extensive study of information workers, to review of existing technology and research, to prototyping and lessons learned

single document belongs to several non-overlapping categories, or these categories erode and change over time. And for the rest of us, it is often a challenge to find the proper place to store the document when we are in a rush, and then to re-find what we prematurely categorized.

To address the challenge of information overload we realized we had to know more about what information work is about, where the main problems seemed to lie, and how people dealt with them. An ethnographic study information work was launched, its result suggested that major challenges for information workers lie in establishing and maintaining relations; an extensive literature and technology review ensued, calling for new approaches to relating documents; proof-of-concept prototyping led to new perspectives on the utility of user activity data

This paper describes the course of this inquiry; see also the outlined mapping of the inquiry in Figure 1. A separate publication [17] describes technical details of the prototype.

## 2 INFORMATION WORK

More than 20 companies were visited over a span of nine months to surface main obstacles to productivity and satisfaction, and untapped capabilities of the information workers. Combined interview and field observations of work practices covered four different vertical markets:

medical practices, professional services (like architecture, engineering, financial services), trade services (like distribution, import/export), and, specialty manufacturing. We present here an overview of the most relevant findings.

The study focused on four different vertical markets: medical practices, professional services (like architecture, engineering, financial services firms), trade services (like distribution, import/export), and, specialty manufacturing. Over 20 companies were visited in a span of 9 months. In-depth studies of work sites were carried out, using a combination of eco-system mapping and ethnographic work practice studies.

The studies provided a wealth of insights about the evolving nature of information work. We briefly review some of the most relevant findings below; some are similar to those found in prior research.

*Repetition and workflow:* Emerging work patterns and flows are not captured. Information workers spend time doing roughly what they have done before; working on roughly the same clusters of documents.

*Multiple applications:* Information workers juggle multiple applications when working on almost every task [11].

*“Computer amnesia”:* As tasks start, are interrupted, stop and restart, the computer “forgets” all the cross application connections that are part of the task. Information workers have to remember the relationship between documents or forms and do not get any help from the computer. Information workers might be able to mitigate computer amnesia through better information organization, but either do not get to categorizing, or do it prematurely, then to find they forgot where information was placed [4]. Adding to the problem is that different techniques are required for different applications or by different people.

*Preserving context:* A major part of information work is to calibrate and reconcile, to bring back context that may have been lost. Information workers often retain the contextual “keys” to information in their heads [11].

*Interruptions:* Interruptions and fragmentation of work is increasing. Workers spend significant time re-finding key information to recover from interruption [3, 5, 18].

*Complex individual work styles:* Many people go to great length to adjust their physical and digital work space to fit their personal work style [12].

## **2.1 Focus on relations and interruptions**

We found that relations permeate everything! People work on the same documents and the same workflows again and again establishing and maintaining intricate implicit relations – often kept in their head! Interruption is another persistent factor in information work. Relations and interruption doesn’t work together. Information workers spend significant time re-finding key information in order to recover from interruptions. In recent years we have seen different approaches to help people get the benefits of

organized document storage without requiring them to do all the work themselves.

An obvious response, we thought would be to make relations prominent – make them 1<sup>st</sup> class citizens in the interface. Then we should allow the user to navigate based on relations not just storage location. We envisioned a mechanism to serve the user ambient information about potentially relevant digital objects using these relations.

There was only one nagging question: wherefrom would we get the relations? This led to an extensive search in existing technologies for making and maintaining relationships between documents.

## **3 EXISTING RESEARCH AND TECHNOLOGIES**

First we considered work that seeks to establish relationships among documents based on their content. Secondly, we looked into work in desktop task/activity monitoring and management systems, in particular looking for techniques to establish relationships among user documents.

### **3.1 Content Based Relation Building**

Implicit structure and similarity in document content can be used to define and measure the relationship between two documents. Three approaches stood out as candidates: (a) schema driven (e.g., FAQs and Info Sleuth [2]), (b) expression driven (e.g., Apple Data Detectors (ADD) [15], and Excel Web Query), and (c) term driven (e.g., Stuff I’ve Seen [8], Haystack [1], and Google Desktop). Conceptually, these three approaches exploit different ways of structuring information in documents.

However, based on our observations of information workers and the material they handle we find that relying on content similarity would miss many documents a user might think of as relevant to particular activity. For instance, an architect would, in some cases, work with her to-do lists (text), CAD drawings (special graphics format), web pages of local building codes we would be hard pressed to find any way to derive their inherent relationship from textual analysis. Also, our study of information work shows that even documents that are basically text and numbers might be inaccessible due to proprietary formats in enterprise systems (e.g., SAP).

### **3.2 Link analysis**

A different way of characterizing documents without doing textual analysis per se is the page rank method deployed in search engines like Google: here the documents obtain a relevance value through an iterative analysis of the relevance of documents that reference it.

### **3.3 Meta-data Based Relation Building**

Meta-data are textual descriptors of digital objects. Some meta-data are generated automatically as a bi-product of the general handling of them, like where the user places an object in the system of folders; who created and/or modified the document, when it was last touched, etc.

Socially based relationships can be discovered from who created the document or who modified it.

Other meta-data is created and explicitly attached to the digital objects when the user applies tags to digital objects. Systems like del.icio.us and Dogear [13] allow users to provide explicit metadata that can then be used to facilitate search based on tag identity or similarity. A problem with tagging is of course that it requires users to make the relationships that are relevant explicit through the application of tags. Finally we see strategies for “automatic tagging” by instrumenting applications to characterize the digital objects further, for instance TeamTrack [6].

### 3.4 Activity Based Relation Building

A completely different view of document relationship discovery is less focused on documents themselves. Instead, the activities of the user around a document or section are deemed the critical information for discovering document relationships. How the user reads, edits, copies, pastes, emails as attachment, receives, or downloads, indeed any action a user can take with the document, are used to discover key relationships

*Task tracing:* Work on task tracing and activity monitoring typically requires the user to initially define the boundary relevant tasks. In TaskTracer [7] the user indicates when she begins a task and when the task is complete. TaskTracer monitors documents and user activity to learn relevant folders and file locations, specific files manipulated, and a range of application settings relevant to the task. Once the user has done this the first time, on subsequent engagements with the task, TaskTracer will identify the active task and reset the state of relevant applications and documents.

ActivityExplorer [14] takes a slightly different approach. In ActivityExplorer the user specifies the boundary of tasks by explicitly indicating the set of documents that are part of the task. In essence, this model has the user explicitly indicate how documents are related; there is no automatic relationship discovery. Explicit articulation of activity in ActivityExplorer, combined with tagging, has been combined in a search interface to exploit the user specified relationships.

Relying on “tasks” as the principle means of document relationship discovery requires identifying the connections of one piece of information in one application to “task” related information in the same or other applications. Often a single task or workflow will require the use of multiple, differing applications, resulting in different interaction techniques and different representations to support the user.

*Usage tracking:* Usage tracking has been deployed for a variety of purposes under a common rubric of looking for the user’s intention. For instance, implicit feedback systems attempt to infer user intent based on observable behavior, such as which documents she does and does not select for viewing, and how long she views them [16].

The Lumiere project went even further in inferring intention and using this information in Office Assistant, almost to level of conversational characters [10]. Among others it taught us some important lesson about carefully situating the help you want to provide to the user and being careful about second-guessing the user.

### 3.5 Still looking...

Let us briefly recap what we were looking for and what we found. We wanted a “live” service that would show documents to the user that were likely to be relevant for her in that very moment.

Many solutions require the user to define the relations explicitly. This requirement of user contribution gets in the way of fluidity as the users frequently switches among unrelated tasks. In general it would be great if less premeditation were required.

Another bulk of solutions for automatic relationship building is based on inferring relations from similarity in content. One challenge of automatic categorization is to create relations that the user will understand and appreciate. For that very reason we were leery about relationships based on automatic categorizing outside of established categories of the user’s profession. In general, building relations based on content has severe limitations: most of our content analysis tools are limited to text, but many documents today are not textual, and relations that matters to users may not be only those of categorical similarity.

The task or activity based approaches relied largely on the user denoting the task parameters. However, real information work is much more fluid and full of fuzzy bookend’ing of tasks and non-explicit categorization.

## 4 OUR APPROACH

Learning from all these existing techniques, we looked for an approach that would not require content analysis nor recognition of task to infer intent.

Revisiting the fieldwork, in particular the findings “*Repetition and workflow*” and “*Computer Amnesia*”, led to the idea of tracking usage activity on the documents: if the user works on a set of documents at the same time they may or may not be related in her mind, but if she works again and again with the same set of documents, chances are that she really see them as connected. We named this approach Ivan for easy reference. In brief, Ivan monitors the user’s activity, taking particular notice when documents are on the screen together, when the user switches back and forth between them, or when the user cuts, copies and pastes from one document to another.

The Ivan approach may be best characterized as a blend of recommendation system like Amazon’s book advice: “when you previously used this document you also looked at these documents,” and Google’s page ranking: “this document is one that you have used so much or so little with other open documents.”

The approach is **radically behaviorist** in the sense that it is of no consequence why the user might use two documents in timely proximity; we just note the fact that she does – assuming that she might later find it useful to be served (information about) documents that were used together regardless of whether they “belong” to a single task or several. That is, we do not attempt to make extensive assumptions about what the tasks a user might be engaged in that again might trigger the use of related documents. In this aspect we differ from past work on behavioral modeling and usage tracking systems.

The captured usage data are generic in several senses of the word. They are **application agnostic**: we do not need privileged access to the individual applications as long as we can monitor the underlying system events. And they are also **media agnostic**, as opposed to most strategies for determining relevance and relatedness that derive relations out of content similarity or meta-data.

We suggest that usage tracking mechanisms like Ivan can improve life for the information worker in two ways, (1) by helping her quickly re-find repeatedly used clusters of documents, and (2) by offloading parts of the mental work that goes into (re-) establishing and maintaining key task/document relationships.

## 5 IMPLEMENTATION

We went on to prove the concept and feasibility of the Ivan approach; we built a running prototype, which was then used for the remainder of the project (4 months) by the

team members as well as a couple of peers.

The prototype consists of three main parts, as also shown in Figure 2: an *event capture and filtering* module, a *relation building* module and a *UI module* for visualizing the relations and receiving user actions on relations. The first two modules were designed to run in the background.

Our implementation was intentionally simple in order to give a rough sense of the usefulness of this kind of service without going into depth on a single visualization. The prototype was built as a stand-alone service for Windows XP. The code is C# for all algorithmic parts and UI, and C++ for the interface with event capture APIs. We used the Active Accessibility API to provide the streams of events from the desktop, and the file system filter driver, FileSpy, to report file system events, both of which are from Microsoft. Similar tools for message spying are built into most operating systems. Relations are stored and maintained in a local Access database, entirely under the control of the user.

A detailed description of the proof-of-concept prototype is provided elsewhere [17]. Here we will briefly describe how the service works, mostly from a user’s point of view.

### 5.1 Event Capture

The event-capture module works by intercepting messages from the file system, keyboard and window manager, and “cleaning” these event streams of all the stuff we are not interested in. This module is responsible for synchronizing

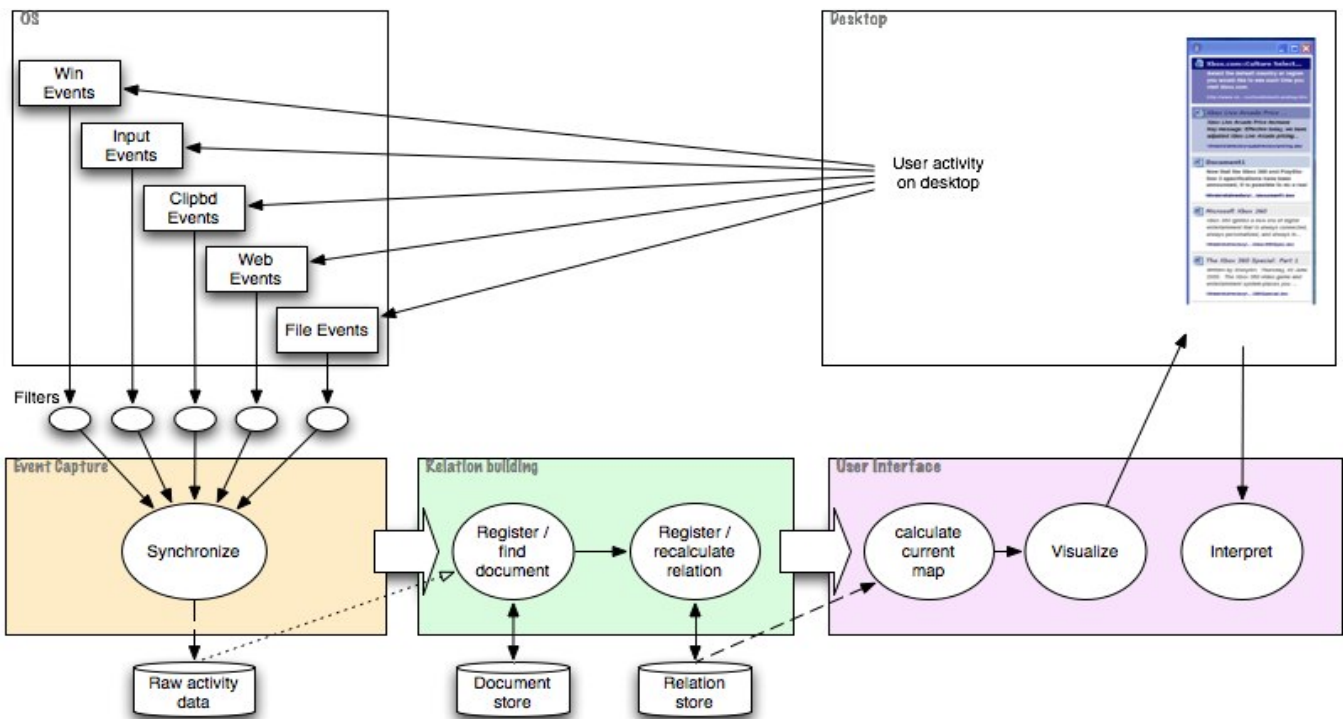


Figure 2: System architecture showing that processing passes through three main phases: event capture, relation building, and UI for suggesting documents that may be of relevance to the document in focus

between the files and the active documents, mapping the displayed documents to either temporary IDs or fully specified file IDs.

The result is a “digested” event of the following types: Create document; Open/Close document; Got/Lost Focus; Save/Save as document; Copy/Cut/Paste material; Mouse Click/Double-Click.

These events are streamed to the Relation Builder. We also store them in a database of raw usage or activity data. This is both for backup and to allow the event-capture to run with delayed relation building.

## 5.2 Relation Building

The relation builder is responsible for two major tasks: (1) determine what document is currently in focus; (2) determine how the current event might influence the relations that involve the focal document.

It creates and maintains the Document Store, which is a mapping from external document ID (file ID or URI) to a local key for all documents touched while the usage tracking service has been running. The key is used to find relation objects in Relation Store. The history of each relation, with time and change is stored with relation object.

The only event that causes a new relation to be created is co-presence (or time proximity) of two documents not previously related. In this event we create a weak, tentative relationship (lowest value). If the co-presence turns out to be merely coincidental, the strength of this relation will soon decrease and eventually vanish.

Relations are strengthened by activity. After creation, the strength of the relation is increased by the following events: repeated co-presence, repeated shift of focus from one document to another, and copy/paste content between documents. A simple logarithmic mapping modulates growth; this is in order to keep values within our range, and also to reduce bias against new documents.

Relations are weakened by lack of use. The current implementation fades relations indiscriminately, but we are aware that we will have to develop a more complex algorithm for “aging” of relations, reflecting the prevalence of periodic workflows and “rhythms” as described by Begole et al [3] and also confirmed among the information worker we studied: Many tasks are performed only once a month or once a quarter, and it is precisely due to their



Figure 3: Implemented side panel UI for usage-based relevance

infrequency the user benefits from these suggestions of relevant material.

When done the Relation Builder passes information about the document that is currently in focus on to the UI module. It is up to that module to figure out how to use the information.

## 5.3 User Interaction

The user interacts with the usage tracking service through a side panel (see Figure 3), which displays a list of items, corresponding to open documents and documents related to those open documents.

The side panel keeps track of which documents are open and, when triggered by user activity, it recalculates the set of relation objects for each open document. In this implementation we are looking at first-order relations only. Deeper relations can be extracted at any time from the Relation Store.

When the user sees a document of interest in the list, she can click it. It will open and gain focus, and the panel will be redrawn according to the new relation network.

Each item shows the title of the document in a top bar, following an icon for its type. Below the bar is a snippet from the document; it may be the starting phrase of the document or, a copied/pasted segment. Finally, the bottom part of the item is the “address” of the document, i.e., a file system path or a URL.

Open documents have colored background; closed document backgrounds are white. The focal document and the closed documents related to it are shown with darker bars on the item. Looking at the side panel we see two open documents (shaded backgrounds), one of which is in focus (darkest shade) and three unopened documents (white backgrounds), one of which relates to the focal document (darkest bar).

## 6 LESSONS LEARNED

The goal of this prototype was to investigate the concept of relation building by usage tracking. That is, whether it made immediate sense and provided useful data to the user, and whether there would be any truly hard technical problems in performing usage tracking. The result was mixed: the user assessments of the service suggest that usage tracking is indeed useful; however, our implementation of the service was not robust enough. A

technical analysis showed that the problem is deeply embedded in the way the desktop operating system works.

### 6.1 Usefulness

Six people used the service over approximately one month; we gathered activity logs from the system and bug reports from the users, and we conducted an evaluation interview at the end. The comments and evaluations collected from our users suggest that usage tracking is potentially a powerful mechanism that should be further developed and perhaps also applied beyond suggestions of related documents. Examples of the feedback were:

*Explaining the immediate affordance*, “When you bring up a document, it will show you all the documents you have worked on at the same time. So I can easily get on from where I left it last.”

*Looking into the creation history*, “Often I loose track of where I get my pictures from. I add a photo to a paper, and - a week later - I want to replace it with a better one. But where is the original? Your system will tell me.”

*One person’s false positive may be another person’s treat*, “I am looking for a particular version of the budget, one of many. But I remember I worked on this one when I got the email about Paris from Jen; that makes it easy - I use Jen’s email as a kind of handle” (virtue of false positives)

*Understanding the mechanism at work*, “It takes some time getting used to. First I expected the system to try to be smart; I dreaded another Clippie! Then I realized: it’s left to me to make sense of the relations - which is fine; that is easy for me” (non-categorical associations).

The user comments were also quite constructive in terms of other UIs for the usage tracking mechanism. We will touch on some of these alternative UI options later in this section.

Having a good confirmation of the potential usefulness of the approach, we decided not to do any further user studies in this round; any deeper studies of usability were likely to gauge the interim UI rather than the usage tracking mechanism itself – which would have been a waste of

### 6.3 Feasibility of hybrid approaches

It is important to note that while this prototype was tightly scoped to demonstrate the feasibility of a pure activity based approach, it can and should be made to work alongside content-based approaches as we see them applied in desktop search and like.

In general we are dealing with two very different paradigms: the primary data in *usage tracking* are the weighted relations, but we can use those weights of the relation to derive a “usage importance” value for each document. It is the other way around for the *content-based approaches*: there the primary data are about the documents, and we derive “relations” between documents by looking for various kinds of similarity in the contents.

effort. In general, engineering prototyping and user studies have to calibrate each other. Too much engineering may be wasted if not frequently vetted for usefulness; and the instruments we use for gauging usefulness have to fit the stage of development.

### 6.2 Technical challenges

Perhaps we should not have been so surprised about how difficult it was to create a generic event capture. Horvitz et al wrote, “(...) it is critical to gain access to a stream of user actions. Unfortunately, systems and applications have not been written with an eye to user modeling” [9].

We expected to fare better than they did because we just wanted to know about a few basic events: identification of the document in focus though the filename (or temporary identifier if not yet saved) ,to receive notification only when documents were created, opened, or closed, and about the copy and paste of stuff between documents.

The problems arose partly from an inherent asynchrony between the window management and the file system: only the application knows how the stuff displayed in a window relates to a file in the file system and each application has its own way of managing this association. For instance, some applications keep a newly created document in memory until it is closed and saved; only then would a file system event appear. Other applications engage the file system immediately and work with temporary file names, thus the first closure of a new file would also be a renaming event. These types of application idiosyncrasies go much broader making the capture of generic events into a maze of conditionals.

Another major cause of complexity is a certain redundancy in the interfaces between applications and underlying modules like the window management. E.g., when looking directly at the event stream as it was sent to our service, we noticed that the open document user action could result in multiple sets of open events, as if numerous independent agents each gave their check off before the application could start using the document.

To illustrate how a hybrid of these two approaches might look, we mocked up two designs of usage tracking integrated with a desktop search service

*Expanding content-based search with usage-based relations*: For instance, if we search on our desktop for “constel” we will get all the documents that contain this string, in filename, tags or contents. If we linger over one of the hits, say the image file *constellens2.gif*, an expanded desktop search service could show all the documents that have been used at the same time as the image file, for instance, as shown in Figure 4 as a new pane in the lower right corner of the desktop search panel. Note that this way we might be able to find related documents that would never been “caught” by a content-based method, for instance the art work file *Focus\_Periphery.ai* in the mock-up, CAD drawings and untagged photographs.

*Filtering content-based search with usage-based importance:* While the essential qualities that we get from usage tracking are relations between digital objects, we can calculate derivatives from them to obtain a usage-based measure of importance or relevance for each object. In the current implementation we calculate a very simplistic measure of relative importance by finding the maximum strength of any relations for an object. As illustrated in Figure 5 we can integrate this in desktop search and it can be used interchangeably with the ranking we get from traditional search.

Another usage-based measure for objects can be calculated directly from the data in Document Store, regardless of their interaction with other objects: how many times has it been opened, how long was it open, was it recent, etc. This would be a measure for the “wear and tear” of the object, along the same lines suggested in [9]. Such a feature would make it easy to quickly identify – and possibly filter out – documents that have no usage value, for instance all the PDF-files that get downloaded and never opened.

#### 6.4 Future work

It is important to note that while this prototype was tightly scoped to demonstrate the feasibility of a pure activity based approach, it can and should be made to work alongside content-based approaches as we see them applied in desktop search and like.

Going forward we are eager to find alternative sources for interaction event capture. Rather than just waiting for the desktop operating systems to accommodate user activity tracking, we see the web platform as a potential shortcut to a friendlier environment for activity capture. The rapid increase in range of web-based applications and services of relevance to information workers (such as various Live services, Groove, Google apps, Salesforce.com) makes this alternative even more interesting.

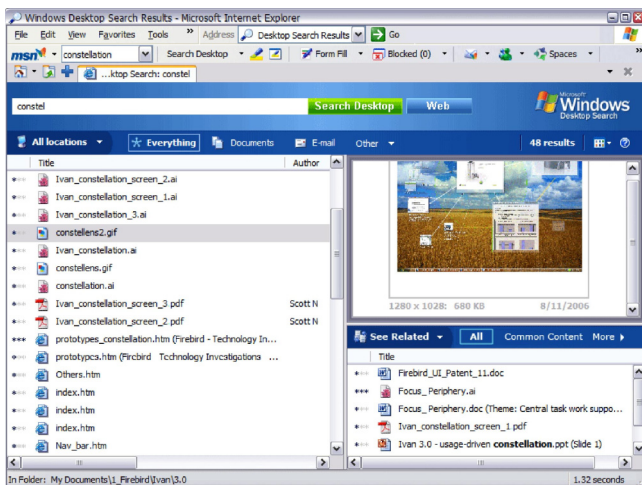
We need to investigate what might be a useful granularity for digital objects by looking at what a user perceives as a perceptual unit. In the current implementation we used a simplistic concept of a digital object to mean more or less a document with its own file name and showing in its own window; but it meant we were unable to cover prevalent email clients (many objects in one file and one window) or transaction data (many views, volatile data). Another simplification we employed was the use of only binary, non-directed relations between objects. It will be interesting to see if improved user experience can be achieved with more complex relationships that include time and sequence data. With directed relations we may be able to more accurately relate object within task flows and derive new flows out of these usage relations.

It remains to be investigated if better, more robust measures of usage-based importance will enable new functionality. In terms of algorithmic approach, we would first to try out a parallel to page ranking by scanning the relation network iteratively. We will continue oscillating between tackling technical problems and fine-tuning heuristics by observing and consulting users; and as the technical prototypes firm up we will also do more rigorous user assessment.

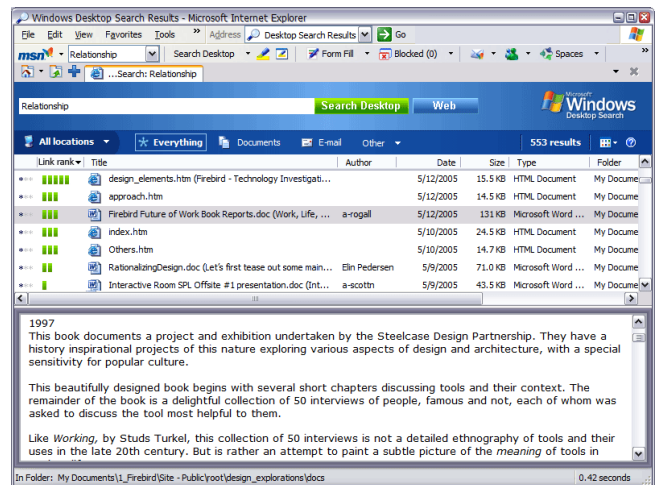
#### 7 CONCLUSIONS

In this project we saw how qualitative user studies inspired technology innovation (the behaviorist approach to relation building), how a simple proof of concept prototype revealed some severe technical challenges (the difficulty of matching file system events with window events), while also allowing us to get a first sense of the usefulness of the approach (users’ immediate grasp of the concept, confirmation of benefit assumptions).

Looking ahead we see usage tracking as an interesting additional source of data that can enable numerous improvements in both functionality and user experience.



**Figure 4: First mockup of Usage tracking in desktop search.** The lower right pane shows object related to the object in the left pane that we are lingering over



**Figure 5: Second mockup of usage tracking in desktop search.** The column with green bars shows the “link level” for each document (link level possibly being equivalent to relevance or

## ACKNOWLEDGMENTS

The authors wish to thank David W. McDonald who spearheaded the literature and technology review and provided invaluable comments to earlier drafts. Thanks also to Scott Neilson for catalyzing our imagination with envisioning illustrations.

## REFERENCES

- 1 Adar, E., Kargar, D. and Stein, L.A., Haystack: Per-User Information Environments. In Proc. CIKM'99, (1999), 413-422.
- 2 Bayardo Jr., et al., InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments. In Proc. SIGMOD '97, (1997), 195-206.
- 3 Begole, J., Tang, J.C., Smith, R.B. and Yankelovich, N., Work Rhythms: Analyzing Visualizations of Awareness Histories of Distributed Groups. In Computer Supported Cooperative Work, 334-343.
- 4 Berlin, L.M., Jeffries, R., O'Day, V., Paepcke, A. and Wharton, C., Where Did You Put It? Issues in the Design and Use of a Group Memory. In Proc. InterCHI'93, ACM Press, 1993, 23-30.
- 5 Czerwinski, M., Horvitz, E. and Wilhite, S., A Diary Study of Task Switching and Interruptions. In Proc. CHI'04, ACM Press 2004, 175-182.
- 6 DeLine, R., Khella, A., Czerwinski, M. and Robertson, G., Towards understanding programs through wear-based filtering. In Proc. Software Visualization 2005. ACM Press 2005.
- 7 Dragunov, A.N., Dietterich, T.G., Johnsrude, K., McLaughlin, M., Li, L. and Herlocker, J., TaskTracer: A Desktop Environment to Support Multi-tasking Knowledge Workers. In Proc. IUI'05. ACM Press 2005.
- 8 Dumais, S.T., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R. and Robbins, D.C., Stuff I've Seen: A System for Personal Information Retrieval and Re-Use. In Proc. SIGIR'03, (2003), 72-79.
- 9 Hill, W.C., Hollan, J.D., Wroblewski, D., McCandless, T., Edit wear and read wear. June 1992. Proc. CHI 1992. ACM Press
- 10 Horvitz, E., J. Breese, D. Heckerman, D. Hovel, K. Rommelse, The Lumière Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In Proc. Uncertainty in Artificial Intelligence. Morgan Kaufmann, 1998.
- 11 Kidd, A., The Marks are on the Knowledge Worker. In Proc. CHI 1994, ACM Press 1994. 186 - 191.
- 12 Malone, T.W. How Do People Organize Their Desks? Implications for the Design of Office Information Systems. ACM Transactions on Office Information Systems, 1 (1). 99-112.
- 13 Millen, D.R., Feinberg, J. and Kerr, B., Dogear: Social Bookmarking in the Enterprise. in Proc. CHI'06, ACM Press 2006, 111-120.
- 14 Millen, D.R., M.J. Muller, Werner Geyer, Eric Wilcox, Beth Brownholtz. Understanding users and usage patterns: Patterns of media use in an activity-centric collaborative environment. In Proc. CHI 2005 ACM Press, 2005.
- 15 Nardi, B.A., Miller, J.R. and Wright, D.J. Collaborative, Programmable Intelligent Agents. Communications of the ACM, 41 (3). 96-104.
- 16 Oard, D. W., and Kim, J., Modeling Information Content Using Observable Behavior. In Proceedings of the 64 Annual Meeting of the American Society for Information Science and Technology, USA, 2001.
- 17 Pedersen, E.R, and McDonald, D.W., Relating Documents via User Activity: The Missing Link. To appear in Proc. IUI 2008, ACM Press 2008.
- 18 Whittaker, S., Frohlich, D. and Daly-Jones, O., Informal Workplace Communication: What is it Like and How Might we Support it? In Proc CHI'94, ACM Press 1994.