Beyond Keywords and Hierarchies

Ian Hopkins and Julita Vassileva
Department of Computer Science,
University of Saskatchewan,
110 Science Place, Saskatoon S7N 5C9, Canada

[ikh328@mail.usask.ca; jiv@cs.usask.ca]

Abstract

As our ability to store information increases, the mechanisms we employ to access that information become ever more important. In this paper, we present Archosum, a prototype of an organizational system that attempts to encapsulate the benefits of both hierarchical and keyword systems. By introducing abstract entities, Archosum provides a simple interface with which users can build and maintain powerful relationship-based organizations. We compared Archosum to both hierarchy and keyword based systems in a user study. Through our users study we begin to expose some of the advantages and disadvantages to three approaches to designing an organizational system. Furthermore, we begin to consider how organizational systems will work when distinct users create organizations for collections and how sharing might be facilitated using Archosum.

1. Introduction

The continuing, rapid expansion of the Internet [25] supplies an ever-increasing quantity of information to the fingertips of connected users around the world. This is a mixed blessing; as the quantity of information increases, our ability to find a particular piece of information decreases. As the size of a collection grows, there is an increasing need for mechanisms to organize that collection. The purpose of any organizational system is to allow a user or group of users to quickly and accurately find a subset of entities within a collection. An information retrieval system is a type of organizational system. This type of system informs the user of the existence and whereabouts of information relating to his/her request [23].

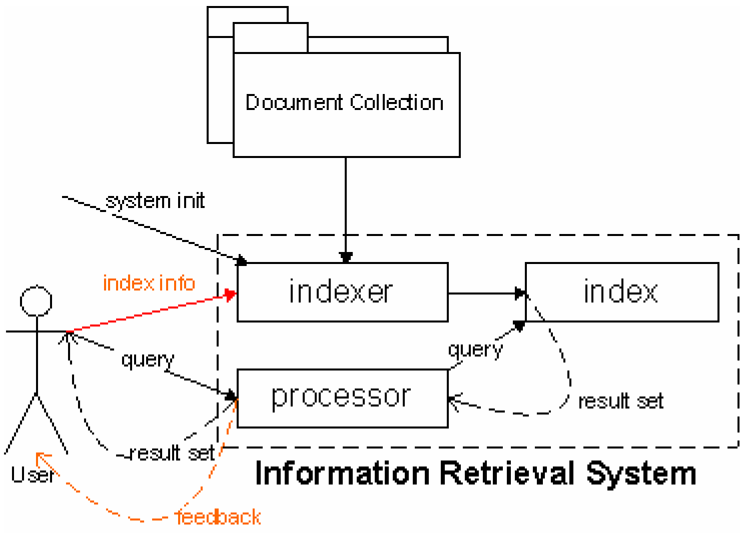


Figure 1. An Information Retrieval System, adapted from [23].

Assuming that a collection contains information that is of value, then an organizational system is a tool that allows us to extract value from a collection as the collection grows. In this sense, an organizational system is of great value. As the Internet continues to expand, it continues to increase in value, but our ability to exploit that value remains in question. Any system that could organize even a portion of the Internet would be of great value; for example Google's initial public offering raised over $30US Billion in capital [11].

Google is not the only company to have noticed the value of organizational systems. Operating systems typically implement an organizational system to keep track of the files saved by users and applications on long term storage media. As storage capacity grows and prices of storage fall, Microsoft, Apple and the open source community

--- To appear in R. Chbeir (Ed.) Special issue on Distributed Data Management of the *Journal of Data and Information Management.*

have recognized the need for a new system in which users can organize their ever expanding collections of files and to share their collections with those of other users [6,15,17,19].

There are many different approaches to building organizational systems, for example, SFS, a file system based on semantic file attributes similar to those used in data-base management systems [9] and more recently, VennFS, a Venn-Diagram based organizational system [5]. Yet most large scale, deployed applications are either based on hierarchies (e.g. computer file systems) or keywords (e.g. web search engines and Google Desktop Search [10]). By learning from the features present in both of the standard approaches and alternative organizational systems, we hope to design an information retrieval system that will allow users to access their ever increasing collections with accuracy, speed and ease and allow them to share collections with other users.

To build an intuitive and accurate organizational system, we looked to several sources including the original ideas of hypertext as advanced by Dr. Vannevar Bush in his article in 1945 describing Memex [3], an approach to organizing information through the use of associations. Tim Berners-Lee used this idea in 1989 to implement a protocol for hypertext: a set of interlinked digital documents which became the basis of the World Wide Web [1] and subsequently advanced this idea in the Semantic Web.

We took a subset of features from the Semantic Web as the starting point for our organizational system, which we call Archosum. In building a semantic web organization, users make associations between entities, but as the size of a collection grows, the complexity also increases. As we will see in section 3, managing an organization for a changing collection becomes very complicated. Archosum abstracts the details and assists users in creating and managing complex organizations, thus allowing users to utilize a Memex style organization without the usual upkeep costs.

Organizational systems can be applied to collections of entities of any type (e.g. text documents, books, music, movies, files on a computer disk or web-addresses (URLs). In the context of this paper, we define entities as anything that can be referenced by a Universal Resource Identifier (URI), e.g. URIs of websites of books, movies, research groups or personal homepages, on-line research articles and others.

2. Organizational Approaches

Library scientists have been working on ways to organize collections of physical documents for decades [13]. With the introduction of computer catalog systems, many libraries have chosen to utilize more than one information retrieval system to give patrons multiple methods of finding documents within their collection. Although many systems exist, there are two general approaches in wide use that we will discuss before presenting the details of our approach.

2.1 Hierarchies

Hierarchical tree structures can be found in a variety of fields dating back to the 18th Century from Biology (Linnaean taxonomy) to Psychology (Maslow's hierarchy of needs) to Linguistics (Chomsky hierarchy) [13]. The world's most famous hierarchy dates back to 1873, the first implementation of the Dewey Decimal System.

The Dewey Decimal System is a hierarchical organizational system designed by Melvil Dewey to catalog printed resources within a library. Each document is assigned a set of digits that correspond to categories in a common hierarchical tree. The categories in this tree are subdivided by further digits with matching subcategories.

From computer desktops to corporate power structures, hierarchies have become an extremely popular way to organize information due to several benefits:

1. **Computational Efficiency –** hierarchies are very space- and computation–time- efficient.
2. **Understandable** – hierarchical structures have been woven into the fabric of society and hierarchical metaphors are used in everyday life (e.g. placing an object in a drawer in a desk, which is in a room, in a house, on a street, in a city, in a country, on a continent etc.). The average user understands intuitively how hierarchies are organized and uses these intuitions when organizing information in a hierarchical way.
3. **Ease of Recognition** – the cognitive process of browsing down a hierarchy and narrowing down the space for search until recognition of an entity or browsing up in a hierarchy and widening the scope (used when browsing a hierarchy) is easier than recalling keywords, labels or formulating queries.

In small collections, these advantages of hierarchies make them a good choice for organization. However, as the collections increase in size, certain problems become evident:

1. **Static Organization** – the structure of a collection can not adapt to changes in the collection, as the organization is specified explicitly by the user.
2. **Single Tree Order** – users must arbitrarily decide the order of directories even when this ordering is not intuitive and furthermore, remember that ordering when looking for a document. For example, we might have decided to save this paper in either of the following directories: "/Docs/Papers/Projects/Archosum/" or "/Projects/Archosum/Docs/Papers/". It isn't obvious whether the first choice is better or more likely than the other [22].

3. **Growing Complexity** – as a collection increases in size, the organizational tree must grow in either depth or child cardinality. Either growth is undesirable; as depth increases, the number of refinements from the root to leaves increases; and as child cardinality increases, the number of choices at each refinement increases.
4. **Lost Relationships** – a single hierarchy represents ideally one semantic relationship. Any other relationships between entities are lost.

Because of these problems, classifying and finding documents in hierarchies becomes more difficult as the size of the collection increases. The Dewey Decimal System is standardized and maintained exclusively by librarians trained in classification, but many users still prefer to search a keyword catalog to navigate the massive library collections.

Although there have been various attempts to organize the Internet into a hierarchy, the Open Directory Project is the most widely distributed and largest human edited directory online [19]. In 1998, Richard Skrenta and Bob Truel started the Open Directory Project (ODP) which today has over 65,000 volunteer editors attempting to organize the internet. Table 1 presents a comparison of the growth of the Internet and ODP since its inception.

Table 1: Internet Size vs. Open Directory Project Size (based on data from [14, 18, 19, 24]).

Date	Internet Hosts	Growth/ month*	ODP Sites	Growth/ month	ODP Coverage*	ODP Editors	new sites /editor /month
Dec 1998	3,689,227	---	172,725	---	4.68%	5,659	
Nov 1999	8,844,573	12.70%	1,131,301	50.45%	12.76%	18,831	4.63
Dec 2000	25,675,581	14.64%	2,269,579	7.74%	8.84%	32,612	2.68
Oct 2001	33,135,768	2.91%	2,924,179	2.88%	8.82%	41,319	1.58
Jun 2002	38,444,856	2.00%	3,500,495	2.46%	9.11%	---	---
Dec 2003	45,980,112	3.27%	4,000,000	2.38%	8.70%	---	---
Aug 2004	---	---	4,412,982	1.29%	---	64,378	0.80

We can make two important observations from Table 1:

1. The Internet continues to expand faster than the index of the ODP
2. As the size and the complexity of ODP increases, the number of sites each editor can add per month decreases.

Our observations are likely conservative since the table compares the number of internet hosts and not the number of organized websites with the growth of ODP. The number of websites per host has likely increased significantly over the last six years (due to advances in server hardware and software); thus the Internet growth is likely much larger than shown in the table.

If editors are having increasing difficulty in classifying websites, it is not for a lack of new websites as we can see from the growth of the Internet, but rather it reflects the crippling effect of the increasing complexity of the hierarchy. If the editors (the creators of the organization) are having increasing difficulty organizing the directory, it seems a logical conclusion that consumers of the directory will have equal or greater difficulty finding information in the directory.

2.2 Keyword-based

1. A keyword-based approach organizes documents by creating an inverted index of keywords related to each entity. In a keyword-based approach, users need not classify individual documents as the system can build an index of keywords based on the contents of any given document (and more recently keywords in documents known to be related [2]). Search engines consist of three components: A web crawler, an indexer and a search interface. The indexer is an automated process, which builds an organization based on the output of the spider. Since the process of building an organization is automated, the scalability of keyword based systems is not limited by human interaction, but rather by computation speed.

Due to this automation, keyword-based information retrieval systems allow us to organize document collections larger than ever before. Google has an index of over 8 billion documents, eclipsing the size of the ODP by nearly 2000 times. Google's index is also updated periodically, whereas the ODP's index is relatively static. However, keyword systems are not without their problems:

1. **Human Recall** – users must synthesize keywords that describe the entity they are looking for.
2. **Word Meanings** – if a keyword has more than one meaning, query revision may be required to obtain the correct result set. The vocabulary of the user may change his/her result set, if it is different than that of the content creator (and thus the index). Although queries can be expanded by user of thesauri like WordNet [7], the problem is rooted deeper in the approach since context is difficult to determine and store.
3. **Abstract Concept** – the keywords associated with a document by the system may not make sense to all users. These keywords also may change with the content or indexing algorithm causing users confusion and frustration when they attempt to find a document in the future.

4. **Misspellings, translation** – if either the content producer or the directory consumer misspells a word, it will not constitute a match. Again, text-processing techniques (e.g. word stemming), the use of dictionaries and other approaches can be used to limit, but not eliminate this problem [4].

3. Approach

We propose an approach that encompasses the benefits of both hierarchical and keyword-based systems while allowing users to create organizations and use them in a familiar way. Our system makes use of a directed graph to organize entities, where an entity is a unit of data (like a file, book, movie, paper, person's homepage etc.). Thus as users organize entities, a web of associations is built, resembling hyperlinks on the Internet. The problem with direct association among entities (the approach used to link hypertext or web documents) is that if we wish to add a new entity, there may be ten related entities that we must associate with it. Similarly, when we delete an entity we must remove ten associations. Updates to the collection would be complicated and consume both human and computer time.

To deal with this problem, we introduce the concept of abstract entities. An abstract entity is a named concept (e.g. a directory / class / category / genre). Users can relate abstract entities in the same way as they would relate non-abstract (existential) entities, but the relationship of an existential entity to an abstract entity is automatically propagated to all other existential entities related to the abstract entity. Let's consider an example of 10 entities (5 songs, 5 reviews). We want to relate all the songs together; all the reviews together; and each review to the song it covers.

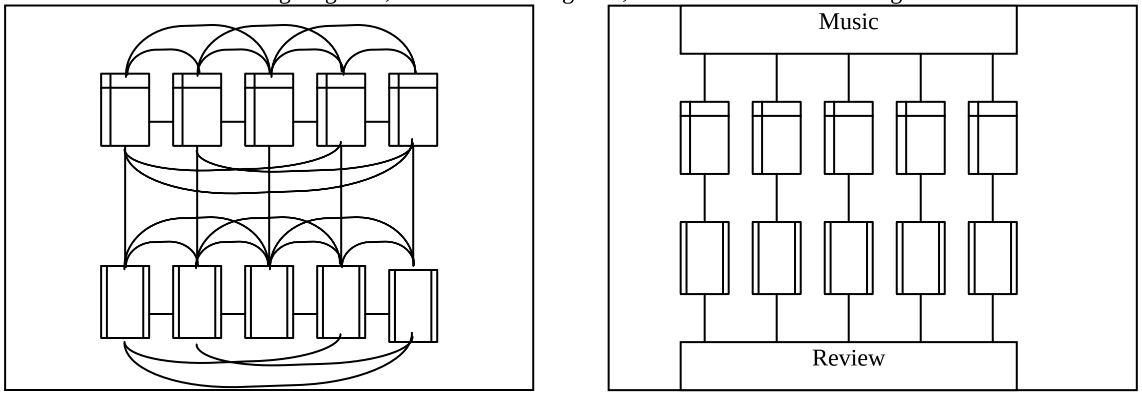


Figure 2a & 2b: Comparing Organizations

When an existential entity (Song) is associated with an abstract entity (Music), it is automatically related to all the other entities associated with that abstract entity; thus one review is associated with every other review even though the user created only one association (to the abstract entity).

Since abstract entities propagate relationships, Figure 2a and 2b express equivalent organizations. Abstract entities help simplify complicated relationship while preserving their meaning. This simplification also lends to scalability; if we wish to add a new song to the organization, the user needs only relate it to music (and all other songs will be related via propagation).

So our solution bears similarity to the ideas of Vanevar Bush, which inspired the development of hypertext, the concept of semantic networks in AI, the web and the semantic web. However, it extends these ideas in the following ways:

- Users manage a group of relationships (via abstract entities) while Archosum manages the resulting individual relationships.
- Users create abstract entities in a way similar to creating nodes in a hierarchy.
- The underlying structure of the organization is abstracted from the user, yet relationships are based solely on the decisions made by a user

When building an organization, users create abstract entities (from now on we will call them "categories" for simplicity) and organize entities (for simplicity from now on we will call existential entities just "entities") by associating them with other categories. The graph does not restrict the number of associations created so a given entity can have more than one association. Users have the sensation that they can put entities in more than one place and consequently they can access entities by browsing along multiple distinct paths in the graph. Furthermore, users can associate entities not only with categories, but with other entities (e.g. "Mark" is an entity. An email (another entity) from "Mark" is associated with the category "E-mails" but also associated with the entity "Mark").

4. Archosum: An Architecture for the Proposed Organization Approach

We have developed an architecture called Archosum for the organization approach described in the previous section. The architecture encompasses a user who creates an organization of entities, a representation of the organization (consisting of a graph with entities and categories) and a search interface, which allows the user to navigate the organization by browsing. The architecture provides also a communication channel allowing users to share their organizations on a peer-to-peer basis. However, sharing is not the focus of this paper. Rather, our focus is on the single user: allowing easy creation of scalable organizations and search by browsing.

Visualizing the organization is an important advantage of our approach, which the Archosum architecture needs to support, since like in hierarchical approaches the user will be able to browse the organization and find entities by recall rather than by formulating queries. However, a graph can be very difficult to visualize, especially as it grows [12]. To allow our approach to scale, the user should not see the entire graph but only a portion based on some starting point. In hierarchies there is a static root node that works as a starting point from which any other node can be reached. In our approach we eliminate the notion of a root node used in hierarchies in favor of allowing any entity or group of entities to act as a starting point.

Consider looking for music in your computer file directory, which is hierarchical. The root node (e.g. "My Computer" in Windows) is not a good starting point since it contains a lot of data that is not music. A better location would be to start with the directory that stores the music files on your computer (if there is such a directory). Using Archosum the user simply needs to enter as a query the category "Music" (as she would do in a keyword-bease search), which will bring up the portion of the graph containing entities and categories related to category "Music".

Archosum requires a starting point from which to display the organization. Selecting a starting point depends on the purpose of the user and we believe applications can effectively capture some element of this purpose and pass it on to Archosum (e.g. a music player knows that the user will be using music files).

Once a node (or group of nodes) within the graph has been selected as a starting point, Archosum needs to display the organization to the user. It begins by building a result set which consists of all entities associated with the starting point. Then it assesses the size of the result set: if it is too large for users to look through (e.g. over 20), it partitions the result set. For this purpose Archosum identifies categories that are common to multiple entities in the result set. These categories are displayed to the user so he/she can refine their starting point to include them.

If a user starts at the category "Music", Archosum looks to the entities associated with "Music" and tries to find a category that is associated with more than one entity in the result set (that would be too narrow) but less than all (that would be too broad). For example, Archosum might find that categories "Dance", "Country" and "Bob Dylan" are each associated with a subset of entities in the result set. A user can now refine their starting point as a combination of the original starting point and one of the abovementioned categories identified by Archosum. Since a category has been added to the starting point, the result set has been constrained (only files associated with both "Music" and "Dance" will be in the new result set if "Dance" is selected by the user as a refinement). This process is repeated until the size of the result set allows the user to look through the possibilities. In this way, browsing in Archosum is very similar to browsing a hierarchy. By giving the user an interface similar to ones they have already used, we hope to reduce the learning curve for our system so users can start seeing the benefits from their first use.

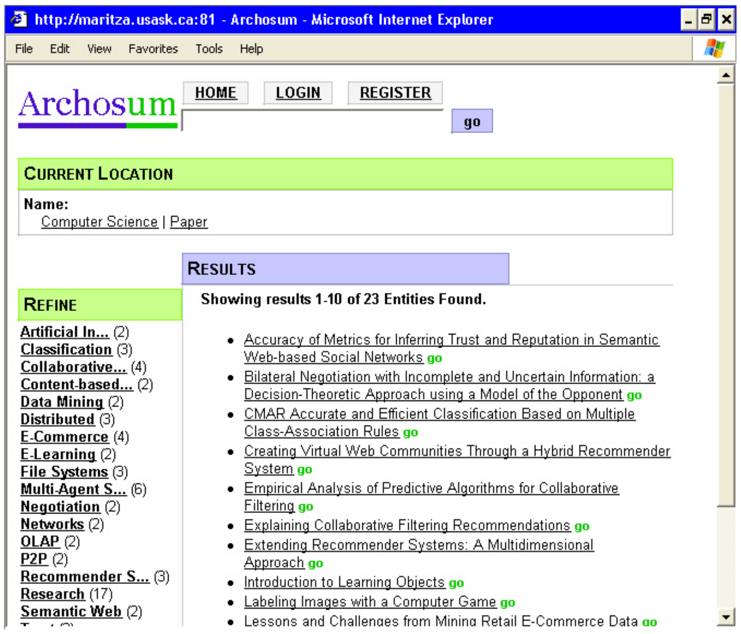
We believe users will create more effective organizations in Archosum because the underlying structure of the organization is abstracted from the user. Users do not need to worry about how relationships are managed. Users simply classify entities into categories as they would in familiar systems (hierarchy and keyword based). Once users build an organization in this way, they can choose to browse or search, but more importantly Archosum can exploit the relationships formed between entities to suggest methods for expanding or refining user queries or show similar and related documents.

5. User Study

The aspirations of our approach are:

- To provide a powerful mechanism for organizing large, heterogeneous collections that is easy to use and easily modifiable.
- To provide both the ability to search directly (by query) and to browse in the organization.
- To allow the comparison and sharing of organizations created by distinct users.

We have not evaluated our approach with respect to all these aspirations yet, but we have started working on the first one by performing a user study to explore how our approach compares with keyword and hierarchical organization approaches. We created a web-based prototype implementation of Archosum, which organizes entities in a fashion similar to a traditional web directory. For the purpose of comparison we created two alternative organizational systems: a keyword based system and a hierarchy based system.


http://maritza.usask.ca:81 - Archosum - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Archosum
HOME
LOGIN
REGISTER
go
CURRENT LOCATION

Name:
Computer Science | Paper

Artificial In... (2)
Classification (3)
Collaborative... (4)
Content-based... (2)
Data Mining (2)
Distributed (3)
E-Commerce (4)
E-Learning (2)
File Systems (3)
Multi-Agent S... (6)
Negotiation (2)
Networks (2)
OLAP (2)
P2P (2)
Recommender S... (3)
Research (17)
Semantic Web (2)

Showing results 1-10 of 23 Entities Found.

- Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-based Social Networks go
- Bilateral Negotiation with Incomplete and Uncertain Information: a Decision-Theoretic Approach using a Model of the Opponent go
- CMAR Accurate and Efficient Classification Based on Multiple Class-Association Rules go
- Creating Virtual Web Communities Through a Hybrid Recommender System go
- Empirical Analysis of Predictive Algorithms for Collaborative Filtering go
- Explaining Collaborative Filtering Recommendations go
- Extending Recommender Systems: A Multidimensional Approach go
- Introduction to Learning Objects go
- Labeling Images with a Computer Game go
- Lessons and Challenges from Mining Retail E-Commerce Data go

Figure 3: Screen shot of the Archosum prototype

To compare our approach with hierarchical and keyword based organizations, we need to define some metrics on which we could compare the three systems. Precision and Recall are standard metrics for comparing information retrieval systems defined by the Text Retrieval Conference (TREC) [9]. Information retrieval systems are typically "user-neutral": their index is based on the collection; therefore the organization is a function of the collection. In our system the user is active in building the organization; thus the organization is a function of both the collection and the user.

Since we have introduced a new variable (the user), we evaluate the process of building an organization and finding entities in this organization as two separate steps. In our experiment, we have only considered the first step of this process, the organization step. The second step - showing that the organization allows more effective search will be explored in future studies. To evaluate the first step, we define the following metrics on which we can judge an organizational system:

Organizational Metrics:

1. **Cross User Similarity** – the amount of similarity between the organizations created by different users in a system
2. **Depth of entity vs. Breadth of category** – This describes the continuum of classification. In one extreme all entities could be classified in one category (high breadth of category). In the other extreme, entities could each be classified in a unique, very specific category (high depth of entity).
3. **Time to Classify** – the amount of human time required to build an organization.

In our experiment, 12 users were asked to organize 50 documents in each of three organizational systems over a 10 day period. We compiled a collection of 150 web addresses from 6 general categories including: news, people, companies, movies, musicians and academic papers. For each organizational system we selected 50 entities with an approximately even proportion of representation of each category in each system. To minimize user learning between systems, the entities in each system had no overlap.

Each user filled out pre and post experiment questionnaires. For each system, users went through a four step process:

1. Short training for how the system works.
2. Chance to browse a working organization in that system.
3. Training for how to organize documents in a system.
4. Classification of 50 entities using the system within 30 minutes.

All participants completed this process for each of the three systems (hierarchy-based, keyword-based and Archosum). We analyzed the timing of actions and resulting organizations in each system.

6. Results

Although each system uses a different mechanism to organize entities, we introduce the quasi-formal general description of an organizational system which allows us to compare the results obtained with the three systems:

In each system we have a set of entities ($E_k \in E$) and a set of users ($U_i \in U$). Each user (U_i) creates a set of categories ($C_{ij} \in C_i$). Building an organization in any of the systems involves placing entities into categories ($E_k \in C_{ij} \in C_i$ for U_i). In each system there are 50 entities (E_k : $k \in [1,50]$), 12 users (U_i : $i \in [1,12]$) and some number of categories created by each user (C_{ij}: $i \in [1,12]$ $j \in [1,\lambda_i]$, where λ_i is the number of categories created by U_i).

Time to Classify

The time to classify the i-th entity for a user ($TC(E_i, U_j)$) was measured as the time between seeing the i-th entity and classifying it. Since users might take a break or leave, if the time was over a threshold (10 minutes), we ignored that time. Times over the threshold were rarely reached (less than 0.5% of classification times were ignored).

$$TC(E_i, U_j) = [Time\ E_i\ Classified\ by\ U_j] - [Time\ E_i\ was\ shown\ to\ U_j]$$

$$TC(E_i)_{ave} = \frac{\sum_j TC(E_i, U_j)}{|U|}$$

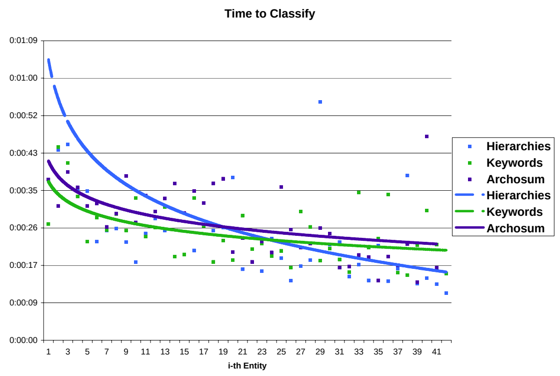


Figure 4: Time to Classify the i-th entity

We can see that in each system users took less time to classify entities as they gained experience with each system. The hierarchy curve is slightly skewed by two users who did not fully understand the interface for this system. These users spent most of their time on the first few entities and rushed on the rest, thus producing the above curve.

Depth vs. Breadth

The depth of an entity ($D(E_i, U_j)$) measures the number of categories with which a user associated that entity. Once we have the depth of each entity, we computed the average depth across all entities for each user and plotted that on the left graph in Figure 4.

$$I(E_i, C_{jk}) = \begin{cases} 1 & when\ E_i \in C_{jk} \\ 0 & otherwise \end{cases}$$

$$D(E_i, U_j) = \sum_k I(E_i, C_{jk})$$

$$D(U_j)_{ave} = \frac{\sum_i D(E_i, U_j)}{|E|}$$

Note: Since all users classified the same set of entities, |E| = 50.

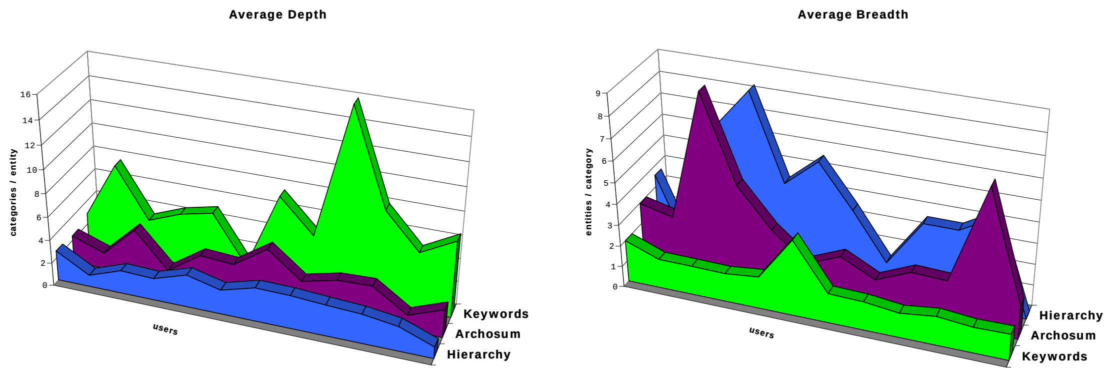


Figure 5: Average Depth and Breadth for the three organizational systems

The breadth of a category ($B(E_i, U_j)$) measures the number of entities classified in that category for a given user. The right graph in Figure 4 plots the average breadth for each user in each system.

$$B(C_{ji}, U_j) = \sum_k I(E_k, C_{ji})$$

$$B(U_i)_{ave} = \frac{\sum_k B(C_{ji}, U_j)}{|C_i|}$$

The results in Figure 5 show that the keywords organization had a higher average depth (more keywords associated with an entity) while the hierarchical organization had a higher average breadth (more entities classified in a given category). These results are not surprising: it is harder for the user to create many hierarchical categories (creating a new category requires identifying the parent category), therefore the few that are created tend to be associated by users with more entities (higher average breadth for hierarchies). Conversely, creating new keywords is very easy, and users easily forget keywords that they have previously used, therefore they associated many keywords with each entity (higher average depth for keywords). Archosum appears more similar to hierarchies with respect to these two measures than to keywords.

Cross User Similarity

Through this metric we are trying to explore the similarity between organizations built by different users. To examine the similarity, we can begin by looking at the categories created by different users (see Figure 6). We believe the labels (names) given by users to the categories provide little benefit in comparing organizations, as they are highly dependant on the vocabulary of a given user; thus, we have chosen to compare the similarity between the patterns of relating entities to categories across users and ignore the labels (names) as provided by users.

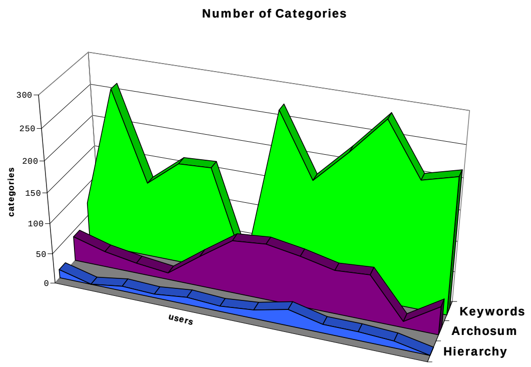


Figure 6: Number of categories specified by each user

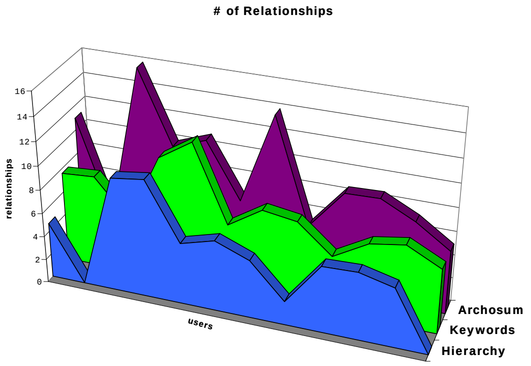
In each system, the users provided different numbers of categories, thus enumerating and comparing them would be difficult and non-intuitive. To facilitate the comparison, we introduce the concept of "relationships".

Each user (U_i) has a set of relationships ($R_{ijk} \in R_i$) that relate two entities (E_j and E_k). A relationship (R_{ijk}) exists if the user (U_i) has placed two entities (E_j and E_k) in a common category (C_c). All relationships are symmetrical, thus $R_{ijk} \in R_i$ implies $R_{ikj} \in R_i$.

The benefit of introducing relationships is that we can easily enumerate the distinct relationships from a given entity (an entity E_i can be related by a user U_i at most to the all other entities, thus $|R_{ij}| \leq |E|-1$). If we know all the relationships (R_{ij}) from an entity (E_j) established by a user (U_i), we can compare this set to the set of relationships (R_{kj}) established for the same entity (E_j) by each user (U_k: $k \neq i$).

First, we will count the average number of relationships made by each user in each system. This tells us how likely users are to relate different entities using each system. The left graph in Figure 6 plots these values.

$$R_{i\ ave} = \sum_j \frac{|R_{ij}|}{|E|}$$



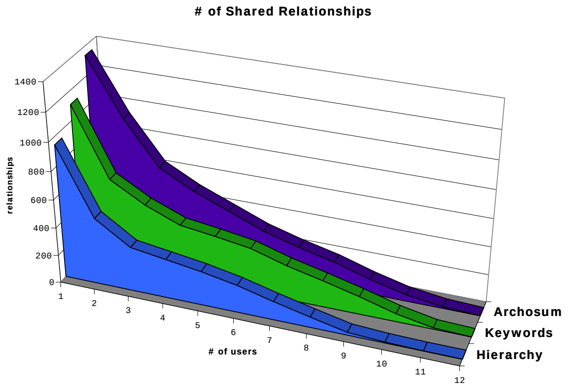


Figure 7a & 7b: The Number of Relationships between entities and the Number of Shared Relationships in the three organizational systems

The most important value is not the number of relationships established, but rather the number of relationships established by a user that are shared by other users. We measure this by constructing the weighted common set of relationships RC.

$$J(U_i, E_j, E_k) = \begin{cases} 1 & when\ R_{ijk}\ exists \\ 0 & otherwise \end{cases}$$

$$RCij = \sum_k J(U_i, E_j, E_k)$$

For example, if U_1 placed E_1, E_2, E_3 and E_4 in the same category($C_{1,5}$) while U_2 placed E_2 and E_4 in $C_{2,3}$ and E_3 and E_4 in $C_{2,5}$ then the weighted common set (RC) would consist of the following values:

$RC_{1,2} = 1, RC_{1,3} = 2, RC_{1,4} = 1, RC_{2,3} = 1, RC_{2,4} = 2, RC_{3,4} = 2$
Note: the symmetric equivalents are not shown but would also exist in RC.

Now that we have compiled the weighted common set, we can extract the number of relationships shared across some number of users (x). In Figure 7b, we have plotted the number of shared relationships for eleven of the twelve users , $x \in [1,12]$. As can be seen in Figure 7a & 7b, users created more relationships in Archosum and moreover, there was a higher number of shared relationships in Arhcosum across multiple users.

7. Discussion

Our results show the differences between the three organizational systems tested. These differences translate into distinct performance characteristics for each system. These characteristics were identified in our small experimental data set and would surely be amplified in a full-scale implementation; thus, making an incorrect choice of organizational system could have far reaching consequences. It is therefore very important to know how the strengths and weakness can be exploited or avoided when implementing an organizational system.

While a keyword based approach provides high depth (i.e. an entity is typically associated with many keywords), it suffers from low breadth (low reuse of keywords). The opposite is true for hierarchical systems: few hierarchical categories are used to classify many entities; therefore each category is associated with many entities. This result is easily explainable as keywords are unstructured and easy for users to specify, but therefore easier to forget when users classify multiple entities. This leads to low rate of reuse of keywords, and smaller number of relationships among entities in the keyword based approach. Meanwhile, hierarchical directories are rigid in structure, but cumbersome to manipulate. The Archosum approach uses categories that introduce structure, but no hierarchy and helps users create more balanced organizations with moderate breadth and depth that are easier to create and update hierarchical organizations..

Since the ideal level of depth and breadth is determined by the purpose and intended audience for a given organization, we cannot state that any organizational system is inherently better than another. However, we can state that for any situation there is an ideal choice since each system has a distinct behavior.

Our results illustrate that even when users are unfamiliar with an interface and system, they are able to quickly learn how to use that system to build effective organizations. The interface for Archosum was designed to resemble familiar systems to reduce the penalties of user learning time. By reducing the learning time to create an organization, we have also decreased the time a user must wait to utilize the unique benefits and features of our system.

When looking at the cross user similarity metrics we observed major differences between the three systems. In the keyword-based approach, users specified over four times as many categories per entity yet established fewer relationships between entities than in the hierarchical system or Archosum. The number of relationships shared across users in Archosum was also higher than in the hierarchical system.

In an increasingly connected world, it seems archaic to consider only a single user's collection. The cross user similarity metric allows us to consider how well we can combine the collections of two or more users to build a larger collection or group of collections. The benefit of being able to combine collections is not evident in our experiment as all users classified the same set of entities. In a real world situation, users would be much more likely to organize document collections of inconsistent contents.

Consider a real world situation in which some user U_1 has organized a collection of entities E^1. Is there a way that U_1 can automatically find new entities that are similar to those in a given category of their current organization? If we examined the collection of entities E^2 organized by U_2, we could find the intersection of both collections ($E^1 \cap E^2$). From the intersection, we could develop a pattern based on the similarity between both users' organizations that we could use to find entities in E^2 organized by U_2 to be similar to those entities organized by U_1 .

Obviously the ability to intelligently combine different semantically organized collections is not trivial, Thus, a system that can identify and represent similarities between organizations of different users may provide a way to organize large collections in a more coherent way than ever before. It is this ability to fuse collections that gave Archosum its name. Archosum is the combination of two Latin words, *archos* meaning ruler and *visum* meaning view. The analogy is that by fusing organizations we can create an organization (or view) that is customized and controlled by each user (the ruler). Archosum may provide a way for humans to organize collections of constantly increasing size without increasing the complexity of organization.

8. Conclusion

Our experiment and analysis evaluates the process of creating an organization of entities in three different systems. The results of this experiment show significant advantages for each system in certain situations. It therefore seems that alternatives to the standard keyword and hierarchy-based systems may be of significant value and warrant more investigation.

This experiment also casts a new focus on the creation of organizations. TREC has evaluated information retrieval systems for many years based on their query performance (judged by recall and precisions), but has not directly examined the creation of organizations since this is usually automatic. The metrics we have presented allowed us to compare systems that do not automatically generate indices, but rely on users to classify the contents of their collections.

With the recent surge of development in consumer file systems and their organizational systems, few large implementations utilize systems that are not keyword or hierarchy-based, yet alternative systems are available in academia [5,9]. We hope our results may help the development of future alternative organizational systems as well as their use in practical implementations.

We believe Archosum could be applied to environments where many users are attempting to find new entities related to their current organization based on the recommendations of other users. Of specific interest would be systems like Comtella [24], a peer-to-peer application developed to share papers in academia. Archosum would allow each user to organize their collection of papers as he/she chose (rather than relying on some centrally determined hierarchy), and fuse these organizations based on similarities to discover new papers relevant to each user. Furthermore, Archosum would better facilitate cross-field work as a paper could be classified into more than one category (and a search could be conducted on more than one category).

9. Further Research

The first avenue of further research is to expand our experiment to include the second step of an organizational system, the actual retrieval of documents given an index in each of the presented systems. If we could test all three systems using the metrics defined by TREC, we might discover new evidence to support the implementation of a particular organizational system, or possibly some criteria for building an even better system. We believe that our work on metrics for building organizations may be used for performance comparisons by other organizational systems as they are developed.

The Resource Description Framework (RDF) is a language designed by the World Wide Web Consortium (W3C) for representing information about resources in the World Wide Web [1]. RDF uses triples to describe resources and relationships between resources. We believe that our approach could easily be implemented using RDF and possibly even expanded beyond simple associations to include multiple types and strengths of relationships using RDF.

An RDF implementation of Archosum brings many new features but also many new questions:

1. **What is the nature of relationships?**
   In Archosum, entities are related to categories in a directed, equally weighted sense. RDF allows us to build unlimited types of relationships including undirected and weighted relationships. Are these concepts intuitive and do they give us a better tool with which to describe relationships, or do they allow us too much freedom?
   VennFS, explores the benefits of relationship weights by organizing documents in a plane, documents that are far apart are loosely related while those close together are heavily related [5].
2. **Should organizations be concerned with time?**
   If we can add properties to relationships, we could easily expand them to exist for only a specific time interval. Should relationships decay over time to create a process of natural selection, could this be used to build an inherent junk or SPAM deterrent?
3. **How well can machines classify documents?**
   Can we combine the concepts presented in this paper with those of machine classification? Can the system provide classification suggestions, then adapt them based on user acceptance and further suggestions?
4. **Can collections organized by different users be shared seamlessly?**
   We will explore sharing collections and automatic search across collections organized subjectively by different users using relationships between entities as defined by the measures presented in this paper.
5. **Can Archosum cope with disjoint entity sets?**
   An organization can be considered as a graph (existential and abstract entities as vertices; relationships as edges). How can a user browse an organization if we have two or more disconnected sub-graphs? When finding similarities between organizations, Archosum begins by considering entities common to both. If no common entities exist no similarity will be found. To what extent would this problem arise in practice, will users have truly disjoint collections; and is there a method by which we can cope?

References

1. Berners-Lee, Tim. (1989). "Information Management: A Proposal". CERN. Internet Available: http://www.w3.org/History/1989/proposal.html (23 Oct 2004).
2. Brin, Sergey and Page, Lawrence. (1998). "The Anatomy of a Large-Scale Hypertextual Web Search Engine." WWW7 / Computer Networks 30: 107-117. http://www-db.stanford.edu/~backrub/google.html (8 Mar 2005).
3. Bush, Vannevar. (1945). As We May Think. The Atlantic Monthly, July 1945: 101:108. Internet Available: http://sloan.stanford.edu/mousesite/Secondary/Bush.html (23 Oct 2004).
4. Dalianis, Hercules. (2002). "Evaluating a spelling support in a search engine." NLDB 2002, The Eight International Workshop on the Applications of Natural Language to Data Bases (June 27-28, 2002). Internet. Available: http://www.nada.kth.se/~hercules/papers/SpellingIR.pdf (6 Mar 2005).
5. De Chiara, R., Erra, Ugo., Scarano, V. (2003). "VennFS: A Venn-Diagram File Manager." IEEE International Conference on Information Visualization.
6. "Data Access and Storage Developer Center: Building WinFS Solutions." Microsoft Developer Network. http://msdn.microsoft.com/data/winfs/ (23 Oct 2004).
7. Fellbaum, Christiane, ed. (1994). WordNet: An Electronic lexical Database. Cambridge: The MIT Press. http://mitpress.mit.edu/book-home.tcl?isbn=026206197X (23 Oct 2004).
8. Giampaolo, Dominic. (1998) Practical file system design with the Be file system. San Fransisco: Morgan Kaufmann.
9. Gifford, D., Jouvelot, P., Sheldon, M., O'Toole, J. (1991) "Semantic File Systems." Operating Systems Review, v25 n5.
10. "Google Desktop Search Beta." Google. Internet Available: http://desktop.google.com/ (6 Mar 2005).
11. "Google Stock Information Summary." Yahoo! Finance. http://finance.yahoo.com/q?s=goog (23 Aug. 2004).
12. Herman, G. Melan¸con, and M. S. Marshall. (2000). Graph visualization and navigation in information visualization: A survey. IEEE Transactions on Visualization and Computer Graphics, 6(1):24–43, 2000. Internet Available: http://citeseer.ist.psu.edu/herman00graph.html (23 Oct 2004).
13. "Hierarchy." *Wikipedia, the free encyclopedia*. http://en.wikipedia.org/wiki/Hierarchy (24 Oct 2004).
14. "History of The Open Directory." *DeemozWatch.* Feb. 2004. http://www.deemozwatch.org/history.htm (15 Aug. 2004).
15. "Mac OS X Tiger: Spotlight." *Apple Computer Inc*. 28 Jun. 2004. http://www.apple.com/macosx/tiger/spotlight.html (23 Oct 2004).

16. Manola, F., Eric M. ed. (2004). "RDF Primer: W3C Recommendation." World Wide Web Consortium. http://www.w3.org/TR/rdf-primer/ (23 Oct 2004).
17. Nickell, Seth. "A Cognitive Defense of Associative interfaces for Object Reference." http://www.gnome.org/~seth/storage/associative-interfaces.pdf (23 Oct 2004).
18. "ODP and Yahoo Size Charts." *Geniac.net*. 10 Jan 2004. http://www.geniac.net/odp/ (15 Aug 2004).
19. "Open Directory Project." *Netscape*. http://www.dmoz.org/ (24 Oct 2003).
20. Robertson, S.E., Walker, S., Zaragoza, H. (2001). "Microsoft Cambridge at TREC-10: Filtering and web tracks." NIST: Text REtrieval Conference 2001, pg. 378. http://trec.nist.gov/pubs/trec10/papers/msr_cambridge.pdf (24 Oct. 2004).
21. Sayers, Craig. (2004). "Node-centric RDF Graph Visualization." HP Laboratories: Mobile and Media Systems Laboratory. http://www.hpl.hp.com/techreports/2004/HPL-2004-60.pdf (23 Oct 2004).
22. Soules, Craig, Granger, Gregory. (2003). "Why Can't I find my files? New methods for automating attribute assignment." Hot Topics in Operating Systems, May 2003: 115-120. USENIX Association. http://www.pdl.cmu.edu/PDL-FTP/Storage/hotOS03.pdf (23 Oct 2004).
23. Van Rijsbergen, C. J. (1979). Information Retrieval. London: Butterworths. http://citeseer.ist.psu.edu/vanrijsbergen79information.html (23 Oct 2004).
24. Vassileva J. (2002). "Supporting Peer-to-Peer User Communitites, in R. Meersman, Z. Tari et al. (Eds.) 'On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE'" Coordinated international Conferences Proceedings, Irvine, 29 Oct – 1 Nov 2002, Springer Verlag: Berlin-Heidelberg, 230-247.
25. Zakon, Robert Hobbes. "Hobbes' Internet Timeline v7.0." *Zakon Group*. http://www.zakon.org/robert/internet/timeline/ (17 Aug 2004).