COLLABORATIVE
Organizing Document Repositories Efficiently

Electronic document repositories allow knowledge communities to store and share documents. To be useful, a large document collection must be organized: some repositories organize documents into a hierarchy of categories while others organize documents into a hyperlink network. No matter what the technique, a great amount of effort is required to keep a large, growing document repository well organized.

We have developed a repository system that organizes documents using both hierarchies and hyperlinks, through individuals’ collaborative efforts. The
system allows an individual user or an organization to organize a subset of a document repository into a personal, local hierarchy, which can also be shared with others. The system utilizes the structural knowledge embedded in these local hierarchies to build a global hierarchy containing all documents in the repository. Utilizing these hierarchies, the system also builds a hyperlink network that covers the entire document repository. The system thus provides valuable document structures to enhance users’ capability to use a large repository.

The structure among documents in a repository is just as important as the content of the documents themselves for tasks such as information storage and retrieval, learning, and knowledge creation. Usable structures in a repository help people navigate from one document to another, understand the relationships among documents, learn the overall domain, and find information more quickly. Effective organization of documents is critical to the success of a document repository.

Many repositories are organized by a hierarchy of categories known as a taxonomy. A hierarchical organization is appropriate for most knowledge tasks. The complexity of a domain is often hierarchical in its nature, as are human cognitive structures [1]. A hierarchy is also efficient, as $n$ documents may be placed in a hierarchy with depth of mere $\log(n)$. Hierarchies allow a document repository to be queried just as with a hierarchical database. One major problem with hierarchies, however, is that a document cannot be easily found through the hierarchy if the document is mis-categorized. This can happen when the repository has a preconstructed taxonomy, and as time passes, the taxonomy becomes outdated by the new documents coming into the repository. This can also happen when the categorizer and the person searching for documents have different perspectives on the topical relationships among documents.

Another way to organize documents is through hyperlinks. The resulting network structure is not good for systematic inquiries, but convenient for casual navigation. Hyperlinks can be created by authors at the time of document creation. Some
recent systems also allow links among documents to be annotated by readers, or generated automatically based on document similarity and/or user navigation. As the number of hyperlinks grows, the average path length between any two documents in a hyperlink network tends to be \( \log(n) \), where \( n \) is the total number of documents [7]. However, a major problem with relying on hyperlinks to organize a repository is that related documents may not be adequately hyperlinked, as authors and readers often are insufficiently motivated to add and maintain hyperlinks.

In an enterprise setting, knowledge workers must be able to retrieve information in a systematic manner. Most state-of-the-art corporate document management systems such as Documentum and OpenText utilize the taxonomy approach. In these systems users (company employees) contribute documents to a common hierarchy based on predetermined categories, which may be periodically adjusted. While many companies have made extensive investments in implementing these systems and setting up the initial hierarchies, they often find it cost prohibitive to keep a large, growing document collection well categorized. When document repositories suffer from lack of organization, which they frequently do, it in turn reduces users’ interest in using them and the ability of sharing information through them. Even if the documents are well categorized on the whole, because the common taxonomy cannot fully match one’s individual perspective, users often must file the same document twice—one in the shared repository and once in their local file systems. The additional filing effort involved and the difficulty of finding information often deter individuals from contributing to the repository.

Collaborative Structuring

The key to organizing a large document repository is to provide useful information to its users, while minimizing maintenance cost. Theoretically, this can be achieved by utilizing the collaborative effort of its users. Each user can organize a subset of the document repository. Together, the user community keeps the whole repository well organized. Allowing individual users to organize documents also elicits structural knowledge—the knowledge of relationship among documents, which comes from individual perspectives and experiences. The valuable structural knowledge can be shared in the repository in addition to documents themselves.

While plausible, the preceding concept of collaborative structuring has not been explored by today’s repository systems. One key challenge is that well-organized local document collections do not directly lead to a well-organized repository at the global level. We have designed a document repository system that addresses this challenge and proves the feasibility and utility of the collaborative structuring concept.

In short, our Collaborative Structuring System allows an individual (person or organization) to organize a subset of documents in a repository into a local hierarchy and share the hierarchy with others. The system generates a “consensus” hierarchy that merges all such local hierarchies. The consensus hierarchy provides a common and emergent view of all documents. The system also automatically generates hyperlinks between documents, categories, and repository users, and hence creates a comprehensive hyperlink network covering the repository. Utilizing collaborative efforts, our system provides valuable document structures that support both systematic inquiries and casual exploration, with low system maintenance costs. We have evaluated the system in several document repositories supporting classes and research groups in large public universities. Through data collected from click-stream analysis, online rating mechanisms, questionnaires, and user interviews, we confirmed the feasibility as well as the benefits of collaborative structuring.
**System Design and Benefits**

Our system is designed on top of a popular open-source document repository system, the Everything Engine (everydevel.com), hereafter referred to as E2. E2 allows users to contribute HTML-based content as well as any other type of electronic documents such as images or PDF files. E2 allows users to insert hyperlinks among documents. It also allows administrators to create categories and allows users to contribute documents into these categories. E2 has advanced search functionality including both full-text search and attribute-based search. Overall, E2 provides a comprehensive set of features comparable to that of other state-of-the-art document management systems. We enhanced E2 with support for building, sharing, and merging local document hierarchies, and generating hyperlinks among documents automatically. Figure 1 is a screenshot of our Collaborative Structuring System based on E2.

The Hierarchies section shown in Figure 1 allows a user to switch between different document hierarchies. A user can organize documents into a local hierarchy (“personal” in the Hierarchies section). The process of organizing existing documents in the repository into local hierarchies is similar to adding and organizing bookmarks in a Web browser. A user can also directly add the document to the local hierarchy when contributing new documents to the repository. The same document, of course, can be categorized differently in many different local hierarchies. In our deployment, 100% of the users utilize local hierarchies to organize documents. Besides saving time accessing the documents repeatedly, local hierarchies also assume an important role in learning and knowledge creation. For example, some users are willing to expend a great amount of effort organizing documents into their local hierarchies, even under imminent deadlines such as delivering a research paper. The process of organizing references and related documents itself is important for users to organize their thoughts and produce the research paper. Users who use local hierarchies more extensively also tend to contribute more documents to the repository. Local hierarchies give users the control of their personal workspaces and relieve users from filing the same document twice, which in turn encourage users’ contributions.

Users can share their local hierarchy with others and utilize others’ local hierarchies (“others” in the Hierarchies section). Browsing these shared local hierarchies is just like browsing network file systems—the access to hierarchies or the categories within it is subject to the permissions set by hierarchy owners. Just as with faceted classification schemes [2], these shared local hierarchies provide users multiple ways to locate information. These shared hierarchies also lead users to high-quality documents, similar to other collaborative filtering systems [6]. Based on online feedback data (see the Feedback section in Figure 1), the documents organized into local hierarchies are voted as useful documents approximately twice as often as average documents.

Local hierarchies elicit users’ tacit structural knowledge. Sharing these hierarchies allows repository users to share this valuable knowledge in addition to the content of documents. In our deployment, experts’ hierarchies are more frequently used by others than novices’ hierarchies, because they contain more valuable structural knowledge. Interestingly, some users like to establish their identities in a knowledge community through sharing their local document hierarchies. These users spend large amounts of time building extremely well-organized document hierarchies that can benefit anyone who wants to explore a domain in depth. Shared hierarchies are also often used for collaboration. For example, to divide and conquer a group knowledge task, individuals use local hierarchies to address parts of the problem and share these local hierarchies among group members. Users often grant write permissions to their team members, enabling the whole team to collaboratively maintain the same document collection.

The system automatically generates a consensus hierarchy (“consensus” in the Hierarchies section) that merges all local hierarchies. A local hierarchy only includes a (usually small) subset of the document collection. In contrast, the consensus hierarchy includes all documents in the repository. In essence, the consensus hierarchy is generated from an agglomerative hierarchical clustering [4] algorithm, utilizing the document-category mappings contained in local hierarchies.

![Figure 1](image1.png)

*Figure 1. The inspectors mechanism of documents, categories, and users.*

![Figure 2](image2.png)

*Figure 2. The inspection mechanism of documents, categories, and users.*
archies. Each document is represented by a vector indicating which categories contain the document. The documents are merged into clusters based on the similarity between document vectors, and clusters are further merged until the entire document collection becomes a single cluster. Reversing the merging steps results in a top-down hierarchy containing the entire document collection, which we call a consensus hierarchy. For non-overlapping local hierarchies—hierarchies with no documents in common—the consensus hierarchy simply concatenates them. For overlapping local hierarchies with conflicting ways of categorizing documents, the consensus hierarchy will represent the common or the most prominent perspective. Most global hierarchies in today's document repository systems are predefined and relatively static. In contrast, the consensus hierarchy contains an up-to-date topical map of the document repository and can be regenerated dynamically. Users, especially novices in a knowledge community, prefer to use the consensus hierarchy as the starting point to explore the repository. Some users check the consensus hierarchy regularly to keep abreast of all documents that have become available on a given topic. Some users follow new trends in a knowledge community by monitoring changes to the consensus hierarchy.

As we noted earlier, using a single global hierarchy to organize a document collection risks a miscategorization problem, often due to conflicting individual perspectives and out-of-date categories. In our system, individuals are allowed to personally organize documents from their own perspectives. The system-generated consensus hierarchy can be frequently updated to represent up-to-date topics emerging from local hierarchies. Of course, user errors and bad judgment in categorization may result in miscategorization and poor hierarchy quality. However, the system-generated consensus hierarchy has been found to be quite robust against a small percentage of miscategorization problems in local hierarchies. In one experiment, after we randomly replaced 1% of the documents in local hierarchies across many trials, the top two levels of categories in the resulting consensus hierarchies remain the same.

The system generates a hyperlink network that enhances users' navigational capabilities. The system automatically generates hyperlinks between documents, categories, and users using a mechanism that we call inspection (see Figure 1 and Figure 2). The system generates a home page for each user containing all documents authored by that user as well as the user's local document hierarchy. Each category in every hierarchy has a system-generated page containing hyperlinks to its parent category, its owner, and also to all documents within the category. When displaying a document, the system dynamically generates hyperlinks to the document's author, to the categories that contain the document, and to the authors of these categories (see Figure 1). Following these hyperlinks appended to a document will likely lead to documents and users of similar interest. The average path length between any documents or users in a repository, or so-called degrees of separation, is roughly the logarithm of the repository size. Therefore any users or documents are just a few steps away through these inspection links.

Users are found to frequently traverse these system-generated hyperlinks, more often than traversing the hyperlinks created by document authors. While a user can navigate the repository through various hierarchies, only one hierarchy can be utilized at a time, which represents the relationships among documents from a single perspective. In contrast, the system-generated hyperlinks allow a user to navigate from one document to multiple hierarchies that contain the document. Such a hyperlink structure across multiple
hierarchies has been studied as “multitrees” and has many advantages for information access and reuse [3]. Besides permitting users to explore documents and hierarchies, the hyperlinks help readers to reach experts in a certain domain, which often is much more important than accessing the documents. One recurring pattern we observed is that a user comes to know an expert as follows. When a user sees an interesting document or category, the user explores the system-generated hyperlinks to see who the author or the categorizer is. If the user finds that many interesting documents in a field are authored or categorized by the same individual, this individual is identified as an expert in the field. From that point on, the user tends to directly access the expert’s home page to browse the documents authored or categorized by the expert. In a way, our system functions as a social intermediary for people to get to know each other.

As described previously, our system organizes documents through both hierarchies and hyperlinks (see Figure 3). In addition to supporting centrally controlled global taxonomies as most state-of-the-art repositories do, the system allows users to build local hierarchies tailored to individual needs and share these local hierarchies. The system automatically merges these local hierarchies into a global hierarchy. The system also generates a hyperlink network among documents, categories, and users of related interests. All these features encourage user contribution and enhance users’ ability using the repository.

Conclusion
We have discussed the concept of collaborative structuring, as well as the design and benefits of such a system. Collaborative structuring is not meant to replace, but rather, to improve the way documents are organized in existing document management systems. The design can be used to support cross-organizational data sharing, which has become a significant problem in recent years. One particularly useful application may be to merge a collection of document repositories by generating a consensus global hierarchy from the hierarchies in these individual repositories. Collaborative structuring enhances people’s ability to interact with each other through a repository system, which is critical to any knowledge community.

Our research relates to others in the fields of collaborative systems, knowledge management, and information retrieval. The concept of collaborative structuring is closely related to collaborative filtering, also called social filtering or recommendation systems [6]. While many other filtering systems have analyzed people’s Web behavior or preferences, our system utilizes individuals’ structural knowledge embedded in local hierarchies. The technique we developed to merge local hierarchies into a consensus hierarchy can be used for ontology management [5], a growing field that deals with mapping, merging, evolving, sharing, and querying knowledge representations. Our consensus-building technique can benefit from advances in document clustering research such as clustering algorithms and cluster-labeling techniques.

There are also many questions left to explore. For example, both free-riding and privacy issues involved in sharing local hierarchies must be explored further. While evaluation confirmed the usefulness of the consensus hierarchy, it also suggested the potential for improvement, particularly in the area of automated category labeling. How semantic analysis, link analysis, and manual efforts may be combined with hierarchical clustering to improve the quality of a consensus hierarchy also requires investigation.

We believe collaborative structuring is an important area to both researchers and practitioners. For example, one critical challenge recently brought to the IT field was how to organize documents across dozens of organizations in the intelligence community, and how to share intelligence analysts’ knowledge about the relationships among these documents. A collaborative structuring system may be an effective and efficient method to address such a challenge.

References

HARRIS WU (hwu@odu.edu) is an assistant professor of information systems and technology at the College of Business and Public Administration, Old Dominion University in Norfolk, VA.

MICHAEL GORDON (mgordon@umich.edu) is a professor of business and information technology at the Ross School of Business, University of Michigan, Ann Arbor, MI.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

© 2007 ACM 0001-0782/07/0700 $5.00