Haystack: Per-User Information Environments
MIT9904-08

Progress Report: January 1, 2001—June 30, 2001

David Karger and Lynn Andrea Stein

Project Overview

The Haystack project is investigating the ways in which electronic infrastructure can be used to triangulate among the different knowledges of an individual, of collegial communities to which we belong, and of the world at large. Its infrastructure consists of a community of independent, interacting information repositories, each customized to its individual user. It provides automated data gathering (through active observation of user activity), customized information collection, and adaptation to individual query needs. It also facilities inter-haystack collaboration, exposing (public subsets of) the tremendous wealth of individual knowledge that each of us currently has locked up in our personal information spaces. The Haystack-NTT project involves augmenting its customization, learning and adaptation, and inter-haystack communication.

Progress Through June 2001

Our recent work has focussed on implementation of a complete redesign of the Haystack system. With the help of our new team of PhD students, we have nearly completed our implementation of the design I discussed in the past report. At the core of this design is a general purpose "RDF triple store" that is used to record semistructured information in the Resource Description Framework being popularized by the World Wide Web consortium. Features of this triple store include:

* The store is built on top of a relational database system with transaction (using Java's JDBC framework) to provide robust and efficient storage of information.

* The store provides APIs based on the SOAP (Standard Object Access Protocol) interface over XML. This means arbitrary applications can store information in the triple store, making it easy to expose information for use by Haystack, or to make use of information provided by Haystack.

* The store is optimized to maintain "attribution" information about the metadata in the system. The fact that Haystack is designed to support collaboration means that many different entities will be recording assertions into the triple store. It is important to know which said what in order to let an entity make its own decisions about which assertions it will believe. To support this, we have designed a "belief server" layer that lets a given entity

specify its preferences about who to believe and transparently provides an entity with a consistent view of the repository containing only that information the entity wishes to believe.

* All the old data extraction services written for the previous incarnation of Haystack have been revised to work with the new storage layer.

At the machine learning layer, we have developed a prototype system called Winnow that explores ways the system can learn over time to provide its owner with "better" information. Winnow spiders a large collection of news oriented sites and disassembles those sites into individual news stories. It then chooses some of them to display as interesting to the user. Based on which stories the user reads, the system learns what kind of stories the user likes, and over time learns to show the user stories of greatest interest to them.

This type of "filtering" application has been studied before. Somewhat novel is our choice of features to be used in deciding what is interesting. Rather than working with the text of the articles themselves, Winnow focuses on the text contained in the URLs pointing at the articles. It turns out that many sites choose their URLs to reflect some hierarchical structuring of their information by topic---for example, the new york times has URLs that begin nytimes.com/business, nytimes.com/sports, etc., and are then extended to more refined subcategories. According to preliminary experiments, this topic hierarchy is much more effective than text as an indicator of what a user will be interested in (it is also extremely good at filtering advertisements out of what is shown to the user).

At the user interface level, we have begun to explore the way user interfaces connect to our semistructures data model. RDF is not just a place to store information; it is also a good place to store information about how to display information. We are building an "object oriented user interface template scheme" that lets each object specify how it is supposed to be displayed, according to a general purpose template. For example, we might want to specify, in the absence of a title for a given object, what other attributes can be used to name that object. This approach raises some interesting question about the balance of power between the data model (which contains specifications for how things should be displayed) and the interfaces (which actually carry out the display).

Plans for the Next Six Months

Over the next few months we hope to carry out a substantial amount of user testing. We are in the midst of experimental design for a user study that will hopefully indicate that Winnow is much better than other approaches at finding articles of interest to a reader. Upon the completion of our haystack reimplementation, we hope to release it to a small group of alpha users and begin gathering information on how that use the system and what needs to be improved to make it truly useful. Finally, we have just begun some extensive data gathering (based on videotaping and interviewing) to learn about the ways people use their current tools (directory structures, windows on the screen, bookmark files) to organize information; we hope to discover some general principles that can be applied to the design of better information management tools within the haystack framework.