# Collections: Flexible, Essential Tools for Information Management

**David R. Karger**
MIT CSAIL
200 Technology Sq., Cambridge, MA 02139 USA
karger@theory.lcs.mit.edu

**Dennis Quan**
IBM T. J. Watson Research Center
1 Rogers Street, Cambridge, MA 02142 USA
dennisq@us.ibm.com

## ABSTRACT
While collections—aggregation mechanisms such as folders, buddy lists, photo albums, etc.—clearly play a central role in information management, the potential benefits of true first class support for collections are masked by disparate implementations that force users to pay attention to technological distinctions such as application, format, and protocol. We argue that systems should expose a single unified concept of collection and that concepts such as portals, cross-application projects, customized menus, and e-mail-task unification come about naturally as a result of our abstraction. In addition, uniform support for collections brings about a new set of capabilities for supporting creative processes. We discuss a prototype implementation of this abstraction in our Haystack system, give several examples of why we believe our abstraction is useful in everyday information management, and present some preliminary results from user studies that support our hypotheses.

### Author Keywords
Collections, folders, portals, taxonomy, semistructured data.

### ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

## INTRODUCTION
Knowledge workers deal with collections of items on a constant basis, both in the physical world and on their computers. Collections, whether they take the form of file system directories or piles of paperwork, are important tools for helping people get their work done. In particular, two aspects of collection management are focused on in this paper. First, all forms of collections help people navigate and retrieve information. For example, collections such as menus (collections of commands), portals (collections of related pieces of content), public taxonomies (hierarchical collections), and directories (collections of files) all help users in locating specific objects. Second, some collections also play an important role in structuring and organizing knowledge. Users can employ mutable collections such as e-mail folders, to-do lists, and photo albums to categorize objects. These collections are then navigated through later during retrieval.


*CHI 2004,* April 24–29, 2004, Vienna, Austria.
ACM 1-58113-703-6/04/0004.

While support for collections appears in nearly every information management application, today's implementations possess numerous shortcomings that hinder computer-based collection management from achieving its true potential. For example, systems such as e-mail clients and file system browsers (e.g., Windows Explorer) expose rigid, strictly hierarchical collection mechanisms in which it is assumed that objects will usually be placed into one folder at a time. However, often one cannot cleanly decide which single collection is the correct home for a given arbitrary object [14], undermining the fundamental assumption of collection construction interfaces such as the File Save and Add Bookmark dialog boxes.

One consequence of this rigidity is the feeling of heavyweight "commitment" associated with collection creation [4]. One of Whittaker and Sidner's test subjects remarked on their e-mail system: "I wish I viewed creating a category as a lightweight activity. And for some reason I don't…it seems like you know the more of them I create, the harder it is to find any of them that are there." For the so-called "filers" of the world, collection creation may occur rarely enough for this not to be a problem; however, for "pilers", collections may need to be created on a whim, and convenient construction mechanisms are essential [3]. Similarly, to encourage the freeform grouping of objects for creative purposes, it needs to be easy to create collections to hold intermediate sets of objects [11].

Another significant problem is that objects needing to be filed together often originate from different applications that require their objects to be organized under separate regimes. Media players, e-mail clients, task managers, file systems, and photo browsers often have their own organization schemes that do not interoperate, making it inconvenient to create a common filing system for e-mail messages, to-do items, spreadsheets, and photos. (One could save all such items as files or e-mails, but one would lose the type-specific tools present in particular applications; this point is discussed further below.) A primary contribution of systems such as TaskMaster [1] and ReMail [12] is their success in enabling pairwise collection management interoperability between specific domains (e-mail/tasks and e-mail/instant messaging, respectively). The common theme underlying systems such as these points toward the need for a more universal solution.

Similarly, websites for creating portals—collections of objects created for the purposes of visualizing them together—are often specialized to work with specific kinds of presentation elements only.

Finally, visualization styles are often highly coupled to the data model of an application, leading to what is known as "habitat" formation [13]. Since creating collections to hold e-mails, documents, tasks, and other objects can be difficult, users frequently rely solely on the application that presents the best interface for the kind of collection management needed (the so-called "habitat") and "coerce" objects into the type supported by that application. In many cases, e-mail is chosen as a habitat because of the relative abundance of e-mail messages with respect to other kinds of objects and because the inbox metaphor has been somewhat successfully used for the task management tasks performed in daily life. Of course, whenever an application is coerced into managing information it was not intended to handle, users experience a degraded quality of interaction, which should be avoided.

As we have highlighted above, a number of the problems with today's applications' support for collection management arise from dealing with objects originating from different applications. We believe the solution lies not in coming up with application-specific approaches for tackling each issue but instead in adopting a *unified* notion of collection across applications. In this paper, we begin with a detailed summary of the key ideas behind our notion of collection and describe our implementation of these ideas in a system called Haystack. Next, we present several motivating examples of how these ideas support daily information management. Finally, we give some preliminary results from initial user studies and describe future user study work that we are looking to perform.

## APPROACH

The concept of collection we explore in this paper embodies four fundamental principles. First, collections are first class objects themselves, i.e., an unlimited number of collections can be created, and a collection is itself an object that can be filed in other collections. Second, collections are heterogeneous, meaning that objects of different types and from different applications or systems can coexist in the same collection. Third, objects can be classified into multiple collections at once. Fourth, collections can be inspected and interacted with through a variety of different presentation styles called views.

We have implemented support for collections in Haystack, an information management platform being developed at the MIT Computer Science and Artificial Intelligence Laboratory [5, 6]. Haystack is built upon a fully general semantic network data model, and metadata for the objects managed by the system, such as e-mail messages, music files, photographs, and collections, are represented in terms of a single labeled directed graph. Our use of a unified data model enables data from different applications to be integrated together. Furthermore, instances of our single, unified collection class support the principles discussed above as a natural result of the flexibility of the data model; membership is recorded as a labeled arc connecting a collection object to a member object.

The Haystack system is Open Source and downloadable from our website at http://haystack.lcs.mit.edu/. While still somewhat rough from a usability perspective, the current prototype illustrates the base functionality upon which we intend to improve the user experience and research the long term effects of unified collection management.

To illustrate the benefits of supporting a unified notion of collection, we describe three example usage scenarios below.

### Cross-application aggregation

One natural capability in Haystack is grouping objects together based on some common purpose or task, rather than by common format, type, or application. However, rather than being limited to only being able to aggregate specific types of objects without coercion, such as e-mail messages and instant messages, we promote a more general notion of aggregation that allows arbitrary types of objects to be included without a loss of functionality, as is done in systems such as LifeStreams [7] and Presto [2]. This is possible within Haystack because developers can specify the components of applications—presentation elements and user interface commands—and Haystack is capable of assembling them in the appropriate contexts. For example, considering the example collection depicted in Figure 1, one notices that webpage bookmarks, e-mails, tasks (e.g., a to-do item), pictures, and music files have been grouped into a single collection. Haystack allows different types of objects to be displayed by different user interface components, and the context menus that appear when a user right-clicks on one of these user interface components is dynamically assembled to reflect the commands known to the system that support the kind of object that was clicked on [5].

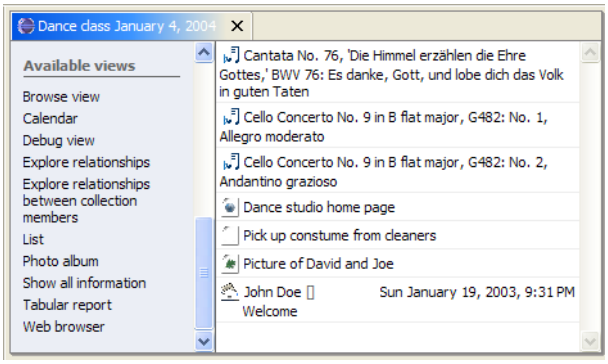


**Figure 1: Collection with items from multiple sources**

Constructing such a collection can be achieved in a number of ways. A user can click Create a collection, and as the user browses through different objects in the system, he or she can simply drag and drop objects of interest into a view of that collection. Alternatively, one can select the File away command (cf. the File Save As command in desktop applications today) for an object and see a relevant set of collections with a checkbox next to each, enabling the current selection to be

easily filed into any number of collections at once (see Figure 2). (At the moment, this set of collections is analogous to the My Documents folder in Windows; support for determining an appropriate set of collections based on context is being investigated.) While user studies will be needed to confirm the efficacy of any specific approach, what is important for this discussion is that they support the four principles outlined earlier.

Furthermore, because collections are easy to create, and an object's membership in one collection does not affect its membership in others, it is simple to build lists of related objects as one explores an information space. Shneiderman points out in his paper that the ability to assemble collections of objects enables certain creative processes [11].

### Navigating corpora

Haystack's interface is based on the Web browser paradigm, complete with back, forward, and refresh buttons and embedded hyperlinks on user interface presentations. A collection that acts as an index for a corpus such as our Starting Points collection (cf. the Start menu in Windows) can be navigated just as menus and taxonomy-driven Web pages (e.g., Yahoo! http://www.yahoo.com/) are.

However, unlike a Web browser, in which the presentation style of an object has been predetermined by the webpage designer, Haystack allows new views to be associated with objects at runtime, and a user can choose to display an object in any view that he or she deems appropriate to his or her current working context. For example, navigating a collection of digital photos is most easily accomplished with a thumbnail view of the collection, whereas browsing a list of project files would be facilitated by a multi-column sorted report view. Similarly, portals in Haystack are simply collections displayed in a view that shows applet-sized presentations of its member objects.

Decoupling an object's data model from any specific presentation style helps eliminate the need for habitat formation as users can choose to represent objects based on the suitability of a specific data model (e.g., task management) rather than on the availability of a specific presentation style (e.g., an e-mail inbox).

Furthermore, we believe general techniques for improving collection browsing such as faceted metadata retrieval, which have been shown to be successful but tested only for specific contexts (e.g., photo browsing [8]) because of the overhead involved in setting up such an experiment, can be applied in general to all kinds of collections when integrated into a system such as Haystack; we are currently investigating such support.

### Capturing metadata

One interesting opportunity arises as a result of our semantic network data model. Haystack allows users to record arbitrary metadata for objects, and we allow users to input such metadata by filling in fields on dynamically-generated forms [15]. However, adding custom properties fields may not always be intuitive to lay users, as schema customization can be an abstract concept to some. On the other hand, collection creation is relatively simple: users manage folders on their machines all the time. A duality exists between collection membership and object properties: specifying the author field of a novel as having the value "John Doe" connotes the same information as placing the novel in the "things written by John Doe" collection. One way to look at this is to say that custom metadata creation can sometimes be reduced to an act of using our analog of the File Save As dialog box. The checkbox mechanism we described earlier is one possible candidate for implementing this idea (e.g., specifying the author of a novel with respect to an extensible list of known authors).

## USER STUDIES

We have begun to test the usefulness of the principles underlying our unified notion of collection with various user studies. An earlier paper examined the notion of multiple categorization—the idea that objects can belong to multiple collections at once [10]; we briefly summarize the results of this study here.

The study compared users' preferences and performance between a system based on multiple categorization and the Microsoft Internet Explorer Favorites manager (representative of the hierarchical folder approach). Twenty-one MIT computer science students (15 male, 6 female) participated. In the first session of the study, users organized two separate corpora of 60 ZDNet.com news articles in two phases, each phase using a different approach. We chose this number of articles both to motivate users to organize the articles (smaller corpora may have been manageable with flat lists) and to prevent users from becoming overly bored or frustrated with a larger number of articles. Users then navigated those two organizational schemes in the second session of the study, again in two phases, one phase per approach, and after a one week time lag, in order to answer questions about several topics brought forth in the corpus.

On a one-tailed t-test basis, $t(19) = 1.1$, $p = 0.14$, users took considerably less time (19% reduction) organizing their corpus using multiple categorization (mean 2778.2, $\sigma$ 833.7 seconds) compared to folders (mean 3441.2, $\sigma$ 1693.9 seconds). (We considered the first phase only in these results because we observed that users took 30% less time in the second phase overall, evidence of a learning

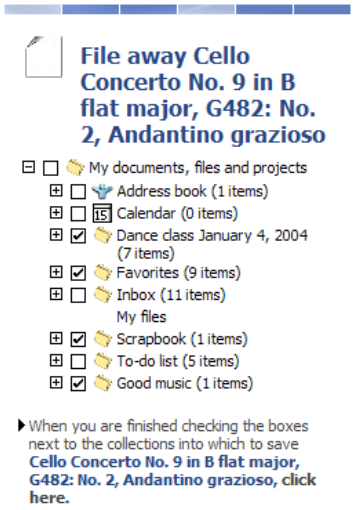


**Figure 2: File away dialog box**

effect; when considering both phases, users took 2754, σ 1434 seconds for multiple categorization versus 2586, σ 1083 seconds for folders, a less statistically significant result.) This result bolsters the claim that multiple categorization is more useful than strictly hierarchical categorization. Furthermore, our study shows that users in general put more organizational information into the system using multiple categorization than folders. On average across both corpora, we found during our study that users created 22 folders and about twice as many categories (45). In other words, users felt less inhibited to creating collections with a multiple categorization system.

We also conducted a preliminary, qualitative user study on general information management using Haystack with four participants, all experienced with computers, over a four week period [9]. We encouraged them to use the system for as much of their personal information as they felt comfortable; in particular we encouraged the use of Haystack for managing e-mails, photographs, flight itineraries, bookmark collections, and text notes. In addition, we requested that users utilize the to-do list system so that we could study the effects of merging task list management (i.e., jotting down to-do items) with organizing information with respect to these tasks (similar to what was done in the study performed by Bellotti et al.). A combination multiple choice and comments survey was administered at the end. Most multiple choice questions asked users to rate the usefulness of features on a 5 point Likert scale from "not useful at all" (1) to "invaluable" (5).

A short summary of the results relevant to this paper appears below; for complete details, refer to previous work [9]. Overall, users responded overwhelmingly positively to our support for collections. Some aspect of our unified collection support represented the favorite feature of every one of our participants. Support for items being in multiple collections at once was one feature whose usefulness was surveyed. Three users rated this support with at least a 4, and the fourth user rated it a 3 because he felt he did not have enough data to gauge the usefulness of this support.

Users especially appreciated the ability to place items of different types into the same collection. Two users rated this feature a 4, and the other two rated it a 5. Users also reported in the freeform text responses that this feature was used in the course of managing actual projects, lending weight to the hypothesis that multiple types of documents are used in the course of a project.

On the issue of switching views, our users reported that this feature was not used much. Two users did not try the feature at all; one of them reported that he preferred viewing collections as lists. One user found the performance of the system inhibiting his use of this feature.

## FUTURE WORK

We feel the results from both of the user studies discussed above and ones performed by others [1] serve as supporting evidence for our views on the importance of unified collection management. Ultimately, thoroughly testing the utility of all four principles together will require a long range user study involving users working with a system that supports a notion of unified collections. Our intention is to finish refining the Haystack system and stabilize its implementation to the point where it can sustain such a user study.

## ACKNOWLEDGMENTS

This research was supported by MIT Project Oxygen and the MIT-NTT collaboration.

## REFERENCES

1. Bellotti, V., Ducheneaut, N., Howard, M., and Smith, I. Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool. *Proc. CHI 2003*.
2. Dourish, P., Edwards, W.K., et al. Extending Document Management Systems with User-Specific Active Properties. *ACM Transactions on Information Systems 18 (2)*, 140–170.
3. Abrams, D., Baecker, R., and Chignell, M. Information Archiving with Bookmarks: Personal Web Space Construction and Organization. *Proc. CHI 1998*.
4. Whittaker, S. and Sidner, C. Email Overload: Exploring Personal Information Management of Email. *Proc. CHI 1996*.
5. Quan, D., Huynh, D., and Karger, D. Haystack: A Platform for Authoring End User Semantic Web Applications. *Proc. Int'l Semantic Web Conf. 2003*.
6. Quan, D. and Karger, D. Haystack: A User Interface for Creating, Browsing, and Organizing Arbitrary Semistructured Information. *Proc CHI 2004*.
7. Freeman, E. and Gelernter, D. Lifestreams: A Storage Model for Personal Data. *SIGMOD Record 25 (1)*.
8. Yee, K., Swearingen, K., Li, K., and Hearst, M. Faceted Metadata for Image Search and Browsing. *Proc. CHI 2003*.
9. Quan, D. Designing End User Information Environments Built on Semistructured Data Models. Doctoral Dissertation.
10. Quan, D., Bakshi, K, Huynh, D., and Karger, D. User Interfaces for Supporting Multiple Categorization. *Proc. INTERACT 2003.*
11. Shneiderman, B. Creating Creativity: User Interfaces for Supporting Innovation. *ACM Transactions on Computer-Human Interaction 7 (1)*, 114–138.
12. Rohall, S. and Gruen, D. ReMail: A Reinvented Email Prototype. *Proc. CSCW 2002.*
13. Ducheneaut, N. and Bellotti, V. Email as Habitat: An Exploration of Embedded Personal Information Management. *Interactions 8 (5)*, 30–38.
14. Lansdale, M. The Psychology of Personal Information Management. *Applied Ergonomics 19 (1)*, 1988, pp. 55–66.
15. Quan, D., Karger, D., and Huynh, D. RDF Authoring Environments for End Users. *Proc. Semantic Web Foundations and Application Technologies 2003*.