Haystack: Metadata-Enabled Information Management

Dennis Quan, David R. Karger
MIT AI Laboratory/LCS, 200 Technology Square, Cambridge, MA 02139 USA
E-mail: {dquan,karger}@ai.mit.edu

KEYWORDS: metadata, direct manipulation, collections, tasks, views, personal information management

INTRODUCTION

Computers force users to organize information using rigid tools such as folder hierarchies—the modern analog of the medieval filing cabinet. Many researchers have begun to examine this problem in earnest by allowing users to associate information with objects such as timelines, tasks, and keywords. Haystack generalizes these approaches by permitting freeform associations between arbitrary objects, which better matches the organizational styles of some users while also supporting those who require a more structured framework for organization [1]. By using a semantic network-based data model, Haystack can integrate information currently dispersed amongst multiple separate applications, such as e-mail, contact lists, calendars, documents, bookmark collections, and notes, and presents a unified environment that allows users to browse and aggregate items of importance in whatever way is most natural to them. Our demonstration touches upon several key elements of our system—views, collections, and tasks—which enable users to take advantage of Haystack's flexibility.

VIEWS

Haystack's user interface, depicted in Figure 1, is responsible for presenting information regarding objects of interest to the user on the screen. When an object, such as an e-mail message, needs to be displayed on screen, Haystack renders a *view* of the object. Views are user interface elements that serve as on-screen proxies for objects in the underlying data model, an idea explored in previous work [2] [3]. Several kinds of views may be associated with any one object; for example, an e-mail may be displayed as a short summary (a "summary" view), with an abbreviated list of headers and the body of the message (an "applet" view), or with a full list of headers, a body, and an area for managing attachments (a "full screen" view). New views may be introduced into the system, allowing freedom and flexibility in terms of how an object may be presented in a manner most suitable to the current context. Views can also be created by composing child views; for example, a view of an address book would incorporate views of the people listed within.

Views enable users to specify metadata to the system in two ways. First, full screen, applet, and other larger-sized views tend to host widgets that allow the properties of the object being represented to be manipulated. A view of a flight reservation object might consist of a form with fields for specifying the airline, the origin, and the destination. Second, all views serve as representatives of objects in the data model, such that when a user drags one view onto another view, he or she is telling the system that a command is to be performed involving the objects represented by the two views. For example, if a user drags an icon view of a person into the CC field of an e-mail message view, he or she is indicating that the person is to be CCed on the message. The nature of how such commands are specified and how ambiguity is resolved can be found in [4].

COLLECTIONS

A semantic network, such as the kind used by Haystack, is capable of representing a variety of different kinds of information [5]. However, one particular type of object occurs commonly enough to warrant special discussion—the collection. Collections are used to group related objects together. Although not often thought of as being the same class of object, items ranging from photo albums and bookmark hierarchies to to-do lists and e-mail inboxes are all collections. When considering that users more naturally group objects by purpose rather than by type (that is not to say that these do not sometimes coincide) [6], the real distinction between these different forms of collection is the style of presentation. In Haystack, all collections are treated the same, collections can contain multiple types of objects at once, and views allow users to control the style of presentation. Furthermore, collection views embed views of the items contained within the collection, allowing users to use drag and drop to control which items are associated with particular collections.

Existing implementations of collections have conflated the two distinct roles collections play in many systems: storage and organization [7]. While restricting files to being stored in one directory at a time may be logical for the purposes of storage efficiency, such a restriction inhibits the use of directories (as well as bookmark folders, e-mail folders, and other strict hierarchies) as effective instruments for organi-

zation. In Haystack, objects can be associated with multiple collections at once.

TASKS

Many personal information managers support basic task management: users can create to-do items and folders for filing away task-specific documents. Haystack supports similar functionality (albeit with more flexibility), but more importantly, Haystack uses knowledge of what task is being performed by the user to determine relevant commands and information. In applications today, a finite number of commands and views exist for application-specific objects specialized to the tasks for which an application was designed. For example, an e-mail application may only support "send message", "add to address book" and "set up appointment" commands for a person object. In contrast, in a system such as Haystack where there is no specific task for which the system is designed, the number of commands may be overwhelming. Instead of "switching" applications to indicate the current context, users can specify the tasks that are active. A pane appears on the left hand side of the screen for each active task. This is done in analogy to people's desks, where a task in progress may have an associated folder or paperwork placed on the desk. Task-specific commands and information are then shown on screen.

ACKNOWLEDGMENTS

This work was supported by the MIT-NTT collaboration, the MIT Oxygen project, and IBM.

REFERENCES

1. Huynh, D., Karger, D., and Quan, D. (2002). "Haystack: A Platform for Creating, Organizing and Visualizing Information Using RDF." Semantic Web Workshop, The Eleventh World Wide Web Conference 2002 (WWW2002).
2. Quan, D., Karger, D., and Huynh, D. (2003). RDF Authoring Environments for End Users. Proceedings of Semantic Web Foundations and Application Technologies 2003.
3. McKay, S., York, W., and McMahon, M. A Presentation Manager Based on Application Semantics. In Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology.
4. Quan, D. (2003). Designing End User Information Environments Built on Semistructured Data Models. Doctoral Dissertation.
5. Brachman, R. J. (1977). "What's in a Concept: Structural Foundations for Semantic Networks." International Journal of Man-Machine Studies 9: 127-152.
6. Bellotti, V., Ducheneaut, N., Howard, M., and Smith, I. Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool. Proceedings of CHI 2003.
7. Dourish, P., Edwards, W.K., et al. "Extending Document Management Systems with User-Specific Active Properties." ACM Transactions on Information Systems, vol. 18, no. 2, April 2000, pages 140–170.

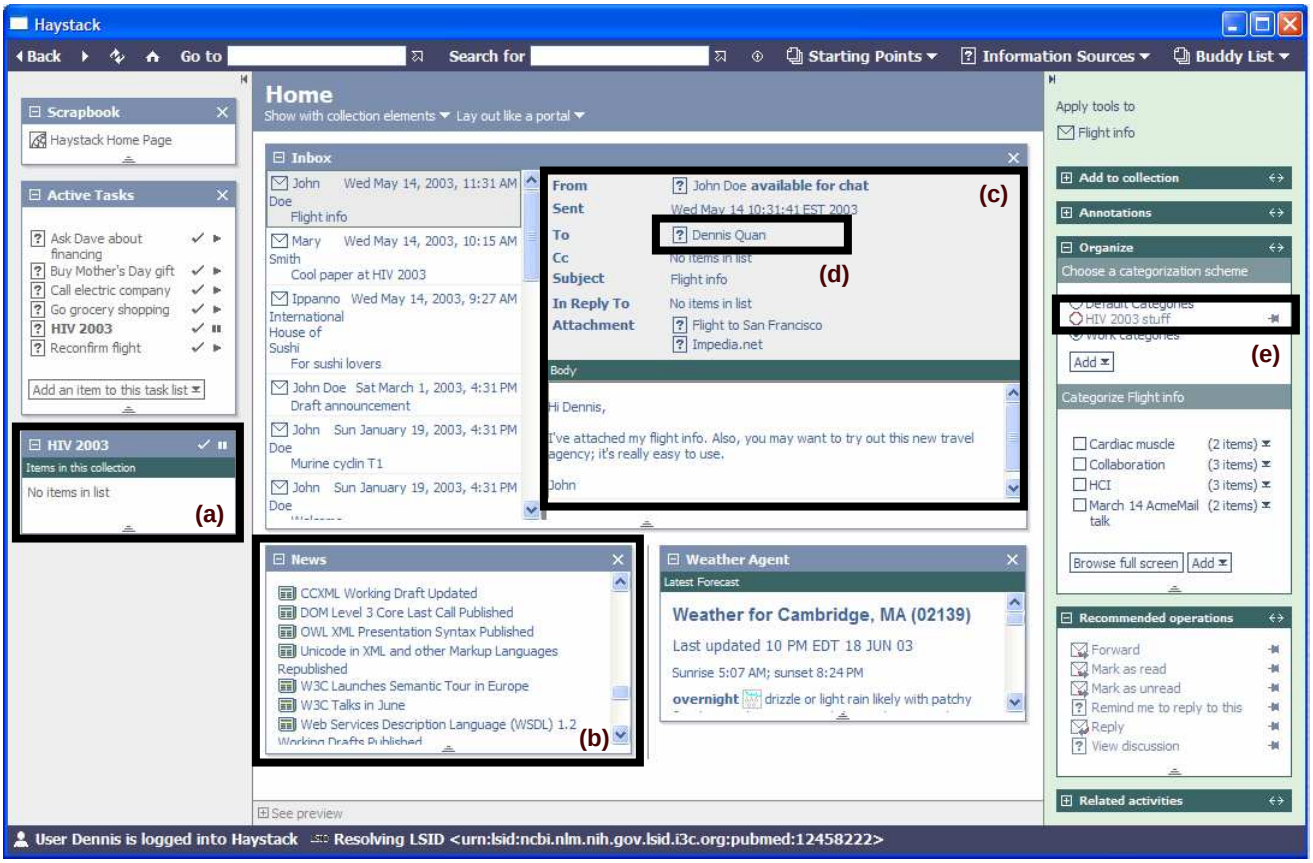

Figure 1: Haystack user interface: (a) the HIV 2003 project task's pane; (b) a collection view; (c) a form view; (d) an embedded view; (e) an option appearing as a result of the active HIV 2003 project task