

Interaction Harvesting for Document Retrieval

by

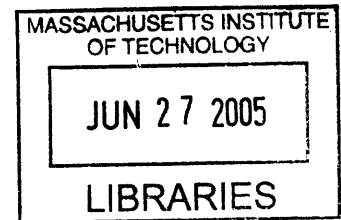
Noah S. Fields

B.F.A. Fine Arts
Massachusetts College of Art (1992)

Submitted to the Program of Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

**Master of Science in Media Arts
at the
Massachusetts Institute of Technology**

June 2005



© 2005 Massachusetts Institute of Technology. All rights reserved.

Signature of Author: _____

Department of Media Arts and Sciences

Certified by: _____

John Maeda
E. Rudge and Nancy Allen Professor of Media Arts and Sciences
MIT Media Laboratory

1

Accepted by: _____

LU
Andrew Lippman
Senior Principal Research Scientist
MIT Media Laboratory

ROTCH

Interaction Harvesting for Document Retrieval

by

Noah S. Fields

Submitted to the Program of Media Arts and Sciences,
School of Architecture and Planning,
on May 6, 2005 in partial fulfillment of the
requirement for the degree of
Master of Science in Media Arts and Sciences

Abstract

Despite advances in search technology, few software systems have been developed which accurately categorize multimedia files. The most successful systems for searching images, sounds, or movies rely on keyword annotation to provide meaningful search terms for non-text documents. Unfortunately, such systems usually require the author to enter the keywords manually, a task that is commonly neglected, or is executed poorly. This thesis proposes an approach to document categorization called Interaction Harvesting, wherein systems establish document relationships based on organizational and curatorial cues, harvested from the mouse and click gestures of an online community. Specifically, the spatial and temporal proximity and placement of documents are taken as indicators of document similarity. We propose an expansion technique whereby such proximal documents exert weighted keyword influences on each other. We hypothesize that these approaches will form a document classification framework that relieves some of the difficulty of the annotation process, while providing keyword-equivalent retrieval performance.

Thesis Supervisor: John Maeda

Title: E. Rudge and Nancy Allen Professor of Media Arts and Sciences

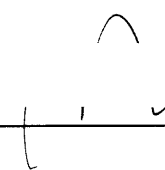
Interaction Harvesting for Document Retrieval

by

Noah S. Fields

Thesis Committee:


Advisor:


John Maeda
E. Rudge and Nancy Allen Professor of Media Arts and Sciences
MIT Media Laboratory

Reader:


Walter Bender
Senior Principal Research Scientist
MIT Media Laboratory

Reader:


Deb Roy
AT&T Career Development Professor of Media Arts and Sciences
MIT Media Laboratory

1	INTRODUCTION.....	8
1.1	Goals.....	8
1.2	Approach.....	9
1.3	Context of project.....	10
1.4	Methodology used.....	10
2	BACKGROUND.....	12
2.1	Motivation.....	12
2.2	Document retrieval principles.....	16
2.2.1	Manual document indexing.....	16
2.2.2	Automatic document indexing:.....	23
2.2.3	Keyword expansion techniques.....	26
2.2.4	Systems that leverage user communities to add meaning to documents.....	27
2.2.5	Systems that observe the user.....	36
2.3	Data storage principles.....	37
2.3.1	Traditional file systems.....	38
2.3.2	Databases.....	39
2.3.3	Geographic information systems.....	40
2.3.4	Relational file systems.....	40
2.4	Findings from historical work.....	41
3	METHOD.....	42
3.1	Observations.....	42
3.1.1	Real-world organizational observations.....	42
3.1.2	Virtual file organization techniques.....	43
3.1.3	Image placement study.....	43
3.2	Software design.....	48
3.2.1	Goals of treehouse project.....	48

3.2.2	Treehouse Studio Faceted File System	49
3.2.3	Relational database storage.....	49
3.2.4	Organizer software	53
3.2.5	Image Landscapes software.....	53
3.2.6	Similar Viewer.....	63
3.2.7	File Organizer	67
4	ANALYSIS	70
4.1	Existing commercial applications	70
4.1.1	iPhoto	70
4.1.2	Picasa.....	71
4.1.3	Flickr	72
4.1.4	del.icio.us	72
4.1.5	Google.....	73
4.2	Analysis of completed software	73
4.3	Future directions	76
5	CONCLUSION	78
6	LIST OF FIGURES.....	80
7	BIBLIOGRAPHY	81

1 Introduction

1.1 Goals

The kinds of information people are consuming and processing is changing. Our information space is rapidly become more expressive. Most of the information we consume and create is stored in multimedia sources: images, movies, and sounds. As a larger percentage of our information is stored in these rich formats, we also seek innovative ways to organize and search for the documents and information we need. Documents can be organized manually by humans, or by

computational processes. Manual document-indexing procedures can be labor intensive, even impractical for the era in which we live. Traditional computation approaches to automatic document categorization and indexing are typically inflexible, and require that the data be in a known content format, such as ASCII text.

Professional stock photographers and photo banks have elaborate organizational procedures that help them categorize their photographic assets. These organizational structures exist because the photographers expend time categorizing and tagging their photographs. The better the organization of the imagery, the easier it is to find photographs for clients. Similarly, digital asset-management software relies heavily on the user to provide text annotation for each asset. Although professionals know the value of keyword tagging, it can be a time-consuming task; non-professionals rarely take this time to annotate their images, and as a result, the knowledge represented by those images is difficult to recover. What is needed is a system that facilitates or partially automates the categorization process, making the media more retrievable, while reducing the cost of annotation for users.

1.2 Approach

The Interaction Harvesting approach that we propose is a system wherein documents are characterized by how the users interact with their documents through an application interface. In the simplest case, the system observes users' actions as they manipulate media files; these actions are recorded in a database, where they are combined with the actions of other users of the system, and analyzed by software agents. A specific application is presented which demonstrates the Interaction Harvesting technique. In this application, users create a spatial narrative with images from their photo libraries. By observing the curatorial decision-making process of the user as she positions documents in the landscape, the system is able to enhance its own representation of how these documents are related. A detailed description of how these observations are

used to establish similarities between documents is given in the proposed approach.

1.3 Context of project

The need to provide a framework for document classification and searching came from findings from the Treehouse Studio, an ambitious project that attempts to create an online creative economy and community of digital artisans. The mission of the Treehouse Studio project is to provide users from all over the world free online access to tools, resources, and infrastructure with which to create collaborative multimedia projects. In addition to tools for creating digital media, the project aims to provide portfolio and document management facilities. A key design goal of the Treehouse Studio must be to provide easy, fast, and accurate document retrieval to support these tasks.

The Treehouse Studio is well suited for research involving document retrieval, because it uses a central, relational database to store file contents and file attributes, and therefore can quickly read and write attributes such as creation time, keywords, and location. Additionally, the Treehouse Studio is a multi-user system already equipped with a document-retrieval framework. Within this framework, applications can be developed to explore document retrieval using Interaction Harvesting document classification.

1.4 Methodology used

The histories and literature of document retrieval and user-interface design were examined. Image categorization techniques used by friends and colleagues were observed during simple informal workshop exercises. Initial software sketches were made based upon observations of how users interacted with documents while engaged in a creative task. An attempt was made to integrate current trends of folksonomic document-classification techniques within these sketches.

Within the framework of the Treehouse Studio, these applications have been developed further to explore opportunities in document retrieval using Interaction Harvesting document classification as a novel approach. Simple non-rigorous observations about the software performance have been noted.

The first example of an Interaction Harvesting application called “Narrative Landscape” has been completed. This application is a presentation application similar to a slide-show editor. Users assemble groups of media files and arrange them in a specific order and on specific locations of the screen as they create their presentation. As the author composes and arranges these media files, the user interface communicates these actions to the Treehouse database, where the interactions are recorded in real time. Information about when documents were added, where they were placed, etc., is stored. Document clustering happens as a result of correlating one or more of these gesture attributes amongst the documents. Spatial clusters, for example, are groups of documents that have been placed near each other.

A second image browsing application was also developed to demonstrate the effectiveness of Interaction Harvesting. This application uses the document-similarity metrics established by an aggregated set of user interactions with the documents as obtained from the Image Landscape software to browse documents on the server.

Finally, a file system called the Faceted File System was architected and deployed within the Treehouse as a result of these trial applications. This file system architecture was informed by lessons learned in the construction of the interface harvesting applications, and features a high-performance and flexible meta-data format, explicitly for use in image harvesting and faceted classification applications.

2 Background

This thesis describes an innovative approach to document retrieval, and, by extension, document classification, storage, and indexing. Although we typically conceive of document retrieval as a contemporary concern, the problem is old, and it is important to consider its origins as we innovate. Our exploration of background material relevant to this topic will take us from cuneiform tablets to the most recent trends of faceted categorization, as well as grassroots approaches to document classification online.

2.1 Motivation

I remember being a very young student, perhaps in the third grade, walking through the bookshelves of my school library. I remember looking at the poster of a gorilla scratching his head as I was walking through the tall towering shelves of books and browsing. This was my first experience with document retrieval. I would walk up and down all the isles of shelves, and occasionally pull a book off, and check out the cover. Examining the cover of a book pulled from a shelf gave me some idea as to what section I was browsing though at any given moment. Before learning how to use the card catalog, I had a spatial map in my head about which books were located where in the library.

I vividly remember the library lesson concerning the card catalog and Dewey Decimal System. It confused me. I was sure that the small pencils and torn scraps of paper were just as important as the small drawers full of the neatly stacked cards. I remember being impressed by the number of neatly typed cards contained within those small drawers. I remember that we transcribed the mysterious numbers from the cards in the drawers onto the scraps of paper, and then wandered the library, in search of these same numbers on the sides of books. I distinctly remember thinking how stupid it seemed to be looking for

these numbers, instead of trying to find the subjects directly in the shelves, using my tried-and-true wandering technique.

I am not sure when it happened, but I eventually learned how to look up subjects in the catalog, and then how to find the books on the shelves based on those magic numbers written on those torn pieces of paper. When I was comfortable with the card catalog, and no longer afraid of it, I never really thought about it again, until quite recently.

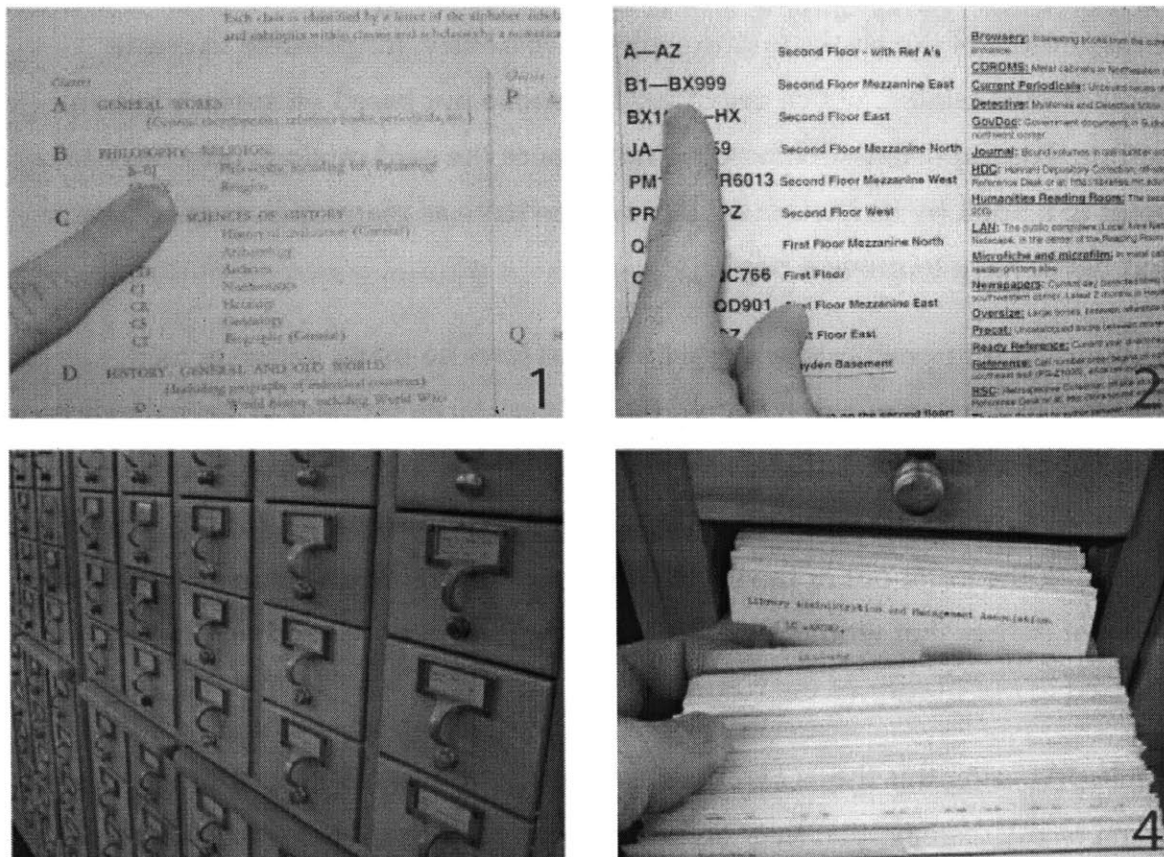


Figure 1: Using a card catalog.

Now I have a great appreciation for the how the Dewey decimal classification system (DDC) concisely organizes documents into well-defined subject-category spaces. The single most revolutionary developments in the history of document retrieval, however, may be the card catalog of a large document collection. Although arranging books on the shelves using the DDC is useful on its own

(after all, as a child I was able to locate books in the library simply by knowing that similar books were located near each other), the index of the card catalog is the abstraction that makes libraries so useful.

The utility of the card catalog is derived from its ability to present the index of the library without any of the bulkiness of the book contents. This allows users to browse all of the subjects quickly, and just in a few square feet. The catalog is a virtual organization of the physical objects: we call such virtual organizations indices. In the case of typical card catalogs, the index is a virtual subject arrangement of the books on the shelves. Obviously, it's possible to create another card catalog, which organizes the books not based on subject, but on author. The important idea here is that because the card catalogs only have to point to a book by its DDC identifier, we can have as many indices as we want, and never have to move a book.

Although it is impractical for a real library to have so many separate card catalogs, in the digital age we could build a very large number of virtual indices. We could create indices organized by the number of times a book was taken off of the shelf, how heavy the book is, when it was printed, or by the last person who took the book out of the library. We could even imagine a meta-index: an index of indices that would allow us to see all the ways a given book has been indexed in our library of the future.

While preparing this thesis I went in search of a card catalog to photograph, and found to my dismay, that they are very uncommon these days, having been replaced by electronic systems that serve the same purpose. It is easy to see why: index cards typed by hand are not easy to maintain. I was able to find some old card catalog index cards, and I took them home with me, because they are such interesting artifacts now that we have entered the era when almost all indices are kept electronically.

As difficult as it is to create and maintain a physical card catalog index, it is inexpensive relative to the cost of continually moving or shelving books. In the world of digital indices and documents, having multiple indices is essentially free. The only cost is that of the human or computer power required to build the index in the first place.

Search engines such as Google represent efforts where computational power is brought to bear on the problem of creating a large number of indices for a large number of documents. As one of my readers comments: "Google is like a robot racing ahead of you in the library stacks, rearranging books to conform to your current whim!" However, software systems such as Google only tend to function well in search domains such as text or structured data.

Multimedia authoring tools, ubiquitous recording devices, new media distribution channels, high-speed networks, and larger storage capacities are all conspiring to turn the vast majority of our digital files into rich media experiences. Text is no longer the predominant document format. The tried and true software-based document-indexing techniques may not be general-purpose enough to see us through this new era of digital media archiving. How do we organize our digital libraries? Are we doomed to archive media without an effective search and retrieval method, a kind of write-only archive?

Some search techniques have been developed which are able to index and search specific media formats. Examples include image searching based on color attributes and music searching based on musical characteristics [BPMS01]. With a few exceptions, these categorization techniques describe formal aspects of the media they are indexing, and are generally unable to interpret or ascertain the subject area of the document in question. Surely a flexible and data-agnostic means of document classification and retrieval can be devised which requires neither document interpretation nor tedious direct user tagging. Can we leverage multi-user environments to add meaning and searchability to all of our

media files? Developments such as Flickr and del.icio.us seem to indicate that it might be possible. What would such a system look like?

2.2 Document retrieval principles

2.2.1 Manual document indexing

People have been organizing documents for centuries. In this digital age, we hardly consider manual document sorting and indexing tasks, but many of the most fundamental and influential principles of document classification are quite old. We examine the origins of manual organizational schemes in order to reveal interesting features and principles, but also to draw inspiration. There is a rich history of document classification and information management to explore.

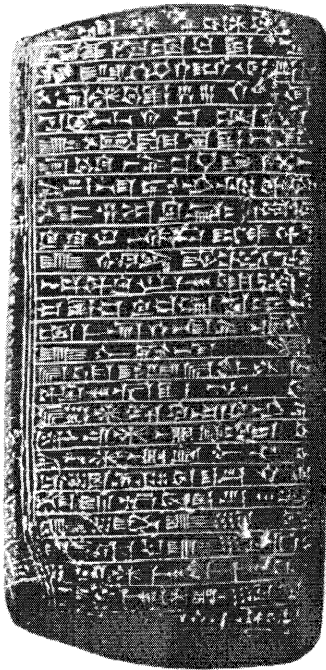


Figure 2: cuneiform tablet.

Origins of Document Retrieval

Babylonian cuneiform documents were written on clay tablets that were bound together and stored in baskets. One of the first advancements in document management was summarizing the tablets' contents on the thin edge, much the way that we write the title of books on their spines today. The advantage of this is clear: stacked tablets could be identified without removing them from the stack.

Not much is known about how documents were stored at the great library at Alexandria, but there are some accounts that state that the scrolls there were stored in bins, and thus could only be vaguely organized into piles. The scrolls were wrapped around rods, which had tags attached to them. These tags were the scroll's index, and listed the author and the title of the document [Bat03].

In both of these examples, document management consists of merely adding a local summary to a document, something easily visible without having to open the document itself. These local summaries are like filenames in our computer systems, only one gets the sense that the indices of Alexandria may have provided slightly more information than the filenames of today. Once you have a document in front of you, these labels and brief summaries are helpful. But these techniques would not help you locate the scroll or cuneiform you are looking for in such a vast building.

One of the earlier examples of a great library complete with a documented method for locating specific works was the great house of learning in Cairo of 1000 AD. There were over half a million books in this collection including many illuminated Korans. The books were kept in large cabinets, each of which displayed an index of the books it contained. Here we have a somewhat improved index, with groups of documents summarized in one location, instead of locally on the outer edge of the document.

In 1290 AD the Sorbonne school cataloged its comparatively smaller collection of 1,000+ books, organizing the written catalog alphabetically by title. Instead of merely identifying the contents of various stacks, this is one of the first known examples of an organizational principle being brought to bear on the entire collection.

By 1475 the Vatican Library, under the direction of Pope Nicholas V and Pope Sixtus IV, took document indexing one step further. A special library was constructed to store and organize the Vatican's large collection. The library was divided into three rooms: a room for Greek, a room for Latin, and a room for Hebrew texts. Each room followed the same layout of many rows of tables where the books were shelved and physically fastened. The tables each represented subject areas, and were the secondary organizational means of the Library. The first tables shelved only Bibles. The second table held works by the fathers of the Church. The third table contained philosophy books concerning

spirituality. This organization continued through multiple subject areas, all oriented in relation to the Bible. Each room had its own catalog of documents. The catalogs organized the entire collection by subject, reflecting the organization of the rooms. These catalogs also included an alphabetized list of the books on each table.

The Vatican system made several improvements upon the Sorbonne system. First, the architecture of the library was exploited to organize the books according to the language of the texts being organized. The second layer of organization was a hierarchical subject index, which mapped onto a physical space. Finally, there was an alphabetized listing of the individual subject areas in the collection's catalog [Bat03].

Dewey decimal classification

Despite the advances of publication tools such as the printing press, not much changed in the world of document retrieval or library sciences until 1876 when Melville Dewey unified some of the organizational musings pioneered by his predecessors in library science. Melville concocted a subject classification scheme that was to be used for all books in all libraries. Up until Dewey's system, all books were indexed according to their actual shelf. Adding a new book to the shelf under the old system could potentially mean breaking the organizational framework for all of the documents. Identifying subject areas by "location" on the shelves was common, and as collections grew, the indices would also often need to be updated. Dewey's contribution was to define a virtual one-dimensional space that defined regions of subject matter. We can think of the Dewey decimal classification space as an infinite line, or a virtual shelving system. Along this line we have regions of subject matter with all related subjects adjacent to each other. In some ways we know that items located near each other on the line are about the same or related topic. By creating a virtual shelving system, Melville solved the problem of addressing books. By mapping a

framework of classification onto the virtual shelving system, Melville has provided us with a reliable retrieval mechanism [Bat03].

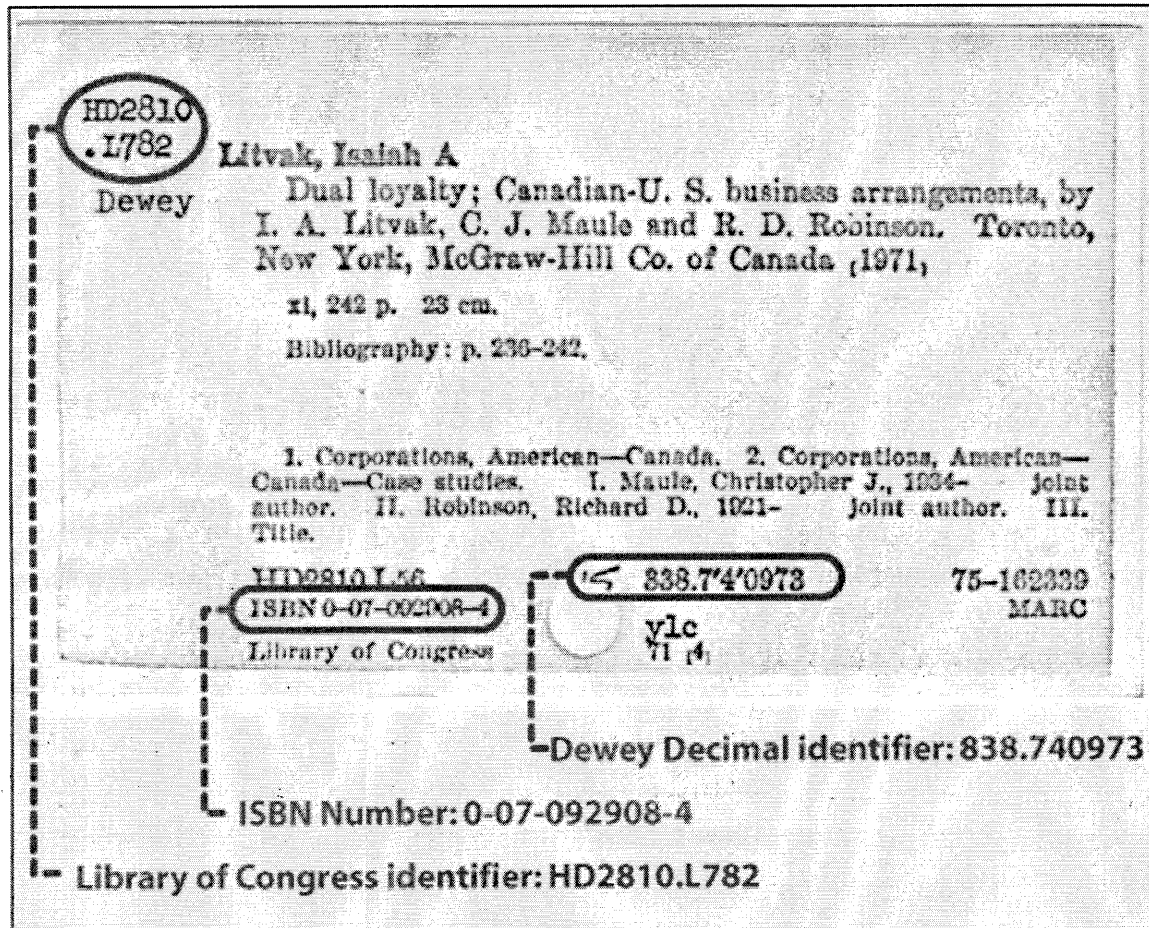


Figure 3: Index card anatomy.

Library of Congress classification of 1897

Herbert Putnam and Charles Ammi Cutter developed this system for library organization in 1897. The Library of Congress classification system is very similar to the DDC, but the LCC uses letters in addition to numbers to break up the subject areas. The LCC system was devised specifically for the Library of Congress, but has enjoyed popularity at most academic libraries in the United States. Public libraries primarily rely on the DDC. For our purposes here, there is no significant differentiation between the DDC and LCC systems [LOC05b] [Bat03].

Universal decimal classification

The Universal Decimal Classification System (UDC) is based on the DDC, but is extended to include a richer categorization method, allowing a document to be categorized by multiple subject numbers and tied together by special operator symbols. Paul Otlet and Henri La Fontaine devised this elaborate scheme at the beginning of the 20th century. Its main purpose is to provide more specific and nuanced subject-matching using a faceted notation. Two or more decimal classification numbers are given, along with a special operator character. Though the UDC is still in use today, it does not have the general popularity of the DDC, or the LLC.

“In UDC, the universe of information (all recorded knowledge) is treated as a coherent system, built of related parts, in contrast to a specialized classification, in which related subjects are treated as subsidiary even though in their own right they may be of major importance. Thus specialists may often be led to related information of which they would otherwise have been unaware.”
[UDC05]

ISBN numbering

Perhaps the latest book classification scheme is the International Standard Book Number. This system of book classification is concerned only with the establishment of creating a unique identifier for every book published. Although other information such as where the book was printed can be obtained from the ISBN number, these serial numbers are not designed to help locate books so much as to assure that every title has a unique identifying number.

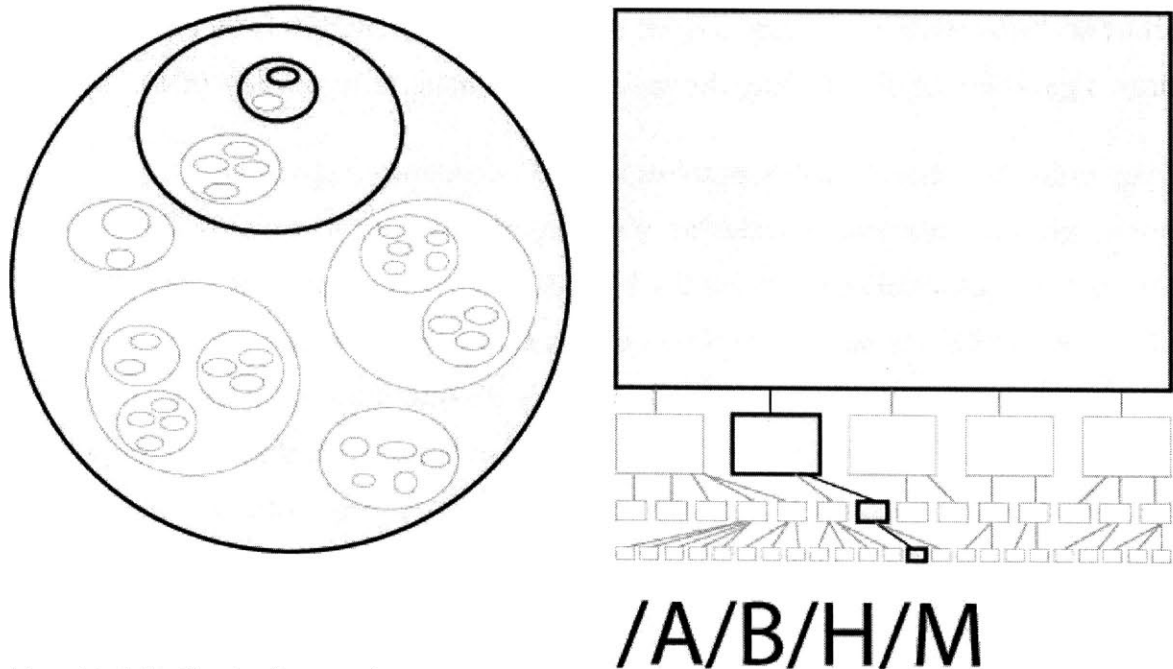
ISBN classification was established in 1966 by the bookstore W. H. Smith in the United Kingdom. Later, in 1970, it was adopted as the international standard (ISO 2108). The ISBN number is a 10-digit number divided into four sections, separated by spaces or dashes. The first digit represents the country where the book was published, the second digit identifies the publisher, the third digit represents the publication number for that publisher, and the last digit is a “checksum” (a method for validating the authenticity of the number). The

identification numbering system does little to organize materials by content, but does a good job of distributing the task of establishing unique identifiers.

Despite the fact that the ISBN number does not inherently specify an organizational structure or indexing mechanism, the very fact that there is one unique and canonical number for the book is very important. This unique identifier will help to search for the books in other contexts. The nature of the number guarantees uniqueness, and therefore disambiguation in the document retrieval process. Numerous reference tools have been created for establishing authoritative mappings between ISBN and DDC LLC entries. One example of such a reference tool is <http://isbndb.com/> , a free online ISBN-to-DDC and ISBN-to-LLC cross-reference tool [ISB05].

Hierarchical classification

History is full of examples of hierarchical systems of classification. In such systems, items are categorized by a series of set membership tests, starting with the broadest categories, and culminating in tests of the most specific criteria and sets with the finest granularity. Arguably the most well-known of such systems is the Linnaean taxonomy, which establishes the classification of species. In this taxonomy all living things are classified into very fine-grained sets of similarity. The name of a species uniquely identifies it, but more than that, the name also specifies all the criteria that the species meets for set membership at higher levels of granularity. Often there is one characteristic which functions as a litmus test for set membership. The species name in this case acts almost like a functional descriptor for the animal in question. However, it should be recognized that the names and classification of species is not transparent to a non-expert. Oftentimes such specialized systems demand a level of expertise – not only by the person doing the classification, but also by the people who will be using the index.



hierarchical classification frameworks

Figure 4: Two ways of looking at Hierarchical Classification Schemes.

Figure 1 is an illustration of two ways of looking hierarchical classification schemes. On the left we view hierarchical classification in a Venn diagram. We see that every entry is contained by the category above it, and that set membership is a refinement process. On the right we view Hierarchical Classification as a tree graph, where it is clear that every entry in the system describes its ancestry. Note that file paths and directory structures are an example of a hierarchical classification system. One obvious advantage of hierarchical classification schemes is a one-to-one mapping of index elements to names. For example, in Figure 1, the box letters that were traversed to get to the bottom-most leaf can be used to name the shaded leaf node (smallest box at the bottom) as “/A/B/H/M”. This name uniquely identifies the shaded leaf node, and also describes its logical heritage.

Hierarchical Classification Systems are tremendously powerful tools; most of modern science is guided by such systems of classification. All great manual indexing systems are powerful because professionals carefully craft them. Searching for content in a card catalog is often more effective than using a

Google search, however we have crossed the threshold where publication is outpacing our ability to index. We have grown accustomed to very fine-grained indexing. We now index entire documents at the page, paragraph, and sentence level. We can no longer rely upon the labor and good will of library scientists to solve our document-classification problems.

2.2.2 Automatic document indexing:

For documents in which the content type is known, such as text documents, it is possible to devise reliable algorithmic methods for creating relevant document indices, saving lots of time for human document organizers. For example, almost all web indices are arrived at computationally. Some attempts have been made to computationally classify and organize other specific media types, such as music and images.

Text document retrieval

It is relatively simple to create a brute force index of a text document. It is somewhat more complex to identify the significant and meaningful words from a document adequately describing and classifying that document. Reliable methods have been demonstrated that accurately analyze a document and return a series of keywords that accurately describe the contents of a text document.

Term significance

We can imagine a naive brute force system for indexing all of the words contained in a document. If we count all of the words of a document and come up with a word-frequency plot, we find that the largest number of words are relatively meaningless, and tell us nothing about the document's subject matter. For instance "the," "it," and "a" all score pretty high on most document word-count indices. We need a method to derive not just the frequency of a word, but also the importance of a word. Algorithms for determining word significance in

text documents have been under investigation for some time, resulting in a series of term-weighting functions.

One technique, introduced by H.P. Luhn, is to measure statistically the term frequency of one document relative to other documents in a collection. The statistical frequency of a term can be then used to derive the term significance after the removal of so-called “stop terms” [Luh57].

Vector-space Model

Gerard Salton refined Luhn’s frequency analysis, so that term distribution for a document collection is taken into account when attempting to weigh the significance of words occurring in documents. Term-frequency variance from document to document is used to as a significant factor in the term-weighting process [Sal83]. A vector of a document’s term frequencies can be compared to other documents in a larger document collection. Terms that have frequencies that differ significantly from the average within the collection are given more significance. The term weight (w) for a document term (t) is proportional to the inverse of the frequency of the term in the document collection (D/df) as a whole.

$$w_i = tf_i * \log\left(\frac{D}{df_i}\right)$$

Equation 1: Term differentiation weighting

Salton also describes an indexing system that categorizes documents based on term-frequency similarities. Once an appropriate keyword dictionary has been established by choosing the keywords with the highest weight, each document is then analyzed for each the frequency of each keyword. Salton uses this vector as a multidimensional space of document similarity. The distances between

documents in each keyword dimension can be computed, and documents can then be grouped in spatial clusters indicating document similarity.

Page Rank System

In addition to the weighted keyword techniques described above, Google uses a system called “Page Rank” to determine the relative importance of a document based on the number of citations a document has. So while weighted keyword techniques can tell us what a document is about, reference counting can provide us with a popularity metric, which translates loosely into how authoritative a source the document is. Here is how the Page Rank system works, according to Google:

“We assume page A has pages T₁...T_n which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:

$$PR(A) = (1 - d) + d(PR(T_1)/C(T_1) + d(PR(T_2)/C(T_2))... + PR(T_n)/C(T_n))$$

Equation 2: Page Rank System

Note that the PageRanks form a probability distribution over web pages, so the sum of all web pages' PageRanks will be one.

PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web.” [BP00].

The main element being analyzed by this system is the number of inbound links to a given page. This score is adjusted by the relative popularity (Page Rank) of the inbound links further diluted by the number of pages that the referring pages link to in addition to the page under consideration.

Because text is a language we use all of the time for describing all manners of concepts and objects, we can mine the subject matter from text documents

without much difficulty. We understand how language works, and we have developed a few great tricks to tease the meaning out of documents.

Non-Text document retrieval

Image attributes

In the digital image domain, visual content analysis has been the most widely studied approach to document search and classification. Many systems attempt to index image files by attribute histograms such as color or texture distributions. Images can then be categorized and grouped based on the similarities of these attribute histograms. Searches can be performed to find the closest matching histogram. Although perceptually similar images can be grouped using such systems, the resultant groups rarely share subject or content items. Additional research has been addressing regional pattern matching and contour matching approaches to visual content analysis [SM00].

Context analysis

Completely automatic annotation methods have been obtained from approaches that focus on document context. In these approaches, a document is characterized by factors such as the document creation time, the document author, and documents that appear in the same location. An example of such a system is Google Image Search, which uses the text of the web page in which an image file was located as a source for keyword search terms. More subtle context data is used by AutoAlbum, software that attempts to auto-correlate document creation time with image analysis signatures [PCF02].

2.2.3 Keyword expansion techniques

A common technique for indexing non-text documents is the use of keyword tags. Document annotation is probably the best current approach to generating searchable indices for multimedia documents. One problem with such systems is that the annotation procedure is time consuming, and is often neglected

completely. The other problem with keyword tagging of multimedia documents is its hit-or-miss nature. Often, document retrieval techniques that rely on keyword strategies require an exact keyword match. Keyword expansion is a technique where one keyword is associated with many search terms, and has been used in the past to avoid the hit-or-miss nature of keyword matching.

Previous architects of the Treehouse Studio have implemented keyword expansion techniques to improve document retrieval for searches conducted within the Treehouse framework. In 2003, James Dai constructed a system that allowed users to annotate images and parts of images. The keywords entered by users had their meanings extended by consulting a word taxonomy and commonsense database. Using WordNet [Fel98] and the OpenMind system [Sto99], Dai developed a system which was able to expand the keywords used in query matching [Dai03].

2.2.4 Systems that leverage user communities to add meaning to documents

Metcalfé's Law states that as the utility and value of a network is approximately proportional to the square of the number of nodes on that network [Met96]. The availability of high-quality Internet connections has grown, and as a result user communities have become an asset not just for content consumption, but also for content development. Perhaps the first widely publicized example of this kind of productive community on the Internet was the Linux community, where large numbers of end users came together over the Internet and collaboratively created an operating system. Since then, numerous examples of user communities creating content and adding value have only grown.

Social filtering networks

In the early days of Usenet News and World Wide Web forums, user communities came together around various topic areas. In many cases, user groups and communities discussed and reviewed products and services.

Frequently, communities with shared, common interests in film, music, or entertainment would exchange their opinions or recommendations.

One of the first research projects to take note of this and attempt to harness and study the power of these social networks was the GroupLens project [RIS94]. In this system, servers attempt to predict which users will be interested in which Usenet articles, based on heuristics about their past preferences and the preferences of other users whose opinions seem to be similar to their own.

Research on social filtering continued at many institutions, including the MIT Media Lab. The technology was commercialized by Firefly, and is used today in systems such as Amazon.com's recommendation system [ama05].

Wikipedia

Currently one of the most talked about grassroots websites is Wikipedia, a website that harnesses the vast number of Internet users to create the most complete online encyclopedia in the world. Just as Linux continues to be developed by thousands of volunteer software engineers from all over the world, Wikipedia is also able to harness the goodwill of many thousands of volunteers to create a tomb of knowledge that is constantly growing and being updated.

“Wikipedia is a wiki, a website in which any visitor may edit its articles and have their changes be instantly displayed. Articles are not controlled by appointed users or special editors, and its volunteer authors warn that contributions will be "edited mercilessly" collaboratively. Its authors need not have expertise or formal qualifications in the subjects which they edit, a point of criticism discussed below. Decision-making on Wikipedia is most often done by consensus, with edit wars often occurring over controversial articles. Articles are always subject to editing, such that Wikipedia does not declare any articles finished.” [wik01]

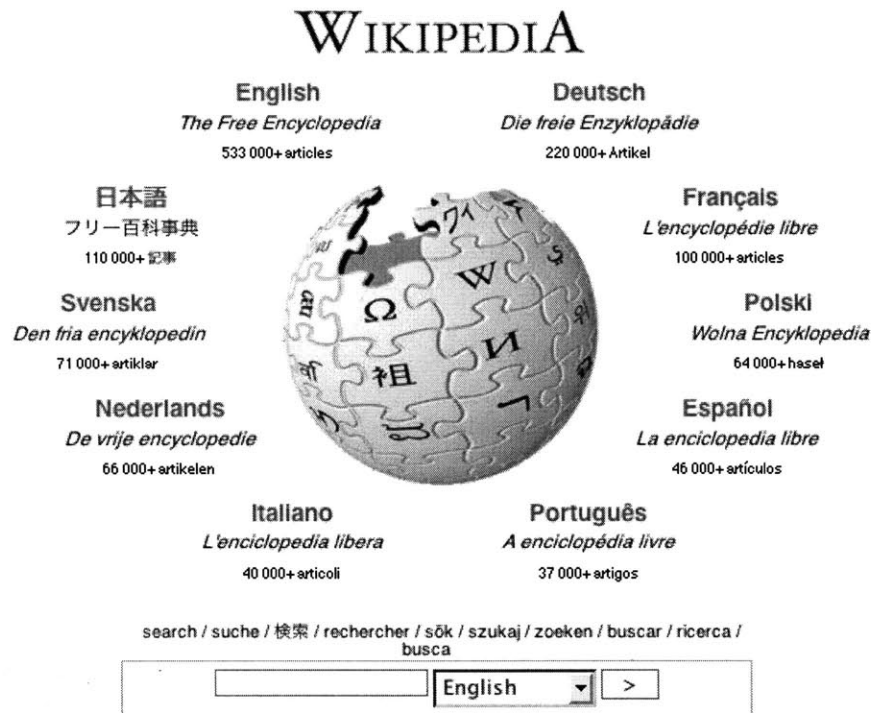


Figure 4: Wikipedia.

The key innovation of systems such as Wikipedia is that they are able to leverage large numbers of people to work on a problem without the use of a huge bureaucracy, and without creating useless or chaotic content. Although Wikipedia has been strongly criticized for its lack of accountability and reliability, there can be no doubt of its general usefulness. Certainly Wikipedia does have its share of vandals and sloppy article writers, but the community is able to fix such problems almost as soon as they appear. Ironically, the same open policy that makes it easy to vandalize a page makes it easy to repair. If the majority of the visitors to the Wikipedia website want a usable service, and they are empowered to repair vandalized pages, then most likely vandalism will be repaired as soon as it is encountered. This same sort of argument could be made about incomplete or incorrect articles.

What Wikipedia definitely gets right is the breadth of knowledge it has harnessed. Because such a large number of people are working on the project,

there are entries in Wikipedia on just about every subject, in a dozen or more languages. More questionable is how useful this information is. Consensus seems to imply that the data is accurate enough for casual fact-finding, but that using the information for serious research is problematic. One reason for this is that Wikipedia is a living document and is constantly undergoing revisions. Another is the general accountability problem mentioned earlier.

Tagging and Folksonomy systems

The idea of keywords used for classification of documents or data is nothing new. A variety of examples of keyword classification have already been given. In simple keyword annotation, a document has one or more ‘keyword’ attributes. This is a piece of meta-data that helps to classify the subject matter of a document. Typically, the document author creates the keywords for a document, or in some cases by a special agent whose job it is to classify documents. In either case, a document has one set of keywords; searching for documents, then, is a simple matter of searching through a keyword index and looking up matching documents.

Keyword annotation can be a very valuable tool for document retrieval, but oftentimes documents are never classified in this way. Sometimes that is a result of the underlying storage technology not having support for such meta-data, but even when such tagging capabilities exist – say in photo organization tools like iPhoto – the meta-data is never entered. Some websites are harnessing the power of large communities to add meta-data to documents. In bringing document classification tasks to the masses, a new kind of keyword tagging has been introduced, something that Thomas Vander Wal has termed a “Folksonomy” [Wal05].

Folksonomies are collections of tags that describe an object. Tags are the Folksonomy communities’ name for a keyword. We think of taxonomies as hierarchical categorizations of objects. Folksonomies are non-hierarchical, and

non-authoritative. They are grassroots, populist collections of tags used to define an object. There are two types of Folksonomies: narrow and broad.

Thomas Vander Wal describes a Narrow Folksonomy as a system where every object has only one collection of tags that are created and edited by every member of the community. Broad Folksonomies, in contrast, have one set of tags per user of the system. In such systems most documents have many thousands of keyword sets associated with them. Because each user has her own set of tags for each document, it is likely that the document has the exact same tag from multiple users. In a broad Folksonomy system, most users would probably tag a picture of a red apple with the word "apple." For document retrieval this is interesting not only because it is a decentralized, grassroots annotation system, but also because it provides us with a democratic keyword weighing system: we can simply count the occurrences of tags to establish a weight metric.

There are many variations on these Folksonomic organizational tools on the World Wide Web. We will present a few examples of these systems here, discussing what novel approaches or benefits each project provides to its user community.

The ESP Game

This website is a 2004 Carnegie Mellon University research project which attempts to label all the images on the web by getting web users to play a game. The game works like this: users log in, and are randomly assigned a partner. The game begins with an image appearing on the screen, and the partners must agree upon possible tags for the image. They are given a fixed time, and must identify as many tags as possible in that time. Both partners must type the same tags, without seeing each other, what their partner is typing, or the ability to negotiate in any way. If both partners choose to label the image with the same

tag then that tag is added to the image and the players get a point; the players are then shown a new image, and the game continues until they run out of time.

“Labeling an image means associating word descriptions to it, as shown below. Computer programs can't yet determine the contents of arbitrary images, but the ESP game provides a novel method of labeling them: players get to have fun as they help us determine their contents. If the ESP game is played as much as other popular online games, we estimate that all the images on the Web can be labeled in a matter of weeks” [Esp03]

By using teams in this way, the ESP Game cuts down on any chances of vandalism, and also cuts down on strange or inappropriate tags for images. This is certainly a novel way to harness the power of a multi-user community. It is different from the other methods listed here, in that the reward the users are deriving from participating is not directly related to the success of the labeling or the general usefulness of the website. The users are being harnessed, but they are not necessarily being trusted or consulted, or charged with developing the content. It should also be pointed out that though this is a labeling system, it is not really a Folksonomy. Images in this system have only one set of tags that cannot not be changed once they have been entered.

Flickr

Flickr is a social photograph-sharing website, where users upload their personal photographs to share with friends and the rest of the world. Like Friendster¹ and other social networking websites, it allows the user to create groups of associates, friends, and family. You can then choose to share your photographs with just your family and friends, or with the world at large.

¹ A popular website in 2003 for visualizing your social network and meeting new people.

For the most part, in Flickr users annotate and tag their own photographs and do not let strangers add tags to them. The author may allow people to add captions and keywords to their photographs, but most people do not allow this. In addition, in Flickr there is only one tag-set per document, so when someone adds a tag to your image, it is there for all to see. Because there is only one set of keywords per photo, Flickr is considered to be a narrow Folksonomy.

“Once you make the switch to digital, it is all too easy to get overwhelmed with the sheer number of photos you take with that itchy trigger finger. Albums, the principal way people go about organizing photos today, are great—until you get to 20 or 30 or 50 of them. They worked in the days of getting rolls of film developed, but the "album" metaphor is in desperate need of a Florida condo and full retirement.

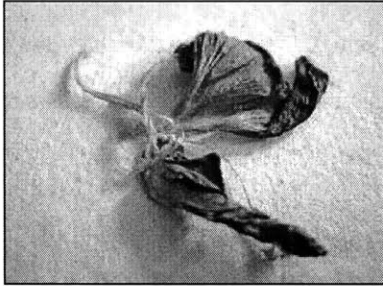
Part of the solution is to make the process of organizing photos collaborative. In Flickr, you can give your friends, family, and other contacts permission to organize your photos—not just to add comments, but also notes and tags. People like to ooh and ahh, laugh and cry, make wisecracks when sharing photos. Why not give them the ability to do this when they look at them over the Internet? And as all this info accretes around the photos as meta-data, you can find them so much easier later on, since all this info is also searchable.” [Fli05]

Flickr has managed to add incentives that cause its users to publish their images with good keyword tags. The reason for its success seems to be that people want to have their images looked at, and in order to facilitate that, the users have taken the time to add tags to their work. Perhaps it is the pride in one’s work when it is going to be possibly published for the world to see that causes the authors to take the extra step of adding useful tags.



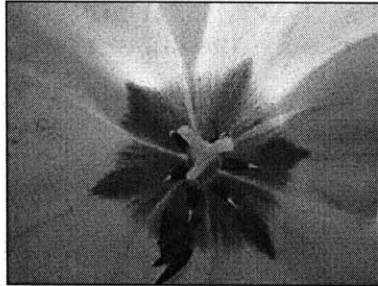
johnmaeda's photos pro
johnmaeda is a friend. ([Change](#))

IMG_248



out-of-pocket colors [simplicity]
 This photo is public. (0 Comments)
Uploaded on [Apr 26, 2005](#)

IMG_0230



aesthetic vegetables [simplicity]
 This photo is public. (1 Comment)
Uploaded on [Apr 26, 2005](#)

- » [Sets](#)
- » [Favorites](#)
- » [Profile](#)

Search his photos

SEARCH

Or, browse with...

- [His tags](#)
- [Calendar](#)
- [Archives \(94 photos\)](#)

Figure 5: Flickr.

All time most popular tags

africa amsterdam animal animals april architecture art austin australia baby
barcelona beach berlin bird birthday blackandwhite blue boston bridge building
bw california cameraphone canada car cat cats chicago
china christmas church city clouds concert day dc dog dogs easter england
europe family february florida flower flowers food france friends
fun garden germany girl graffiti green halloween hawaii holiday home hongkong
house india italy january japan kids lake landscape light london losangeles
macro march me mexico mobile moblog mountain mountains museum music
nature new newyork newyorkcity newzealand night nyc ocean old orange
oregon paris park party people phone photo pink portrait red reflection river
rome sanfrancisco school scotland sea seattle sign sky snow spain
spring squarecircle street streetart summer sun sunset sydney taiwan texas thailand
tokyo toronto travel tree trees trip uk unfound urban usa vacation
vancouver washington water wedding white window winter work yellow zoo

Figure 6: Flickr tags.

Another feature of Flickr is that it publishes complete programming documentation for their website, along with software tools to allow random hackers all over the world to write Flickr applications, applications which use the Flickr site more as a data store than as an actual website. It offers libraries for various programming languages such as perl and python, to do things like retrieve 10 of the most recent images with the tag "sun," for instance.

del.icio.us

del.icio.us is another community tagging site. In this case,, the objects that are being tagged are websites. del.icio.us allows every user to save her own set of tags for any given website. When someone searches for websites using tags, she can choose to search just her own tags (her private keywords), or to search everyone's tags. Because each user has a unique keyword list for each document, del.icio.us is said to be a broad Folksonomy.

"del.icio.us is a social bookmarks manager. It allows you to easily add sites you like to your personal collection of links, to categorize those sites with keywords, and to share your collection not only between your own browsers and machines, but also with others.

Once you've registered for the service, you add a simple bookmarklet to your browser (see below). When you find a web page you'd like to add to your list, you simply select the del.icio.us bookmarklet, and you'll be prompted for information about the page. You can add descriptive terms to group similar links together, modify the title of the page, and add extended notes for yourself or for others.

You can access your list of links from any web browser. By default, your links are shown to you in reverse chronological order, with those you've added most recently at the top. In addition to viewing by date, you can also view all links in a specific category (you define your own categories as you add the links), or search your links for keywords.

What makes del.icio.us a social system is its ability to let you see the links that others have collected, as well as showing you who else has bookmarked a specific site. You can also view the links collected by others, and subscribe to the links of people whose lists you find interesting." [Del05]

Vandalism, lack of accountability, and other fears

Community-developed sites such as those mentioned above all suffer similar criticisms. The very fact that these sites rely upon the good nature of their members is often treated as a criticism, even though this trust is the very thing that makes these sites strong and vibrant. Critics point out that because these sites are managed and created by nearly anonymous users, their content cannot be trusted, or treated as an authoritative source. In the case of community tagging sites, critics have pointed out that there is an uncontrolled vocabulary of the tags that can result in ambiguity, and that there is no accounting for synonyms. While it is true that Folksonomies will suffer from these problems, they still seem to function quite well.

Similar complaints of accountability are aimed at Wikipedia, and the site acknowledges that there is a potential problem with trusting individual editors to craft respectable and accurate entries. The Wikipedia community is quick to point out that it is never the case that one individual is editing an article, a very large community edits all articles continuously.

2.2.5 Systems that observe the user

Henry Lieberman has done extensive research into agent systems that observe the user interface [Lie97]. Often these agents attempt to cooperatively assist users as they are learning how to operate the user interface. In some examples, such as Letizia, the agent observes the user by noticing changes in the user interface, and then extrapolates information about the user's intentions, and tries to assist the user from there. Observation of the user interface can also be a boon for document categorization.

Aria, a software effort by Lieberman [HEP01], is an example of a software system that observes user interaction to improve document retrieval. This software implements an agent that observes users as they compose email messages. As the user types an email message, the agent suggests possible

images to accompany the text. It does so by extracting keywords from the typed message in real time. If no matching images suit the email, and the user wishes to attach some other image, the system will automatically keyword-annotate the new image with keywords from the email message. This system is similar to techniques used by Google Image Search, which uses the textual context in which images occur to define keywords for images. The improvements in the Aria system are that the system is used both to suggest images for inclusion in a just-in-time retrieval application, and that to some extent document classification and retrieval are seamlessly unified into one set of user gestures.

Chen, et. al. [CLS02] have done work in observing how users traverse websites, in order to suggest broad categorizations for documents. They term this classification framework "Access Based Categorization." This system requires a fixed and pre-existing categorization system in use on a particular website. In their example they use a consumer ratings site, where users navigate consumer items in a hierarchical structure. The basic proposition of this framework is that by observing a user's browsing history through the site, the system can make predictions about which categories a user will visit. The system classifies new items by correlating the actions of a user with predictions about what the user is shopping for, based on the users browsing history. This user-observation model requires a fixed set of document categories, and continuous observations and predictions about user paths.

2.3 Data storage principles

Digital data needs to be stored in a readable, possibly re-writable storage medium if it is to be used more than once. Many strategies for storing data have been suggested and implemented over the years. Inherent in a storage mechanism is some minimal retrieval mechanism; it is not surprising, then, that storage format and mechanics eventually play a limiting factor in what types of search and retrieval processes are possible in a given time frame. What follows is a description of several data-storage techniques used in diverse applications.

Observations of the techniques' major advances and unique attributes are discussed.

2.3.1 Traditional file systems

Traditional file systems, such as those used by personal computers, rely upon unique file names to identify document data. We think of these name spaces as file paths. Typically, there is a one-to-one mapping between a file path and some document data. A path is nothing more than a text label which uniquely identifies a digital document. Some minimal meta-data may also be stored with the file as well, but we have little information available to us other than the file's data and its name. Beyond this simple namespace hash, there is typically no other useful document index in such a file system. The greatest advantage of the file system is that the file format is completely flexible. Files can contain any type of data, representing any media type.

File Data

Most file systems in use in personal computers rely upon naming conventions to describe the contents of a file. We think of the first part of a file's path as the file's 'location.' In fact, this text tag is parsed out traditionally as a set of containing folders or directories. We can think of the front part of the path as a single containment hierarchy. Typically, each file belongs to one and only one taxonomy of folders. This gives us some sense of where a file is located. Again typically 'similar' files are grouped in similar containers, but the naming convention is weak. Another convention – assigning a three-letter extension to the file's name such as “.txt” or “.pdf” – gives us some idea of what kind of data is contained in the file. These conventions are well understood by individuals who are conditioned to think as their operating systems dictate, and are helpful in that respect, but more could be done. We see that file-naming conventions alone are able to give us some understanding of the file's contents and hierarchical location. For most personal computer users, this is the only method

for organizing documents, and it is completely inadequate. Some operating systems support 'symbolic links' or aliases to files. These enable a file to belong to multiple container hierarchies, but this feature is often underutilized.

Meta-data

In most operating systems, some information about when a file was created, when it was modified, and who may have edited the file are available as file meta-data: information about the file that is stored separately from the actual contents of the file, or the file's data. For the most part, this data is minimal, but might allow a user a few additional organizational options, such as sorting and searching for files based on their creation and modification times.

Unfortunately, the file system rarely keeps a live index of the files based on this meta-data. Instead, the operating system must typically do a live search on non-indexed data if the user is searching for files created on a specific date.

2.3.2 Databases

Databases represent a richer format for storing long-term data than file systems. Typically, information from databases can be retrieved quicker because many indices on fields have been created. Many database systems sacrifice flexibility for indices and search performance.

Flat databases

The simplest example of a database storage mechanism is something called a flat database system. In this system, data is represented as records in rows, fields, and columns, just as in a spreadsheet. In this two-dimensional array, each cell that is an intersection of a row and a column represents a field of a record. These values in the cell are literal values that are used to build indices. Searching records based on columnar criteria that have been indexed is a simple and efficient solution to text data search. Flat databases are immediate in that each cell contains a literal value which can be easily sorted, searched, and read.

Relational databases

Relational databases were first described by E.F. Codd in 1970 [Cod70]. A relational database contains multiple tables, each similar to the one in the "flat" database model. Tables are related to each other using keys. Keys are simply columns of a table that reference index values of another table. Two tables that can be joined by linking key columns are related. Data from one table can be used to build upon the data of another. Relational databases create one or more index on each table. The result is that relational databases are very easy to search, and quite flexible.

2.3.3 Geographic information systems

Geographic Information Systems (or GISs) are spatial databases. GIS attempts to bring the advantages of structured query and relational databases to the world of cartographic or spatial data, such as can be found in mapping and planning contexts. The first actual GIS software package was probably the Canadian CGIS developed by Roger Tomlinson in 1967, for use by the Canadian Department of Energy and Mines, and Resources. Geographic Information Systems represent an example of a database system that is specifically tailored to one problem domain, in this case spatial data types. Geographic Information Systems are listed here as relevant background technology for two reasons: they illustrate that in many instances it makes sense to use a specific database format that is closely coupled to a specific problem domain; second, spatial queries are unique in that they allow the user to ask questions in terms of proximity, density, distribution, and containment. An example query in a GIS might be something like "Show me all the schools within two miles of a river within the state border of Vermont." [NW79] [GIS05]

2.3.4 Relational file systems

Because relational databases have proven so useful in many areas of information retrieval, there are efforts underway to implement relational structures at the

operating system level. While there are several such efforts under way – including efforts by Apple, and from the open software movement – we will be describing only one system, proposed by Microsoft.

Microsoft has been working for some time on a file system to solve the document-retrieval problem by providing a very flexible document-storage file system supporting extended rich meta-data [Gri04]. In their vision, the operating system allows the user to search for documents by attributes specific to the type of file the user is looking for. For example, if you were looking for images, you could specify the shutter speed for matching documents. This type of search is supported at the file-system level, because the file system supports many document schemas for many different file types, which break out specific document properties. This kind of rich meta-data support would be welcome in any future operating system.

2.4 Findings from historical work

The purpose of this section was to briefly reexamine the long history of document retrieval, highlighting key developments and unique approaches to storing and finding information. Simple document indexing of cuneiform tablets led eventually to more usable catalogs based on logical classification frameworks such as the Dewey Decimal System. In the digital age, computational power was brought to bear on the problem of document indexing. In the modern era of ubiquitous, media-rich communication, current trends are pointing to specific advantages of decentralized non-hierarchical approaches to information management. Folksonomies and faceted classification systems are currently garnering lots of attention (good and bad) in the communities of library science and information management.

Many of the salient ideas described above have also been borrowed from for the development of the Interface Harvesting software presented in this thesis.

Specifically, the classification framework of the Dewey Decimal System was inspirational because it presented a single metric for document affinity, and suggests that a spatial metaphor between related documents would be appropriate. Similarly, sites like Flickr have demonstrated that user communities can be motivated in such a way as to provide a reliable subject classification of non-text documents. Finally, for the distributed file system required by the software, we identified the need to be able to quickly search through a large number of non-text documents for arbitrary features. This led us to the development of the Flexible File System, which is essentially a very flexible relational file system.

3 Method

3.1 Observations

3.1.1 Real-world organizational observations

Observations about document management and retrieval in the physical world have been used in the past to inform the design process of user experiences of an operating system with virtual files. Metaphors have been found to be useful when introducing users to new software [Sza95]. The desktop metaphor established by Xerox Parc and later by Apple Computer is the primary example of this. The desktop metaphor models the user experience of browsing digital assets to the organizational structure of a desk. Apple was able to deploy the desktop metaphor in order to allow novice computer users to feel comfortable manipulating files. Culturally we are so used to this metaphorical device now that we may forget its origins, and its significance.

In addition to providing a familiar framework for thinking about documents and files, there is some research that claims that the placement of icons in specific locations on the virtual desk surface may function as a memory trigger [Mal83], providing the user with a spatial modality for document organization, an organizational principal with which they may already be familiar. Many studies

have been conducted to analyze the utility of spatial organization of files in order to enhance document recall. Some results seem to support spatial recall [CMK91], while others refute it [DJ85].

Despite the above-mentioned human factors studies that attempt to ascertain if users' personal memory recall is improved by spatial clues, no studies have attempted to record document location information for the purpose of operating-system analysis or categorization of documents based upon the virtual locations where users have chosen to place their files.

Even though the question of the utility of spatial organization is not fully resolved, it has been well-established that metaphors are genuinely useful in guiding the design of new user interactions and experiences. The first physical system that was investigated in order to inform the Interface Harvesting application was the photo album. Photo albums are sets of images that have been arranged into a book form. These books are generally thematic, representing discrete events in our lives. It is quite common to see a photo album that relates the story of a wedding, or of a vacation.

Another observation about photo albums is that a user must manually curate them. The user selects the images to represent, and the order in which to present them. Considerations of chronology, subject matter, and layout all figure into the curatorial decisions of the photo album creator.

3.1.2 Virtual file organization techniques

3.1.3 Image placement study

In order to examine how people arrange documents, six participants performed the following informal exercise:

Subjects were told that they were taking part in a very informal exercise, in which I was examining, among other things, what kinds of organizing principles people use when sorting collections of photographs.

Each subject was asked to grab a large background image sheet from a selection of twelve images. The background images were similar to placemats, two feet wide and one foot tall. These placemats were images printed onto a thick, glossy paper stock. Some of the images were photographs of landscapes, supermarket shelves, and city street-scenes. Other images were more abstract and illustrative, rather than photographic. These images included things like regular grids, concentric rings, and collections of circles and squares.

After choosing a background image, the users were handed a stack of miniature photographs, each measuring about 1.5" wide. These photographs were images from my personal photo collection and were well mixed up. Each of the six participants was asked to choose about twenty photographs. There were over 140 photographs to choose from.

After participants had chosen the images with which they wished to work, they were asked to place them on the placemats in any sequence, arrangement, or composition they so desired. They were given approximately 5 minutes to complete this image-placement task.

When the images had been placed, participants guessed what principles the others used to organize their images. After we made our guesses, the participant explained her decision process. Strategies for image organization varied wildly. It is worth presenting some of the strategies here, enumerating the various strategies for image classification that were used. Regardless of the strategies used in guiding the image placement, in every case 'similar' images were placed next to each other. What 'similar' means is different in each instance because the curatorial decisions differ; however, the similarity is plain to the viewer. In each case, the nature of the background image also affected the image placement procedure, and influenced the curatorial decisions.

Organizational practice #1: Context placement in image landscape.



Figure 7: Organizational practice #1

This participant attempted to place the miniature images on the photographic context in such a way that the image logically 'fit' there. For instance, images of windows were placed on buildings; images of gauges were placed on top of machines. In some instances the small photographs blended into the background scene, looking very natural in their new habitat.

Organizational practice #2: Narrative placement of images on the landscape.

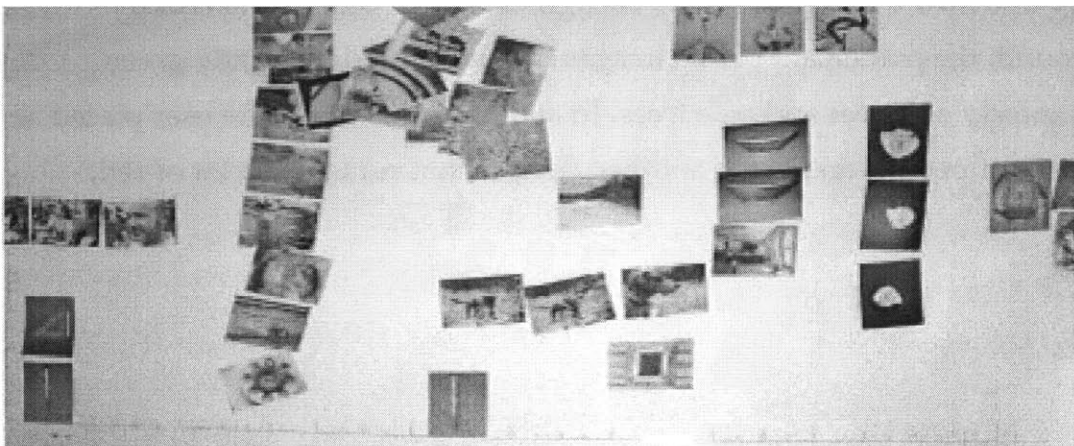


Figure 8: Organizational practice #2

Given a relatively featureless image consisting only of an unlabeled X-axis and unlabeled Y-axis, one participant treated the background as a context in which to tell a comic-book-like story, using the smaller images as sequential narrative frames.

Organizational practice #3: Grouped “related” items.

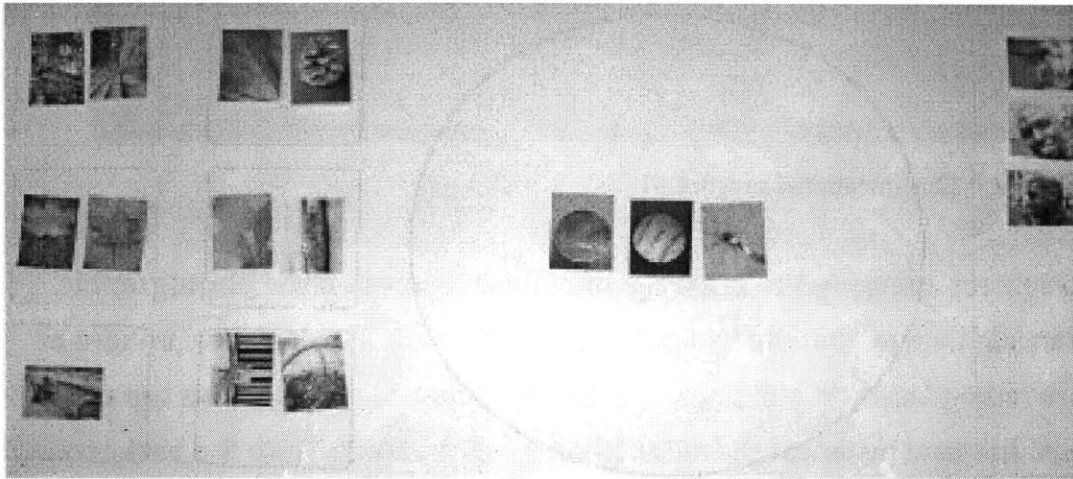


Figure 9: Organizational practice #3

In this case, the participant used formal features of the background image to create logical divisions, pockets in which to place the smaller photographs. For instance, round items such as wheels, balls and bowls were placed together inside of a circular dotted line on the background. In another section of the background, the participant placed images that were predominantly green, images mostly of leaves and pine trees. In another square box, the user placed all the images of people, and in yet another, images that contained a lot of red.

Organizational practice #4: Continuum in one axis, organic to inorganic.

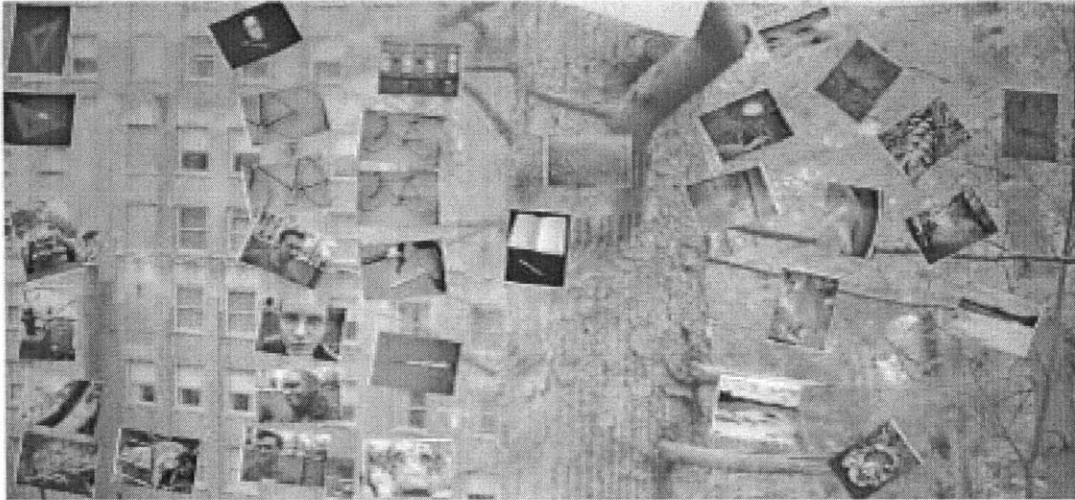


Figure 10: Organizational practice #4

This participant was presented with a background image that was a blend between two photographic images. This mat convolved an image of the woods with an image of downtown Manhattan. This image was continuous, and the participant placed images of inorganic objects such as rusty mechanical equipment on the urban side of the continuum and photographs of leaves and acorns on the rural side of the context image. In the middle were placed images of people.

Organizational practice #5: Multiple Axis on a photo: Size / Man-made subject matter



Figure 11: Organizational practice #5

One of the more interesting results was a context that had an image of a grocery store freezer section as a background image. The participant used two criteria when placing images on the placemat. The first attribute to be mapped was the size of the object in the photograph. Small objects were placed lower vertically on the map, so that at the very bottom were items such as pens and acorns. Towards the top of the placemat were placed images that were larger, such as images of buildings. Vertically, in the middle, were approximately human-sized objects such as bikes and cars. The horizontal organizational principle was what the participant described as 'man-made-ness.' Objects that were produced by people were positioned to the right; objects that occur naturally were placed to the left. Photographs of people were placed in the middle.

3.2 Software design

3.2.1 Goals of treehouse project

The Treehouse Studio is an ambitious project of the Physical Language Workshop under the direction of John Maeda. The project aims to establish an infrastructure that supports the grassroots development of an online community

for digital creativity and commerce. To this end, the Workshop is charged with the task of developing the back-end and client-side technologies that will enable the document sharing, document versioning, multimedia authorship, project management, commerce transactions, and document-retrieval tasks required by such a community. On the server side, we are developing a framework to handle user authentication and file management. On the client side, it means developing a framework for cross-platform application delivery via the World Wide Web. Because the goals of this project are so large, we have given careful consideration to the Treehouse Studio as both an application-delivery platform and as a distributed-document storage facility.

3.2.2 Treehouse Studio Faceted File System

3.2.3 Relational database storage

The Treehouse uses a flexible file system that uses a relational database for document storage. The database scheme was designed with the following goals in mind:

- 1. File meta-data of all kinds should be indexed for speedy search;**
- 2. Applications should be able to quickly fetch only pertinent parts of documents in a relatively short time;**
- 3. Applications should be allowed to add their own meta-data fields to documents without requiring a redesign of the database, or degrading database performance;**
- 4. All documents should support ad hoc keyword annotation and tagging by multiple users;**
- 5. Document versioning should be inherent in the database design;**
- 6. The API for writing to files and adding meta-data should be very simple for developers to use.**

The Treehouse Studio file system is now in its second revision. The first revision, completed last year, failed to address several of the key design goals. In

particular, the first design did not allow for arbitrary meta-data fields to be added in an ad hoc manner, and it had a very complicated API. The new file system, called the Faceted File System (FFS), is by comparison quite easy to use, and very flexible.

The FFS uses two main tables for document storage. The first table is the Document Table. This table stores all mandatory file attributes. There are only a small number of mandatory attributes, those which are most commonly used in all file systems. This table contains a `document_id` field, which is a 32-byte unique identifier for this file. Similar to the ISBN number in the book world, this identifier is designed to be unique across servers, as we anticipate that the FFS will become a distributed file system in the future. In addition to the ID field, this table also stores the document's Parent ID, in the event that this is a version of an existing file. If the Parent ID value is null, it is assumed that the file has no parent, and is at the head of any version tree. The table also contains columns for document creation time, modification time, access time, document type, and document data. The document data is where arbitrary document content can be dumped, such as the pixels of an image.

The second Document Table is called the DocumentProperties Table, and is designed for speed and flexibility. There are only four fields in the DocumentProperties table; they are `property_id`, `document_id`, `property_name`, and `property_value`. The DocumentProperties Table functions as a hash table of properties and values for all documents in the file system. All of the document meta-data and as much of the document data as the developer would like can be stored in this table. Indices exist for all columns. In the current implementation, all hash keys are generated in full at the application level. Ideally, there would be something in the API that would prevent namespace collisions for these hash keys.

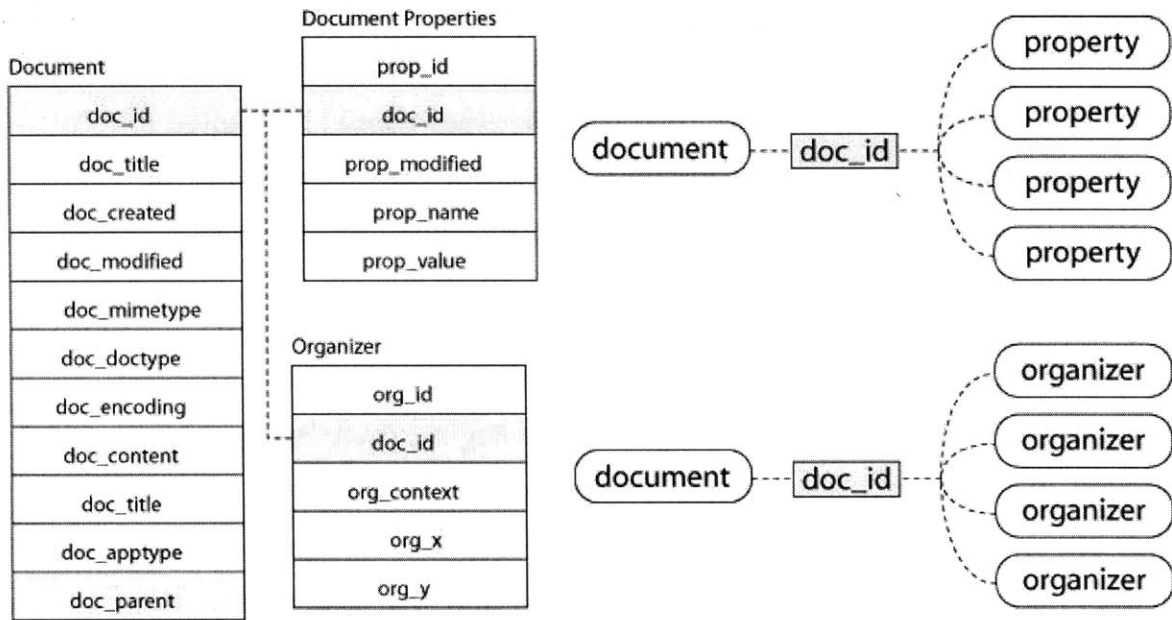


Figure 12: Database representation of the Faceted File System

Searches across the FFS system will be much faster than searches on traditional network-mounted volumes such as NFS or AFS. In a typical network file system, the searching algorithm is executed on the local machine, while the data being searched resides across the network. Such a search procedure is very network intensive, requiring constant communication from the client to the server. In the case of the FFS, all the searching takes place on the remote server, so there is virtually no network traffic when executing the search.

The architecture of the FFS tables makes document versioning and document history very easy for developers to implement. Because each document in the database retains a pointer to its parent document, versioning is simply a matter of copying the document ID to the parent field, and then assigning a new unique ID to the document. By traversing the parent tree, a document's entire life cycle can be described.

Because the Document Properties Table is fully indexed, it is possible to get fast search results by querying document properties. This enables searches such as finding all documents that have a property named "color" which is set to the

value “yellow” to execute quite quickly. In the current implementation, the value fields are represented in the database as simple strings. In future versions of the file system, a separate table for numerical properties should be created to accommodate searches such as finding all the documents which have the property named “number of views” set to some value greater than 14.

Because the file system is based on an industry standard Postgresql implementation, it is possible to develop distributed applications using the mirroring features of the underlying database engine. Enabling mirroring features in the file system makes clustered and distributed server topologies possible, which could improve the performance and redundancy characteristics of the system, allowing the file system to grow and scale seamlessly.

The tables of the FFS were designed specifically with Folksonomic tagging schemes in mind. One of the design goals was to support ad hoc tagging of documents. Applications can define their own meta-data on the fly, thereby enabling searching and categorization on document-specific terms and attributes. This ad hoc classification mechanism extends traditional tagging because each tag can also have an associated value. Instead of merely tagging a document with a keyword such as “birthday,” users could actually define tags such as “birthday” and assign a metric value, such as “0.23”.

Because FFS uses a networked storage device, accessed through a web server, documents can be published to the web by the author with no additional work. If an author of a document decided to make a document available to the wider public, the document would immediately have a URL that would allow other people to get at the document. Similarly, collaboration between users of the system is also very simple. In many ways the FFS is a zero-configuration network file server, allowing people to collaborate on documents from anywhere in the world if they so choose.

3.2.4 Organizer software

In addition to the server database architecture and FFS, three proof-of-concept end user applications were developed. These applications were developed in order to demonstrate systems that can observe user interactions with documents, and create knowledge from those observations. These software projects are described below.

3.2.5 Image Landscapes software

Goals and inspiration

The first application was called “Image Landscapes,” and its purpose was twofold. From the user’s perspective, Image Landscapes was an innovative slide show editor and presenter, an application for sharing images and telling a story. From the perspective of the Treehouse Studio database, and from the perspective of other Treehouse Studio users, this tool was a way to encourage users to annotate images.

Originally, Image Landscapes was conceived of as a way to explore and extend the concept of the digital photo album. For these purposes, some effort was made to understand how traditional book-bound photo albums function in a narrative sense, and also how they function as document archives, seemingly without indices.

The observation was made that photo albums were bound collections of photographs that are thematic in nature. For example, a typical photo album tells the story of an event such as a vacation to Hawaii, a wedding, or a birthday. A counterexample would be a photo album full of things that are all the color red. Such a photo album would not be thematic, but would be formal, describing the attributes of the photographs, but ignoring the subject matter of the attributes. It is telling that we do not typically arrange our photo albums this way; instead, we typically arrange them with the express purpose of telling a continuous story about an event.

Authoring a photo album is a complex creative process, involving a series of compositional, narrative, and curatorial decisions. The author decides which photographs to include in the album, based on a number of criteria ranging from thematic coupling, image quality, and space constraints of the album. Additionally, the sequence and composition of the images are also taken into consideration. A photo album then is a kind of creative work, one with the ultimate goal of telling a story or documenting an event.

Once the photographs have been organized into categorical albums, storytelling becomes a matter of locating the appropriate album on the bookshelf. The albums that bind the photographs may be labeled, or have distinctive features such as a thematic image or sample photograph on them. The physical album functions as a container, and an index into a complex organizational structure.

The goal of the Image Landscape application was to create an online digital photo album tool, which captured the relevant features of photo albums in the physical world. This tool would allow users to create compelling narratives using photographic sequences and compositions. It would also serve as a means for a user to organize her photographs. In addition to these traditional functions of photo albums we wanted to capture all of the various curatorial and compositional decisions that the user made while composing her album, and utilize the user's decisions about composition and sequence to add a layer of meaning to the photographs contained in the album, meaning which could be extracted and used in other applications and by other users.

User experience description

We will begin our description of Image Landscapes from the vantage point of a Treehouse Studio user who wishes to share a series of photographs with her friends and family. The Image Landscape application is a Java application. It can be run from inside a web browser, or as a stand-alone application, possibly running in full-screen mode. In either case, it is important to understand that the

application is in constant communication with the Treehouse Flexible File Server, using http. The Image Landscape application does not run without a network connection.

Uploading image files, using THS tools

All of the Treehouse Studio tools are online and store their files online. The first thing the user must do is to take the photographs from her digital camera and upload them to the Treehouse Studio. Assuming that the user already has a Treehouse account, all that is required is for the user to log in to the Treehouse website and run the online application called "Photo-Album" (confusingly this application is not the same as the Image Landscapes application, it simply allows users to upload cameras full of digital images to the Flexible File System). The software will automatically find the photographs on the camera, and then upload the images to the user's account.



Figure 13: Importing images from the Treehouse Studio

A user may elect to select files from her local hard drive for importing into the Treehouse Studio. This is done with the same “Photo album tool.” Or, users might email photographs to the system if they choose, perhaps from a cellular phone.

If the user has not uploaded images in advance, she can upload images from her hard drive to the server from within the Image Landscape application.

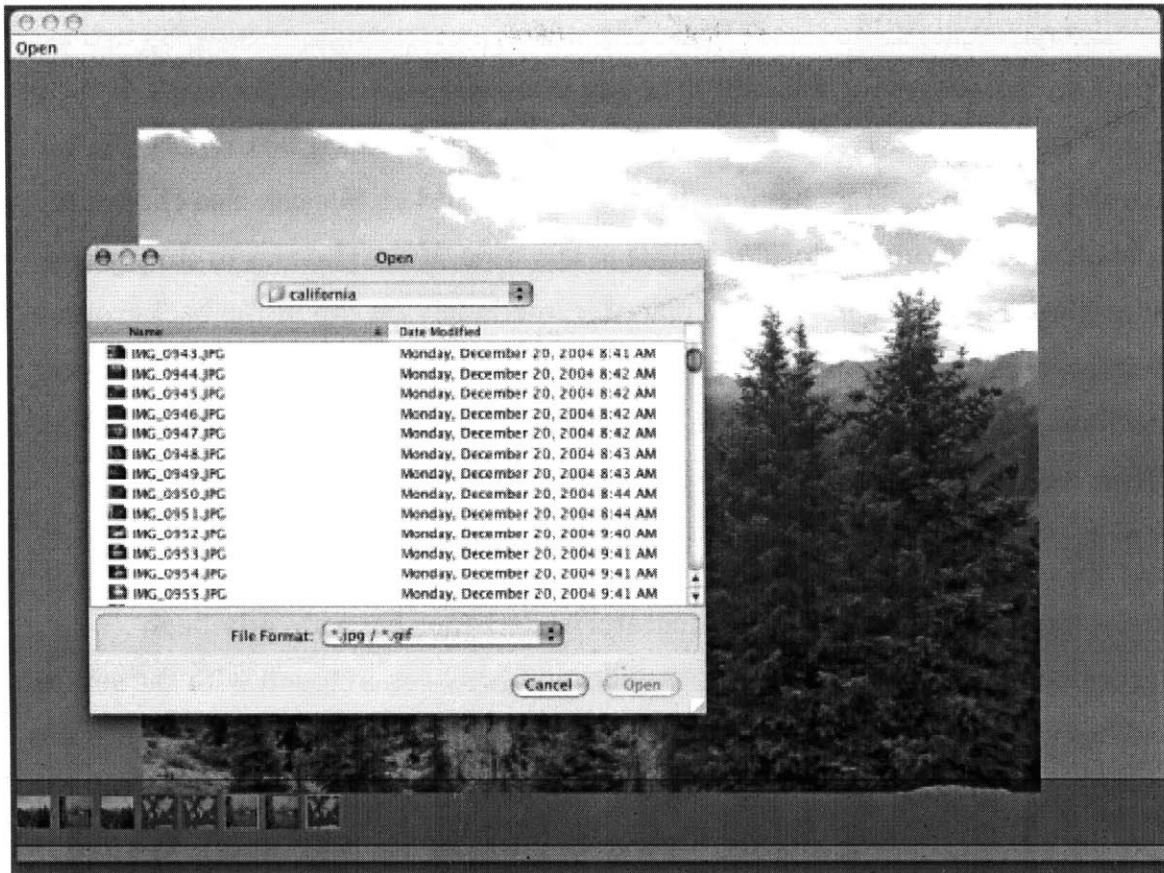


Figure 14: Importing images into the faceted file system

Selecting a background context for the album

The author first must select the appropriate background image, also called “Context” for her photo album experience. This image is like an index image, and shoulders the burden of having to characterize the entire album. This image becomes the landscape in which other images are arranged. For example, let’s say that the author has recently gotten back from a trip to the Aran Islands in Ireland. She must first pick out an image that she believes is representative for the photo album she is going to create. She could choose a photograph of the rough and rugged terrain of the island of Inishmore, or an image of green grass cropping up between the wild boulders and rock walls that dominate that landscape.

Adding photographs

There are two ways for the author to add photographs to the landscape. She can select "Import from Treehouse" which opens a search dialog box where she can search for her images by keywords, title, or date added. She can also choose to upload images from her local machine at this time. If she chooses to simply use an existing Treehouse image, it is placed immediately on top of the background landscape, as a small thumbnail. If she chooses to insert an image from her local machine, it is first uploaded to the Flexible File System on the Treehouse server. When the upload is complete, the thumbnail appears in the landscape just as above.

At this point some of the author's curatorial decisions have already been harvested by the system. The local application has been in touch with the server, and has notified the server of the author's decision to load photographs into this landscape. New tags have been created in the database, associating these images with each other. Placing images together in the same visual context is just like tagging images with keywords. Salton describes binary keyword vectors, where keywords for text documents are unweighted. These keywords either exist, or do not. The fact that these photographs have been placed together in the same photo album or Image Landscape means that they have been 'tagged' with each other. This is one simple observation the Image Landscape software relates to the Flexible File System. Curatorial decisions of the author have been harvested in order to bring meaning to some of the digital assets in the File System.

When the image is imported into the landscape, it becomes an active thumbnail. When the author mouses over the image thumbnail, the thumbnail becomes larger, and is highlighted with a dashed line. The author can then position the thumbnail into any location in the composition. The positioning of the images in the landscape is completely up to the sensibilities of the author. The author may continue to place photographs of her trip to Ireland until she feels that she is

finished. As she works, her decisions and the photo album are saved to the server automatically in the background.

As the author is moving thumbnail images around in the Image Landscape, the client application is transmitting the mouse gestures of the user in real time to the Faceted File System on the Treehouse Studio server. The database columns that store document tags and properties are being updated to reflect the current relative distances of the images in this context. These metrics are stored as pixel value differences in image location, but are used to compute a document similarity measurement, a processes which will be explained later. Where Salton proposes a virtual an n-dimensional keyword space for computing document similarity, we use an actual two-dimensional spatial metric to compute document similarity.

Clicking through to other contexts

The thumbnail images that are placed in the thematic landscape are clickable entities. If the author clicks on a small thumbnail image, that image expands to fill the entire screen. That photograph now becomes the new photographic context. In this way the Image Landscape application is a bit like hypertext on the World Wide Web. When the author or a viewer clicks through to a new context, a history bar at the bottom of the application makes a note of the last image context. A user can navigate back and forth between recently visited contexts by using the history bar.



Figure 15 Zooming into a new context

The Image Landscape author can now create multiple paths through image landscapes. It is important to note that images can be placed multiple times into the same landscape, or can be placed in multiple landscapes. This is important because it allows arbitrary graphs of linked contexts to exist. Users may create vast image landscapes to be explored by others.

These arbitrary graphs have the qualities of broad Folksonomies, in that images can be tagged by other images multiple times, and in multiple contexts. This approach is more the del.icio.us model than the Flickr model. In the current implementation the Image Landscape application, all authors use the same tagging scope, the same set of Landscapes, so that they are creating a virtual web of associations between all of the imagery in the Treehouse Studio.

Example landscapes

The author of these virtual image spaces is free to use whatever design or metaphor she would like. For instance, users have created image landscapes that are virtual replications of physical spaces; in these landscapes the virtual landscape links map directly onto the real physical proximity of the photographs. In this way, authors can generate virtual walk-through tours of locations. A completely different strategy is to use the contexts as chapters of a story, adding granularity to the traditional photo album by breaking up one large album into thematic sections. In the Aran Islands example, there might have been separate image contexts specifically for photographs of the ferry ride, the coastline, and the pubs.

Radio Frequency ID Tag Reader interface

The Image Landscape application has an optional navigation method that utilizes Radio Frequency ID tags that are attached to the back of photographs. Once an image has been uploaded to the Treehouse system, the asset gets tagged with a unique identifier in the Treehouse database. This unique identifier is like the ISBN number of books discussed earlier; using this one identifier we are able to tie this resource into multiple indexing schemes. The Treehouse Studio system also has the capability to create RFID tags that act as physical indexes to these digital assets. In this application, the user adheres the RFID tag to a printout of the digital image. This physical photograph is now a token that represents a database entry, and can now be used as an input object to the photo album application. A user may now have a number of photographic prints, all of which have unique RFID tags on them. This collection of photographs now functions as an index into the digital photo albums. If the user finds a photograph of the Eiffel Tower, and passes the photograph near the RFID reading station, the Image Landscape changes to the context of the Eiffel Tower, a context where perhaps a large number of other Parisian monuments have been placed. The RFID input tokens are not central to Interaction Harvesting, but they allow us to explore a

more natural interface to image browsing. These physical tokens act like the indices of scrolls, the cuneiform marks on the side of tablets, or the covers of books: they provide us with a superficial peak at the document, a summary of the digital Image Landscape “contained” within.

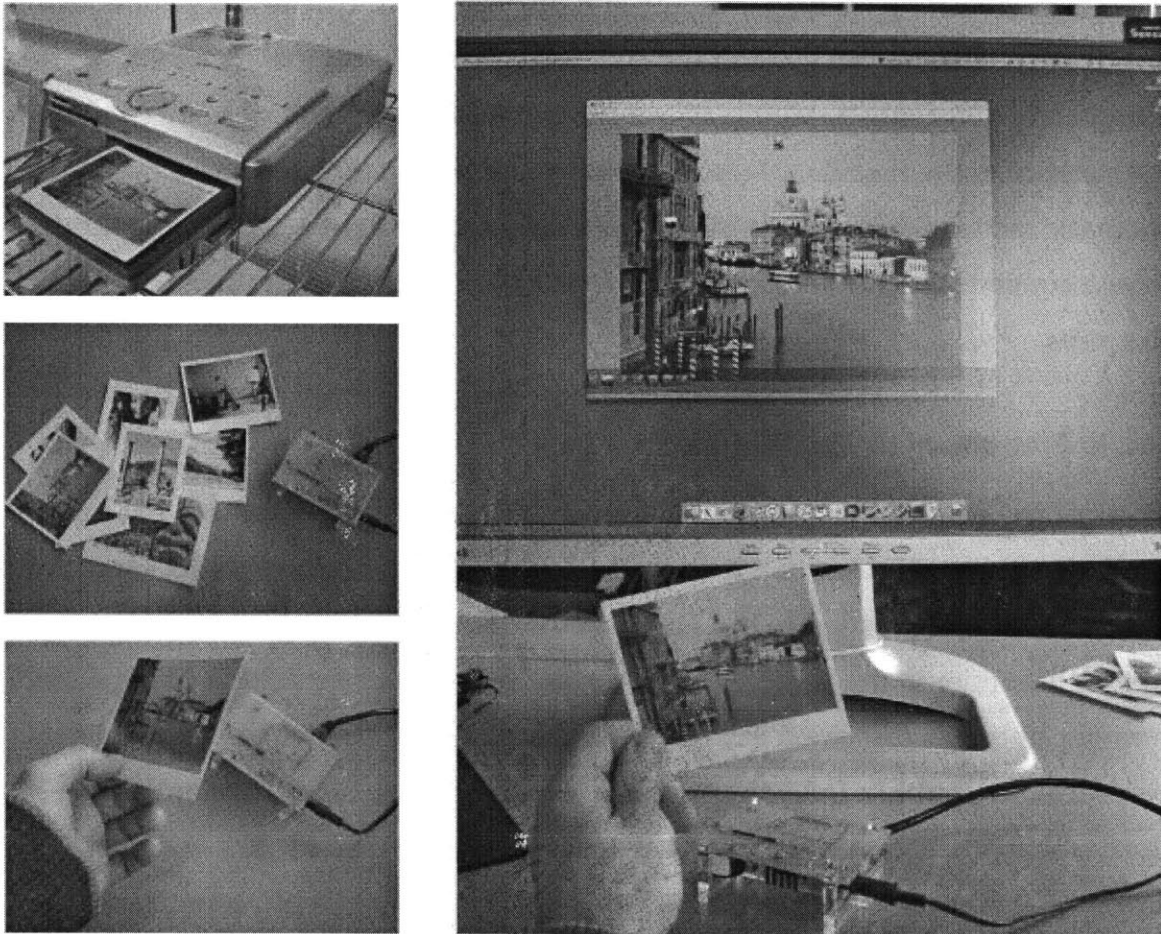


Figure 16: Radio frequency tag interaction

Back-end description

While the user is engrossed in composing her photo album of Ireland, her actions are being saved on a remote server. The server is watching where she positions the image thumbnail images, and which photographs appear where in the various image contexts. This information is being transmitted in real time to the Flexible File System. As soon as the user releases the mouse, the position of the thumbnail is updated on the Treehouse server.

For each image in the Treehouse database a record is kept for each time an image appears in an image context. The system stores the ID of the image context, and the X and Y pixel location of the thumbnail image.

3.2.6 Similar Viewer

Goals

Similar Viewer is an application used to query the Treehouse database to find related images based on a seed image. The goal is to provide innovative image-search functionality to the Treehouse image database, and to test the document organization structure created by the aggregate use of the Image Landscape application.

User Experience description

The Similar View application is a full screen application. A random sequence of small thumbnail images is displayed at the bottom portion of the screen; these images are retrieved from the Treehouse server, and represent a sampling of recent image-content stored there. The left half of the screen displays the key image; this is the largest image on the screen, and represents the seed image for an image search. The right half of the screen displays the query results. All the images in the database are scored in comparison to the seed image. The most relevant matches show up in the top row, left to right. The next row shows weaker matches, and so on. The images that have the highest-matching score appear as the larger images on the right-hand side. All matching images also have an ordinal number stamped on their bottom left corner, indicating their match fitness relative to the seed image.

As in the Image Landscape application, the user may browse the image space by clicking on any of the images in the display. When a user clicks on an image, it becomes the seed image, and the search results are adjusted to show the closest matches relative to the new search seed. These results are live and will change in

real time if changes are being made in the database. In this way, the Similar Viewer may be used as an ambient display of the Treehouse Database.

Back-end description

The Similar View application constantly sends queries to the Treehouse database asking for the most related image to the key image. The actual SQL query looks something like this:

```
select two.imageID as id, count(two.imageID) as
count, sum(two.xloc) - sum(one.xloc) as dx,
sum(two.yloc) - sum(one.yloc) as from organizer
one, organizer two where
one.parentID=two.parentID and
one.imageID=theMainImage.id and one.imageID <>
two.imageID group by two.imageID order by
one.parentID DESC LIMIT 14;
```

The query is designed to find all the occurrences of the key image in all of the photo albums on the Treehouse Studio server. For each Image Landscape that the key image appears in, a search is conducted to find all the other images that also exist in that landscape. For each image that shares one or more Landscapes with the key image, a metric score is computed. The score for each related image is the maximum pixel distance possible for a context image minus the actual measured pixel distance between the related image and the key image. This computation is repeated and summed for each time the two images appear together. This sum is the related images' score. This simple metric was designed to explore simple linear relationships between image context and image meaning.

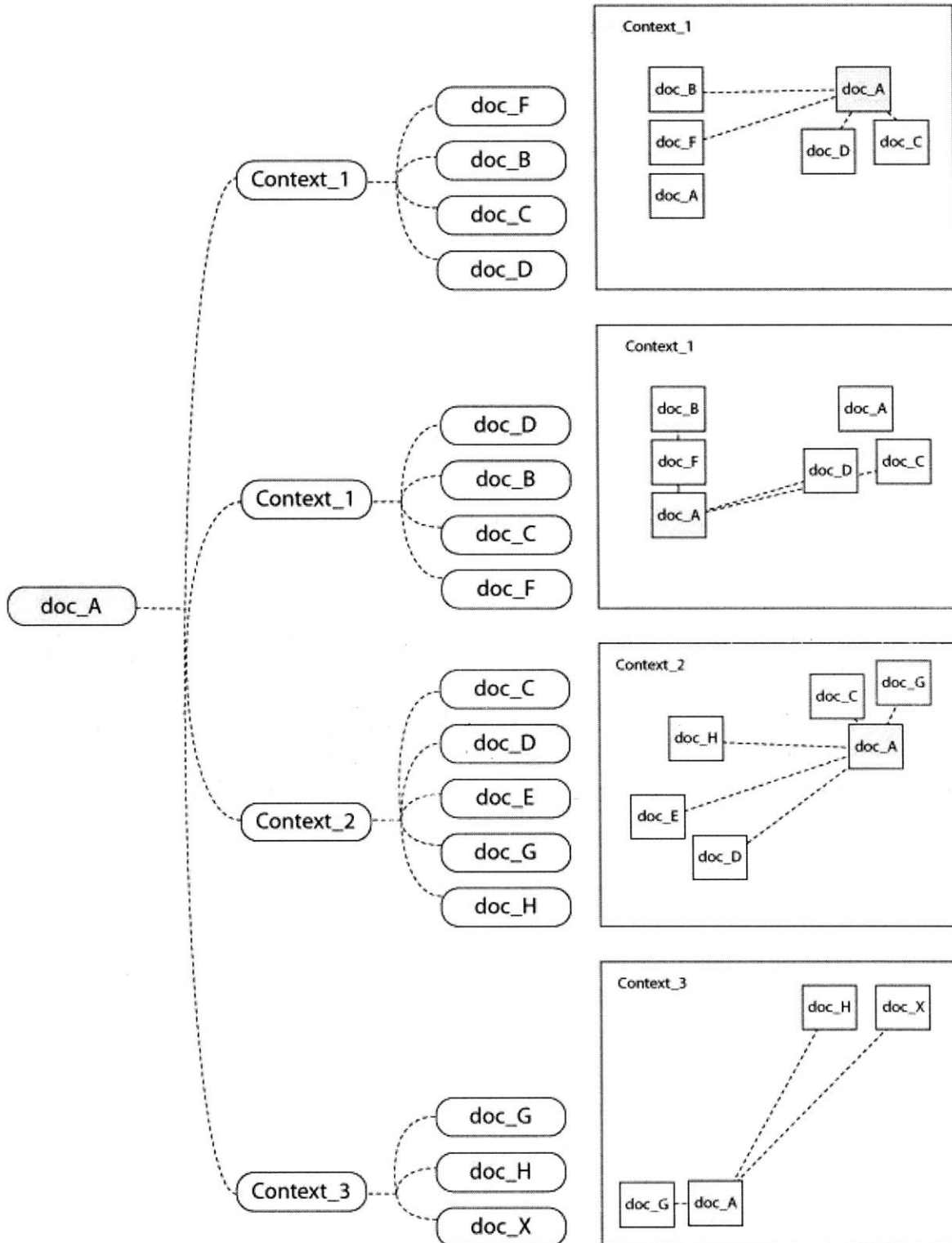


Figure 17: Representation of a document in multiple contexts

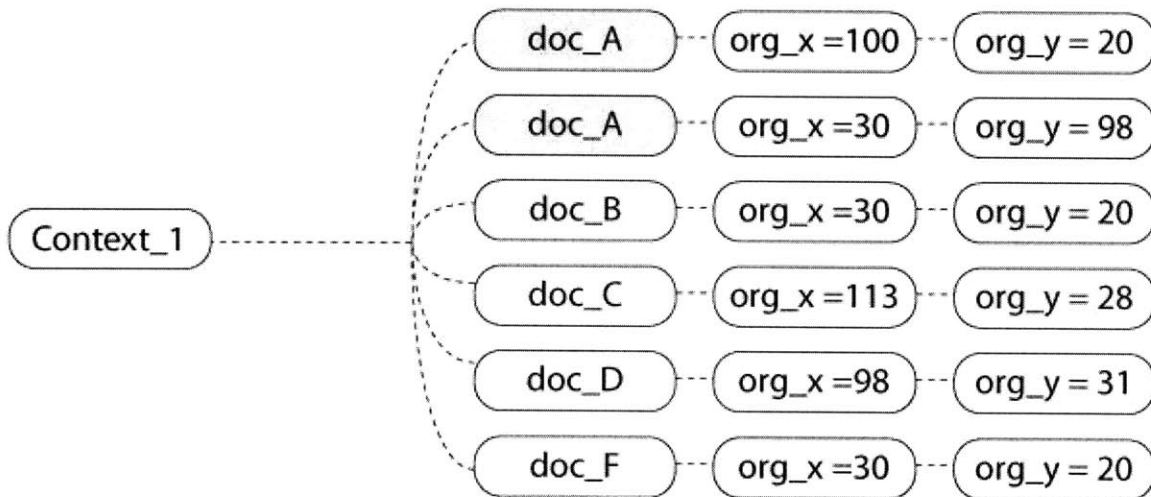
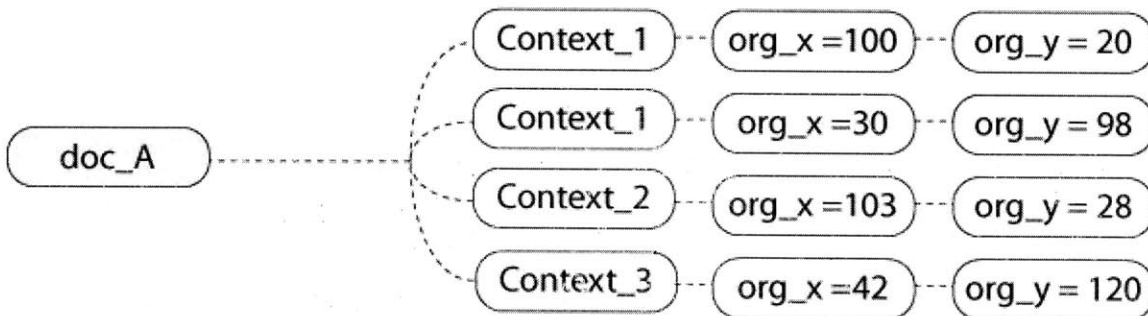
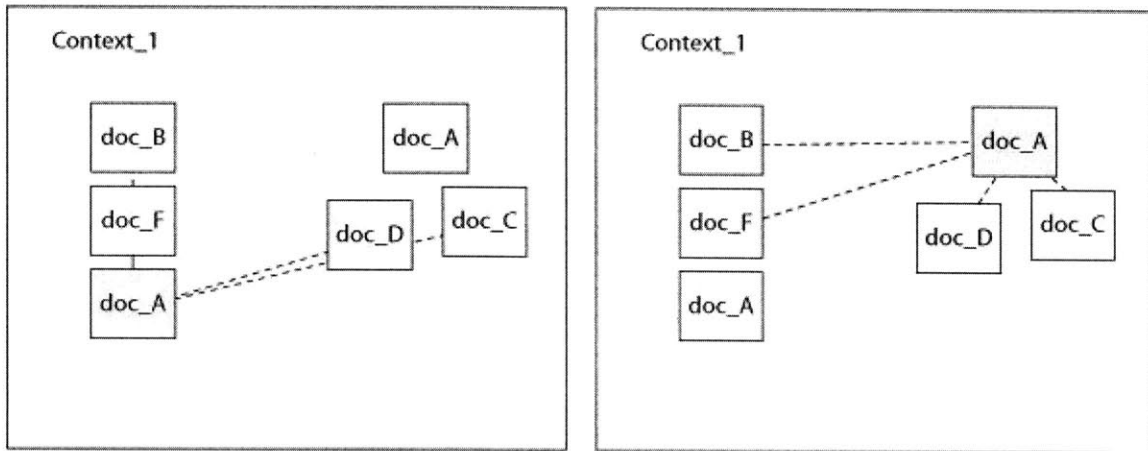


Figure 18: Aggregate distance between documents in various contexts

3.2.7 File Organizer

Goals

The File Organizer was designed as a more general-purpose revision to the Image Landscape application. Where the Image Landscape concerned itself only with viewable images, the File Organizer allows for the spatial organization of arbitrary shared documents. The File Organizer was also designed to be the default method by which documents are transferred from the local machine into the Treehouse Studio system. It was designed to be a fast, flexible, easy-to-use interface to the Flexible File System, and it was also designed to use special metaphors as one of the key document indexing and searching mechanisms. The File Organizer application was not completed, and many future directions remain for this promising work.

User Experience description

File Organizer is a Java application that runs as a stand-alone application. The interface is a large panel, which is vertically split into two. On the left-hand side is a spatial viewing area, similar to the Image Landscape application. This side displays icons and thumbnails of documents that have been positioned in this Landscape. The right-hand side behaves as a traditional file browser, showing the names of files. Above the file list are two search fields, for searching document names and document tags. The tags are user-defined and allow for arbitrary document tagging in a Folksonomic manner. As users type in tags, keywords, or name space letters, the documents that are listed are updated on every keystroke. The user can view her own documents as well as the published documents of her friends and associates.

Documents from the listing panel can be dragged into the spatial canvas on the left-hand side of the interface. Once a document is in the spatial canvas, the

document can be placed anywhere on the arbitrarily large two-dimensional plane. There are tools for zooming in and out of this space, and for navigating the space. Document proximity in the File Organizer application is handled the same way as in the Image Landscape application. When documents are positioned close to each other, there is an explicit relationship or similarity between the documents. Unlike the Image Landscape application, document locations are not stored in pixel space units, but instead are stored in arbitrary floating-point units.

Users can easily bring files into the Flexible File System by dragging items from their local file system into the document landscape of the File Organizer application. This drag-and-drop procedure creates a new file on the Treehouse server, which makes the file instantly available to all Treehouse Studio applications. Also, when a user drops the file into the Document Landscape, the user can reposition the document to anywhere she would like in that Landscape.

The spatial canvas area of the File Organizer could support multiple contexts, just as the Image Landscape application. These contexts would have keyword names as identifiers, opening up the possibility of spatial Folksonomies.

Back-end description

As files are dragged and repositioned inside the File Organizer application, the database on the Treehouse server is being updated with the location information of the documents. This spatial information is being stored in the document attribute table of the Flexible File System design. This enables the File Organizer application to monitor how users interact with their documents. Mouse motions are sent directly to the server, where the latest document locations are always recorded. The users interactions are relayed to the server as soon as the gestures are complete. This allows us to track similar files using the same algorithm as described in the Image Landscape application.

The retrieval of documents can now be based on text tags, and also on proximity information harvested from the user. Document-retrieval tasks can leverage both the simple tagging system and the spatial proximity data. Users can formulate queries around keywords, or key documents. If a keyword is provided, the database is first searched for documents that contain that keyword. The search can then be expanded to include documents that are spatially nearby documents that have matched the keyword criteria. Keywords are expanded based in part on the spatial relationships established by observing the user's document organization techniques. If a key document is provided, instead of a key word, the first step of this process can be ignored, and the nearest documents are returned directly, as described in the Image Landscape technique.

4 Analysis

Though some of the software is still not complete, and a detailed user study has yet to be conducted, the results of the software thus far are encouraging. The software has proven that the concept of harvesting user gestures for the purpose of document categorization is sound and feasible. It is also indisputable that large user communities are capable of taking on enormous and complex organizational tasks. A critique of existing document-retrieval techniques, specific shortfalls of the Treehouse Studio's Interface Harvesting software, and directions for future development are outlined in the section below.

4.1 Existing commercial applications

While no single application exists that solves the document management problem, there are a number of commercial offerings and websites that have proven very inspirational in the pursuit of developing an Interface Harvesting approach. An attempt is made below to acknowledge applications that have established baseline expectations for document retrieval. Additionally, promising new trends and advancements are also described and critiqued in this section.

4.1.1 iPhoto

Because of the digital photography revolution, and the associated difficulties of categorizing images, many innovations in document management software are concerned specifically with organizing digital images. Apple Computer's photo browsing tool, iPhoto, ships standard with every Macintosh computer, making it one of the more widely used document organization applications of today.

iPhoto has an attractive interface for viewing photo libraries. It shows an adjustable-sized thumbnail preview of every image in your library, arranged primarily by the date that the photograph was taken. The previews look very

nice, and scroll performance is decent for libraries up to a thousand or so images. Beside chronology, iPhoto allows the user to create photo albums, collections of photographs that the user selects. In addition to the standard photo albums, there is a special kind of photo album called a “smart album,” that allows the user to set conditions for set membership on the photo album. For instance, a user could create a smart photo album that always contained the photographs of the last month. So there are three ways of navigating iPhoto collections: chronological browsing, browsing photo albums, and browsing smart albums.

Versions of iPhoto up to and including Version 4 have only weak support for keyword tagging of images. The interface for tagging images is crude, and involves unnecessary windows. The keyword list is small initially, and though it can be added to and changed by the user, the amount of space allowed would accommodate no more than 20 keywords. In addition to a weak interface for tagging images with keywords, iPhoto Versions 4 and lower also lack any sort of ‘search’ feature. Though I have not tried it yet, Version 5 of iPhoto does have what looks to be a better keyword-labeling interface, and also has a search bar. It will be interesting to see if people will start to use keywords to manage their photographs in this new version of the Apple product. Without proper motivation, my prediction is that the majority of users will not add captions or keywords to their photographs.

4.1.2 Picasa

Picasa is said to be iPhoto for the non-Macintosh crowd. A free software tool distributed by Google, it has many of the same features of iPhoto, plus quite a few nice tricks of its own.

“Picasa is software that helps you instantly find, edit and share all the pictures on your PC. Every time you open Picasa, it automatically locates all your pictures (even ones you forgot you had) and sorts them into visual albums organized by date with folder names you will recognize. You can drag and drop to arrange your albums and make labels to create new groups. Picasa makes sure your pictures are always organized.”

One of the key features of Picasa seems to be that it locates your photo assets on your hard drive by itself, and then organizes them based on EXIF and file name data. This is a great start, because most users will begin organizing their images only after they have acquired a large number of images.

In addition to its auto-discovery feature, Picasa supports image tagging, image searching (based on tags), photo albums, and chronological sorting. Additionally Picasa adds a lot of methods for sharing photographs. Essentially, Picasa really is iPhoto for Windows-based computers. They are approximately functionally equivalent.

4.1.3 Flickr

Flickr, as discussed earlier, is a very innovative and promising website. Offering a service for sharing photographs is the core functionality of the website, but the manner in which they achieve this is astounding. For many users, uploading their photographs from iPhoto or Picasa to Flickr would make it easier for them to find their photographs. This is because of the rich tagging and searching system that Flickr integrates with its publication system. Because users typically want to publish their images, they are motivated to provide the appropriate tags to their photos; this means that their documents are more retrievable by everyone. Flickr begs the question, could this technology be extended to incorporate data besides photographs?

4.1.4 del.icio.us

Though it bills itself as a social bookmark manager, del.icio.us is really an open-ended tagging system. The best example of Folksonomies to date, del.icio.us shows us that aggregate information from a large user community matters. Consider that it is much more meaningful to say that over 300 people have tagged one document with the word "Programming," than it is to say that a document has been tagged with the word "programming" alone. del.icio.us gives us not only a keyword, it also gives us a metric, some way to measure just

how much 'about' something a document is. This is one of the fundamental differences between broad Folksonomies and narrow Folksonomies.

del.icio.us was designed as a shared bookmark manager, and it does a great job at that. It also points us to a new way of classifying documents. Although many of the items we would like to track in our digital lives have urls, many of them do not. We need a way to lift the Folksonomy concept found in del.icio.us out into the desktop space.

4.1.5 Google

As of this writing (April 2005) Google is still be the dominant player in document retrieval. They do a great job indexing text documents (PDFs, PostScript Documents, web pages, etc.). They also have a great way to identify images based on the context in which the images are encountered. The main feature of the Google search engine is defining the relative importance of a reference in addition to finding its primary subject areas. Google does this with a system called Page Rank, which basically gives every web page on the network a relative value, based on how many other web pages point to it (and also what their ranks are).

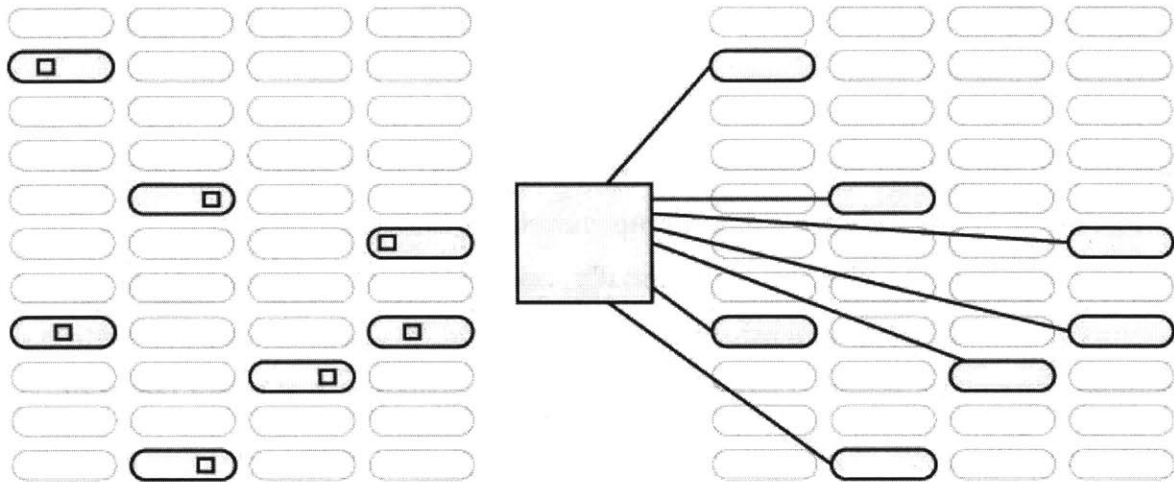
Google is unquestionably the document-retrieval victor in webs of linked text documents. Documents that do not reference each other in such a direct-linking mechanism as used on the World Wide Web do not fit immediately into the Google-searching framework.

4.2 Analysis of completed software

The document organization software developed for use in the Treehouse Studio attempts to demonstrate that observing and recording how users interact with documents as part of their creative process can add meaningful meta-data to the document which can be used later as an organizational parameter during the retrieval process. The existing interface retrieval software presented is critiqued

in this section. The software is critiqued on its own merits, but is also analyzed in terms of one specific example of a general approach to document classification based on how users interact with their documents. Opportunities for future work that may address some of the problems of the existing software are presented in Section 4.3

The Image Landscape software is a unique collaborative experience, where multiple users create and share a rich and boundless visual space. Each user can contribute to the arrangement of narrative and visual elements. In many respects, the hyperlink structure resembles something like the World Wide Web. In the case of the Image Landscape application, browsing and editing the web are one unified task, similar to Wikis. Additionally, because of the placement of one image adjacent to another or inside another adds meaning to the system as a whole, Image Landscape functions as a type of fuzzy non-text Folksonomy. A multidimensional association matrix is established between images based on summation of proximities; these spatial relations are dictated by the culmination of actions of a large user community. The Image Landscape software succeeds in at least two ways: it is a unique publication framework for sharing images, and it also functions as a unique tagging system, in which objects are given associated meanings by a cumulative community-curated spatial relationship.



1B,1H,2E,2K,3I,4F,4H

classification by tagging (folksonom)

Figure 19: Faceted tagging

Evaluation of the first feature, the publication of framework for sharing images, can only be established through user feedback. We want to understand how difficult and enjoyable it was for users to publish their images this way. Similarly, we might wish to understand how difficult and entertaining it was to browse the published work.

Evaluation of the tagging system should be less subjective, and should involve some metrics indicating the discoverability of documents from within this framework. Additionally, we could interview participants, who should indicate how difficult or enjoyable they found the process of searching for or browsing images.

We believe that the Image Landscape software does succeed in providing a proof of concept model for Interface Harvesting, however many features could be added to a future implementation which would make for a more generally useful application. In some cases, these shortcomings have been partially addressed in the second version of the software, called the File Organizer.

Additional features and directions are missing entirely and should be addressed in future software.

Currently, the Image Landscape software only records the spatial relations of image documents. Arbitrary documents cannot be stored or categorized by the Image Landscape application. Additionally, no other meta-data is used in conjunction with the spatial information. We believe that identifying correlations between various meta-data elements will provide a deeper and richer sense of meaning. One of the key benefits of keyword expansion based on the proximity of documents has not even been achieved in the Image Landscape application at this time.

The File Organizer software does attempt to address both of these shortfalls of the Image Landscape software, but at the moment this software is incomplete. While the File Organizer software does allow users to place keyword-tagged documents inside of a spatial context, it is less visually appealing than the Image Landscape application. Additionally, because the document types are unknown, the File Organizer software cannot function as a platform for Image Sharing.

4.3 Future directions

Both the Image Landscape and File Organizer software would benefit from some batch processing, and default spatial-mapping operations provided to the user. For instance, when importing image files, there should be the option to import entire sequences, folders, or directories of files from the local operating system. Additionally, there should be the option to automatically map some file attribute, such as file size or modification date, to either the X or the Y-axis. Having this default mapping-mechanism for files would be a great way to initiate the curatorial process, and could provide a default visualization for multiple pieces of meta-data. Such a tool would facilitate the importation of files into the system.

Another area for future exploration is extending the Image Landscape application to include support for annotated regions of the Landscape. We could provide an interface for labeling salient sections of the underlying background image for a given context. For example, assume that we have a context image of a seascape. The user could annotate the background image of the seascape by selecting a rectangular region of the ocean, and optionally label the section "Ocean." Images or documents that are then placed within that region would have an "Ocean" attribute associated with them. This particular extension may not be necessary for the Image Landscape scenario, because in most ways it is redundant. Simply by placing images of all things "ocean" near each other, the same effect will be achieved. However, it would be interesting to explore the idea of sub-regions in more detail.

Another direction for exploration would be differentiation between various types of spatial relationships. The Image Landscape application primarily investigates spatial proximity of documents; however, it would be interesting to also track objects which overlap, touch, and contain each other. It is an open question how to map these spatial relationships in ways that would be meaningful for the purposes of document retrieval. Specifically examining and classifying the containment hierarchy of image documents may yield additional information about the contents of those documents. In the current implementation the Image Landscapes only consider one level of document containment. Future versions of the software could consider the entire document graph of each element.

5 Conclusion

There can be no doubt that the way in which we are communicating is changing rapidly. More documents are being published online than ever before, thanks to technologies that encourage grassroots participation, such as blogging and Wikis. Increasingly, the types of documents we work with and publish are not text documents; instead we publish images, audio, and video content. How we access, store, and search through digital documents needs to be reexamined in this context. In this era of very rich media, technologies that were designed to store, process, and analyze text documents are no longer adequate. There is a need to develop systems that classify documents based on subjective similarity metrics. For search flexibility and extensibility these metrics should work across all media formats, regardless of internal file structure.

The manual categorization of documents, artifacts, and information, which evolved over many thousands of years, has served us quite well up until this time. In fact, systems such as the Dewey Decimal System or the Linnaean taxonomy would continue to serve users quite well today, apart from the fact that media is being created at a rate and in a manner which makes centralized classification practically impossible. As much as we may love these definitive classification schemes, their days may be numbered, given this inability to handle the majority of the content being created today.

Several interesting tools for classifying and categorizing documents have developed which reflect the latest publication paradigms. Curiously, these classification tools bear a lot of resemblance to the publication tools that created the massive surge in media documents in the first place. Just as new publication and production tools make it simple for anyone in the world to publish a multimedia document, new paradigms in document classification allow groups of individuals to establish their own democratic grassroots organizational

systems. As more applications are delivered in part or whole over the World Wide Web, it will become increasingly trivial to analyze the aggregate activities of online communities as they interact with documents. Even the most fundamental and simple document gestures can be revealing when community trends can be studied as a whole.

This thesis has proposed one specific method for classifying documents based on the interaction patterns of individuals and groups. The general strategy of this approach is to add meaning to document collections by observing how users interact with their documents. We believe that this technique is unique in that it is flexible enough to help categorize a wide variety of document types, and does not rely upon a manual document-annotation process. We have described a flexible and network file system that enables faceted document classification. We have exhibited sample applications that demonstrate Interaction Harvesting techniques, and we have shown how these techniques can be used to establish metrics for document similarity. We have also demonstrated an example retrieval application that let us browse documents based on these similarity metrics, in order to assess the viability of the similarity metrics created.

These early explorations indicate that recording the ways that users interact with their documents could significantly enhance document classification and retrieval. The creative process which gives rise to documents, and the way that documents are experienced and used are important pieces of a document's metadata. We hope that in networks and operating systems of the future, these important aspects of documents will not be discarded, but rather used to help others find the information that they seek.

6 List of figures

<i>Figure 1: Using a card catalog.</i>	13
<i>Figure 3: Index card anatomy.</i>	19
<i>Figure 4: Wikipedia.</i>	29
<i>Figure 5: Flickr.</i>	34
<i>Figure 6: Flickr tags.</i>	34
<i>Figure 7: Organizational practice #1</i>	45
<i>Figure 8: Organizational practice #2</i>	45
<i>Figure 9: Organizational practice #3</i>	46
<i>Figure 10: Organizational practice #4</i>	47
<i>Figure 11: Organizational practice #5</i>	48
<i>Figure 12: Database representation of the Faceted File System</i>	51
<i>Figure 13: Importing images from the Treehouse Studio</i>	56
<i>Figure 14: Importing images into the faceted file system</i>	57
<i>Figure 15 Zooming into a new context</i>	60
<i>Figure 16: Radio frequency tag interaction</i>	62
<i>Figure 17: Representation of a document in multiple contexts</i>	65
<i>Figure 18: Aggregate distance between documents in various contexts</i>	66
<i>Figure 19: Faceted tagging</i>	75

7 Bibliography

- [ama05] Amazon.com retrieved from <http://amazon.com/> on april 25th 2005, April 2005.
- [Bat03] Mathew Battles. *Library, an unquiet history*. W. W. Norton & Company, 500 Fifth Ave New York New YOTK, 2003.
- [BP00] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine retrieved from <http://www-db.stanford.edu/backrub/Google.html> on april 25th, 2005, 2000.
- [BPMS01] William Birmingham, Bryan Pardo, Colin Meek, and Jonah Shifrin. Musart: Music retrieval via aural queries. Bloomington, IN, 2001. ISMIR 2001.
- [CLS02] Mao Chen, Andrea LaPaugh, and Jaswinder Pal Singh. Categorizing information objects from user access patterns. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 365–372, New York, NY, USA, 2002. ACM Press.
- [CMK91] John M. Carroll, Robert L. Mack, and Wendy A. Kellog. *Handbook of Human-Computer Interaction 2nd. ed.* North Holland, Amsterdam, Netherlands, 1991.
- [Cod70] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970.
- [Dai03] James J. Dai. Visual intelligence for online communities, commonsense image retrieval by query expansion. Master's thesis, MIT, 2003.
- [Del05] del.icio.us, retrieved from <http://del.icio.us/doc/about> on April 25 2005, April 2005.
- [DJ85] Susan T. Dumais and William P. Jones. A comparison of symbolic and spatial filing. In *CHI '85: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 127–130, New York, NY, USA, 1985. ACM Press.
- [Esp03] The esp game: Labeling on the web retrieved from <http://www.espgame.org/> on april 25h, 2005, October 2003.
- [Fel98] Christiane Fellbaum. *WordNet An Electronic Lexical Database*. Number ISBN 0-262-06197-X. The MIT Press, 1998.

- [Fli05] Flickr retrieved from <http://flickr.com> on April 2005.
- [GIS05] Geographic information system retrieved from <http://en.Wikipedia.org/wiki/gis> on April 25th, 2005, 2005.
- [Gri04] Richard Grimes. Revolutionary file storage system lets users search and manage files based on content. *MSDN Magazine*, January 2004.
- [HEP01] Lieberman Henry, Rosenzweig E., and Singh P. Aria: an agent for annotating and retrieving images. *Computer*, 34:57-62, July 2001.
- [ISB05] What is an ISBN retrieved from <http://www.isbn-international.org/en/whatis.html> on April 25th, 2005.
- [Lie97] Henry Lieberman. Autonomous interface agents. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 67-74, New York, NY, USA, 1997. ACM Press.
- [LOC05b] Library of congress classification outline retrieved from <http://www.loc.gov/catdir/cpsol/lcco/lcco.html> on April 25th, 2005.
- [Luh57] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1:309-317, 1957.
- [Mal83] Thomas W. Malone. How do people organize their desks? : Implications for the design of office information systems. *ACM Trans. Inf. Syst.*, 1(1):99-112, 1983.
- [Met96] R. Metcalfe. There oughta be a law. *New York Times*, July 15 1996.
- [NW79] George Nagy and Sharad Wagle. Geographic data processing. *ACM Comput. Surv.*, 11(2):139-181, 1979.
- [PCF02] John C. Platt, Mary Czerwinski, and Brent A. Field. Phototoc: Automatic clustering for browsing personal photographs. Technical report, Microsoft Research, 2002.
- [Per02] Petra Pernert. *Data Mining on Multimedia Data*. Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2002.
- [RIS94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175-186, New York, NY, USA, 1994. ACM Press.

- [Sal83] Gerard Salton. *Introduction to Modern Information Retrieval*. McGraw-Hill Inc, New York, 1983.
- [Sha94] Upendra Shardanand. Social information filtering for music recommendation. Master's thesis, Massachusetts Institute Of Technology, 1994.
- [Sin02] Push Singh. The public acquisition of commonsense knowledge. Palo Alto, CA, 2002. AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access.
- [SK00] Ben Shneiderman and H. Kang. Direct annotation: A drag-and-drop strategy for labeling photos. London, July 2000. International Conference on Information Visualization, IEEE.
- [SM83] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Publishing Company, NewYork, NY, 1983.
- [SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888 – 905, August 2000.
- [Sto99] David G. Stork. The Open Mind Initiative. *IEEE Expert Systems and Their Applications*, 16-20, May 1999.
- [SW81] Gerard Salton and Harry Wu. A term weighting model based on utility theory. In *SIGIR '80: Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 9–22, Kent, UK, UK, 1981. Butterworth & Co.
- [Sza95] Katalin Szabo. Metaphors and the user interface retrieved from <http://www.katalinszabo.com/metaphor.htm> on April 25th 2005.
- [UDC05] About universal decimal classification retrieved from <http://www.udcc.org/about.htm> on April, 25th 2005.
- [WDS01] Liu Wenyin, Susan Dumais, Yanfeng Sun, HongJiang Zhang, Mary Czerwinski, and Brent Field. Semi-automatic image annotation. Conference on Human-Computer, 2001.
- [Wal05] Thomas Vander Wal. Folksonomy: A wrappers delight retrieved from <http://www.vanderwal.net/essays/folksonomy/050307folksonomy.pdf> on April 25th, 2005. March 2005.
- [Wik01] Wikipedia, the free encyclopedia retrieved from <http://en.Wikipedia.org/wiki/Wikipedia> on April 25th, 2005, 2001.

