Haystack: Per-User Information Environments

David Karger
Motivation
Individualized Information Retrieval

• One size does NOT fit all
  – Library is to bookshelf as google is to ….

• Best IR tools must adapt to their individual users
  – Hold content that is appropriate to that user
  – Organize it to help that user navigate and organize it
  – Adapt over time to how that user wants things done
  – Like a bookshelf, or a personal secretary
Haystack Approach

• **Data Model**
  – Define a rich data model that lets user represent all interesting info
  – Rich search capabilities
  – Machine readable so that agents can augment/share/exchange info

• **User Interface**
  – Strengthen UI tools to show rich data model to user
  – And let them navigate/manipulate it

• **Adaptability**
  – People are lazy, unwilling to “waste time” telling system what to do, even if it could help them later
  – System must introspect about user actions, deduce user needs and preferences, and self-adjust to provide better behavior
Data Model

A semantic web of information
The Haystack Data Model

- W3C RDF/DAML standard
- Arbitrary objects, connected by named links
  - A semantic web
  - Links can be linked
- No fixed schema
  - User extensible
  - Add annotations
  - Create brand new attributes
Agent Environment

- **Various types rooted in RDF containers**
  - Extract structured data from traditional formats
  - Extend RDF through analysis/integration of other RDF
  - Take actions (notify user gui, fetch web info, send email)

- **Various Triggers**
  - Scheduled actions
  - Actions triggered by arrival/creation of new RDF patterns

- **Belief Server**
  - Agents will disagree
  - User specifies which are more trustworthy
  - Belief server filters each disagreement

- **User is ultimate arbiter (via user interface)**
Database Needs

• **Power**
  – Support general purpose SQL-style queries over arbitrary RDF

• **Speed**
  – Haystack stores all state in data model
  – So issues huge number of tiny, trivial queries to model
  – Traditional databases assume real work of query will dominate initialization/marshalling costs
  – So traditional databases don’t work for haystack

• **Wanted: all-in-one data repository**
Gathering Data

- **Active user input**
  - Interfaces let user add data, note relationships

- **Mining data from prior data**
  - Plug-in services opportunistically extract data

- **Passive observation of user**
  - Plug-ins to other interfaces record user actions

- **Other Users**
User Interface

Uniform Access to All Information
Current Barriers to Information Flow

- **Partitions by Location**
  - Some data on this computer, some on that
  - Remote access always noticeable, distracting

- **Partitions by Application**
  - Mail reader for this, web browser for that, text editor for those
  - Todo list, but without needed elements

- **Invisibility**
  - Where did I put that file?
  - Tendency for objects to have single (inappropriate) location (folder)

- **Missing attributes**
  - Too lazy to add keywords that would aid searching later
Goal: Task-Based Interface

- When working on X, all information relevant to X (and no other) should be at my fingertips
  - Planning the day: todo list, news articles, urgent email, seminars
  - Editing a paper: relevant citations, email from coauthors, prior versions
  - Hacking: code modules, documentation, working notes, email threads

- Location, source and format of data irrelevant
Sign of Need: Email Usage

- **Email as todo list**
  - Anything not yet “done” kept there
  - Reminder email to ourselves
  - Single interface containing numerous document types

- **Overflowing Inboxes**
  - Navigate only by brute-force scanning
  - Unsafe file/categorize anything: out of sight, out of mind
Options

• Folders
  – Out of sight, out of mind
  – Still need applications to see data
  – Which is the right folder?

• Desktops
  – Allow arbitrary data types
  – But coupling between applications & data types too light
  – A smear of many tasks, so hard to focus
    * Hundreds of icons, tens of windows, huge menus
    * No partitioning

• RDF (our choice)
  – Treat information uniformly
  – Let each information object present itself in context
The Big Picture

Ozone Personal Edition

Welcome Home

Items Needing Attention
Send Message  Go to Full View

Your Calendar
Record Appointment  Set Up Meeting

May 24, 2002  May 25, 2002  May 26, 2002
12:00 am
12:30 am
1:00 am
1:30 am
2:00 am
2:30 am
3:00 am
3:30 am
4:00 am
4:30 am
5:00 am
5:30 am

Search Your Haystack
What are you looking for?
[Search]

Weather
Weather for Cambridge, MA (02139)
Last updated 10 AM EDT 24 MAY 02
Sunrise 5:15 AM; sunset 8:07 PM

today mostly sunny with a chance of showers and thunderstorms late in the afternoon. Highs 73 to 83. West winds 15 to 20 mph, chance of rain 30 percent.

tonight partly cloudy. Lows 45 to 50. Northwest winds 10 to 15 mph.
saturday mostly sunny and cooler. Highs 51 to 66. North winds

Your News

REUTERS

Health Information

Hormone replacement linked to gallstone risk
User Interface Architecture

- Views: Data about how to display data
- Views are persistent, manipulable data

![Diagram showing the User Interface Architecture with views and mappings between data and UI data.]

David Karger — MIT Laboratory for Computer Science and Artificial Intelligence Laboratory
Semantic User Interface

• Present information by assembling different views together

• Information manipulation decoupled from presentation
  – Lower barrier of entry for development
  – New data types can be added without designing new UIs

• Uniform support for features like context menus
  – Actions apply to objects on screen in various “roles”
  – E.g. as word, as name of mail message, as member of collection
Tasks Become Modeless Data
Persistence of Views

• Views are data like all other data
• Stored persistently, manipulated by user
• User can customize a view
  – View for particular task can be cloned from another
  – Can evolve over time to need of task
  – To an extent previously limited to sophisticated UI designer
• Views can be shared (future work)
  – Once someone determines “right” way to look at data, others can benefit
Adaptation

Learning from the User over Time
(Future Work)
Approach

• **Haystack is ideally positioned to adapt to user**
  – RDF data model provides rich attribute set for learning
  – In particular, can record user actions with information
    * (which flexible UI can capture)
  – Extensive record can be built up over time

• **Introspect on that information**
  – Make Haystack adapt to needs, skills, and preferences of that user
Observe User

- **Instrument all interfaces, report user actions to haystack**
  - Mail sent, files edited, web pages browsed
- **Discover quality**
  - What does the user visit often?
- **Discover semantic relationships**
  - What gets used at the same time?
- **Discover search intent**
  - Which results were actually used?
Learning from Queries

- **Searching involves a dialogue**
  - First query doesn’t work
  - So look at the results, change the query
  - Iterate till home in on desired results

- **Haystack remembers the dialogue**
  - instead of first query attempt, use last one
  - record items user picked as good matches
  - on future, similar searches, have better query plus examples to compare to candidate results
  - Use data to modify queries to big search engines, filter results coming back
Mediation

- **Haystack can be a lens for viewing data from the rest of the world**
  - Stored content shows what user knows/likes
  - Selectively spider “good” sites
  - Filter results coming back
    * Compare to objects user has liked in the past
  - Can learn over time

- **Example - personalized news service**
News Service

• Scavenges articles from your favorite news sources
  – Html parsing/extracting services
• Over time, learns types of articles that interest you
  – Prioritizes those for display
• Uses attributes other than article content
  – Current system based entirely on URL of story
Personalized News Service

Welcome karger logout

Your main page new page edit delete

Your Pages: main research

www.firstmonday.org/issues/cur

- Space traffic control Reading for travel crunch cnn.com
- Shuttle schedule in limbo due to Alpha robot arm cnn.com
- Report: Mars could have icy equator cnn.com
- Deep Space Network cnn.com
- Violence tests Mideast cease-fire cnn.com
- Hundreds injured in Algerian riots cnn.com
- Violent clashes at EU summit cnn.com
- 'Abortion ship' arrives in Ireland cnn.com
- EU leaders not converted cnn.com
- Bush to Russia: U.S. 'no longer your enemy' cnn.com

---
cnn.com configure refresh updated 19 minute(s) ago

June 15, 2001 -- Updated 10:36 a.m. EDT, 1436 GMT

President Bush with Polish President Aleksander Kwasniewski in Warsaw

Bush to Russia: U.S. 'no longer your enemy'

President Bush says he is looking forward to meeting Russian President Vladimir Putin on Saturday because it will be an opportunity to tell the Russian leader that "the United States is no longer your enemy."
Underway Projects

- Mail Auto-classifier
- Generalized querying/relevance feedback based on Haystack’s rich attribute set
Collaboration

Haystack’s Ulterior Motive
Hidden Knowledge

- People know a lot that they are
  - Willing to share
  - But too lazy to publish
- Haystack passively collects that knowledge
  - Without interfering with user
- Once there, share it!
  - RDF---uniform language for data exchange
- Challenges
  - As people individualize systems, semantics diverge
  - Who is the “expert” on a topic? (collaborative filtering)
Example

- **Info on probabilistic models in data mining**
  - My haystack doesn’t know, but “probability” is in lots of email I got from Tommi Jaakola
  - Tommi told his haystack that “Bayesian” refers to “probability models”
  - Tommi has read several papers on Bayesian methods in data mining
  - Some are by Daphne Koller
  - I read/liked other work by Koller
  - My Haystack queries “Daphne Koller Bayes” on Yahoo
  - Tommi’s haystack can rank the results for me…