Enterprise and Desktop Search

Lecture 5: Desktop Search and Personal Information Management

Pavel Dmitriev
Yahoo! Labs
Sunnyvale, CA
USA

Pavel Serdyukov
Delft University of Technology
Netherlands

Sergey Chernov
L3S Research Center
Hannover
Germany
Searching Personal Collections with Memex

Posited by Vannevar Bush in “As We May Think”
The Atlantic Monthly, July 1945

“A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility”

Supports: Annotations, links between documents, and “trails” through the documents

“yet if the user inserted 5000 pages of material a day it would take him hundreds of years to fill the repository, so that he can be profligate and enter material freely”
Desktop Search and Personal Information Management

• **Desktop search** is the name for the field of search tools which search the contents of a user's own computer files, rather than searching the Internet. These tools are designed to find information on the user's PC, including web browser histories, e-mail archives, text documents, sound files, images and video.

• **Desktop Search** is a part of a more general field of **Personal Information Management (PIM)**.

• **Personal Information Management (PIM)** refers to both the practice and the study of the activities people perform in order to acquire, organize, maintain, retrieve and use information items such as documents (paper-based and digital), web pages and email messages for everyday use to complete tasks (work-related or not) and fulfill a person's various roles (as parent, employee, friend, member of community, etc.)

Desktop Search: Motivation

- Why *desktop search*?
  - Size of data on the desktop is big (50k – 500k items) and continuously growing
  - Moving towards Social Semantic Desktop
  - Social – communication in a social network
  - Semantic – metadata descriptions and relations
What is Desktop?

• Documents (doc, pdf, ppt, xls, html, txt, …)

• Email

• Calendar

• Instant Messengers (ICQ, Skype, MSN messenger, …)

• Pictures

• Music

• Videos
Desktop Search – Current Status

- Documents on the desktop are not linked to each other in a way comparable to the web
- Simple full text search
  - no personalization
  - no context
  - no ranking possible or too poor
- Metadata enriched search makes use of
  - associations to contexts and activities
  - provenience of information
  - sophisticated classification hierarchies
Differences between Web Search and Desktop Search

• Search on the desktop vs. Search on the Web
  – Re-finding vs. finding
  – Integration across many applications and file formats
  – Users prefer to navigate, not to search
  – Many information types: ephemeral, working, archived
  – Extra sources for ranking improvement:
    • File metadata
    • Usage metadata
    • Folder structure
  – Privacy concerns
Outline

• Today we will talk about:
  – Modern Desktop Search Engines
  – Research prototypes
  – Just-In-Time Retrieval
  – Context on a Desktop
    • Using context to improve Desktop Search
    • Context Detection
  – PIM Evaluation
Modern Desktop Search Engines

- Google Desktop (from major web search engine vendor)
- Windows Search (from major OS provider)
- Copernicus (company specialized on DS engines)
- Beagle (open source DS for Linux)
- Yandex (Russian DS)

Some more:

Desktop Search Architecture

# Desktop Search Engines in 2005

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop Search Tool</td>
<td>Version</td>
<td>Score (Min = 1.00, Max = 5.00)</td>
<td>Better</td>
<td>Desktop Search Tool</td>
<td>Version</td>
</tr>
<tr>
<td>Copernic</td>
<td>1.0 Beta</td>
<td>4.44</td>
<td></td>
<td>Copernic</td>
<td>1.0 Beta</td>
</tr>
<tr>
<td>Archivarius</td>
<td>1.1 Beta</td>
<td>4.76</td>
<td></td>
<td>Yahoo!</td>
<td>4.68</td>
</tr>
<tr>
<td>Google</td>
<td>3.0 Beta</td>
<td>4.40</td>
<td></td>
<td>Blinkx</td>
<td>4.76</td>
</tr>
<tr>
<td>MSN</td>
<td>3.14</td>
<td>4.40</td>
<td></td>
<td>ISYS</td>
<td>3.76</td>
</tr>
<tr>
<td>Ask Jeeves</td>
<td>3.45</td>
<td>4.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copenic</td>
<td>1.0 Beta</td>
<td>4.06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Archivarius</td>
<td>3.0 Beta</td>
<td>4.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSN</td>
<td>3.14</td>
<td>4.20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dtSearch</td>
<td>3.59</td>
<td>3.90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yahoo!</td>
<td>3.29</td>
<td>3.20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask Jeeves</td>
<td>3.14</td>
<td>3.14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google</td>
<td>3.13</td>
<td>4.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Copernic with Covers, and Yahoo with X1

Source: UWE-Business Consortium

# Sample Criteria for DS Comparison

<table>
<thead>
<tr>
<th>Search Format</th>
<th>Platform(s)</th>
<th>Feature</th>
<th>Opt-in Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain text</td>
<td>Windows Vista</td>
<td>Specifying index location</td>
<td>Default search engine</td>
</tr>
<tr>
<td>HTML pages stored locally</td>
<td>Windows XP</td>
<td>Incremental indexing</td>
<td>Web integration</td>
</tr>
<tr>
<td>Microsoft Word (.doc)</td>
<td>Mac OS X</td>
<td>Legacy index by scanning</td>
<td>Insecure search</td>
</tr>
<tr>
<td>Microsoft Excel (.xls)</td>
<td>Linux</td>
<td>Engine download size</td>
<td>Registration</td>
</tr>
<tr>
<td>Microsoft PowerPoint (.ppt)</td>
<td>Mozilla/Firefox</td>
<td>Install size</td>
<td>Engineering feedback</td>
</tr>
<tr>
<td>Rich Text Format (.rtf)</td>
<td>Internet Explorer</td>
<td>Combined local/remote search</td>
<td>Software updates</td>
</tr>
<tr>
<td>Portable Document Format (.pdf)</td>
<td>Opera</td>
<td>Non-anonymous connections</td>
<td></td>
</tr>
<tr>
<td>Microsoft Outlook email</td>
<td>Safari</td>
<td>Excluding files</td>
<td></td>
</tr>
<tr>
<td>Microsoft Outlook Express email</td>
<td>Languages</td>
<td>Indexing progress indicator</td>
<td></td>
</tr>
<tr>
<td>Microsoft address books</td>
<td></td>
<td>Recoverable index</td>
<td></td>
</tr>
<tr>
<td>AOL Instant Messenger</td>
<td></td>
<td>File type filtering</td>
<td></td>
</tr>
<tr>
<td>Standard email folder support</td>
<td></td>
<td>Deskbar</td>
<td></td>
</tr>
<tr>
<td>Standard news folder support</td>
<td></td>
<td>Support for compressed files</td>
<td></td>
</tr>
<tr>
<td>Browser web history</td>
<td></td>
<td>Support for legacy file formats</td>
<td></td>
</tr>
<tr>
<td>Browser secure web history</td>
<td></td>
<td>Ignoring networked drives</td>
<td></td>
</tr>
<tr>
<td>Browser bookmarks</td>
<td></td>
<td>Click to suspend</td>
<td></td>
</tr>
<tr>
<td>Browser address books</td>
<td></td>
<td>Click to exit</td>
<td></td>
</tr>
</tbody>
</table>
Windows Desktop Search
Copernicus Desktop Search

License Agreement

This License Agreement (this “Agreement”) is made effective as of 08-01-15 (the “Effective Date”) between The Software Company Inc. (“Licensor”) and Edwards Construction Inc. (“Licensee”).

In consideration of the mutual promises contained herein and other good and valuable consideration, the receipt and sufficiency of which is hereby acknowledged, the parties agree as follows:

1. Content of Licensed Materials; Grant of License

The materials that are the subject of this Agreement shall consist of all software and manuals for The Software Company Inc. as hereinafter referred to as the “Licensed Materials.”
Beagle Desktop Search
Yandex Desktop Search

Персональный поисковик Яндекса — это программа на вашем компьютере, осуществляющая поиск файлов и писем с учетом коррекции русского языка.

Форма поиска

Управление представлением результатов
Можно выбрать способ группировки и сортировки.

Конфиденциальность
Можно запретить искать в определенных папках или целых дисках.

Доступ к Поисковому поиску из панели задач

Описания запросов и результатов поиска открываются в обычном браузере.
Research prototypes and Semantic Desktops

- Beagle++ *(extended open source DS)*
- Semex *(includes Malleable Schemas)*
- Haystack and Magnet *(Semantic Web approach)*
- Stuff I’ve Seen *(Phlat predecessor)*
- Phlat *(was used as a basis for Windows DS)*
- PIA *(semantic desktop solution from DB area)*

Some more:
  - Gnowsis, CALO
Beagle++

• Why is it so hard to find what you need on your desktop – “You still use Google even for files stored on your computer?”

• Current desktop search engines use only full text index

• People tend to associate things to certain contexts

• For desktop search we need to support contextual information in addition to full text!
  – Relationships between information items (citations)
  – Relationships based on interactions (email exchange, browsing history)
  – Relationships between different types of items (authorship, publication venues, email sender information, recommendations)
  – Other situational context


Scenario 1: The Need for Context Information

• Alice and Bob are working together in the research group

• Alice is currently writing a paper about searching and ranking on the semantic desktop and wants to find some good papers on this topic, which she remembers she stored on her desktop

• Some time ago Bob sent her a very useful paper on this topic as an attachment to an email, together with some useful comments about its relevance to her new semantic desktop ideas

• Will Alice find the paper from Bob when issuing a query on the desktop, using the search terms “semantic desktop”?
Context Information is necessary!

- **Problems:**
  - (Mail) Documents sent as attachments lose all contextual information as soon as they are stored on the PC
  - (Web) When searching for a document we downloaded from the CiteSeer repository, we would like to retrieve not only the specific document, but all the referenced and referring papers which we already downloaded as well

- Current desktop search approaches don’t make use of desktop specific information, especially contextual information, like:
  - **Email** context
  - **Web** context
  - **Publication** context
Representing Context by Semantic Web Metadata

- Metadata for resources can be created by appropriate metadata generators
- Ontologies specify context metadata for:
  - Emails
  - Files
  - Web pages
  - Publications
- Metadata have to be application-independent!

→ Store Metadata as RDF
  - generated and used by whatever application you can think of
Beagle++ Layer Architecture

Beagle++ is our extension of the open source Beagle search project, enabling it to exploit context information.

RDF metadata are generated based on ontologies for specific contexts (email, web, etc.).

Indexing and metadata generation on the fly - triggered by events upon occurrence of file system changes (*inotify-enabled linux kernel*)

Benefits:

- Context allows us to better organize and find information.
- Context gives us the possibility to compute the value / importance of resources.
Beagle++ Architecture

- Kernel
- Inotify
- Beagle Control Process
  - Indexing and Search Requests
  - Search Requests and Results
- Queryable Data Sources
  - Google Queryable
  - Email Queryable
- Filesystem Queryable
- Content Filter
  - Delegating the Data Extraction
  - Content
- Metadata Filter
  - Metadata
- Metadata Storage Module
- Lucene Full-text Index
  - Queries and Updates
  - Queries and Updates
  - Metadata
  - Metadata
- Best++ Search Interface
  - Metadata
- Metadata Generation
  - Metadata
- Metadata Enrichment
  - Metadata
  - Metadata
  - PDF Filter
    - Office Docs Filter
    - PDF Filter
    - BibTeX Filter
Beagle++: Find more than documents
Beagle++: Display additional context

<table>
<thead>
<tr>
<th>Activity Based Metadata for Semantic Desktop Search</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authors:</strong></td>
</tr>
<tr>
<td>Paul - Alexandru Chirita</td>
</tr>
<tr>
<td>Rita Gavriloaie</td>
</tr>
<tr>
<td>Stefania Ghita</td>
</tr>
<tr>
<td>1. Searching and Ranking on the Semantic Desktop</td>
</tr>
<tr>
<td>2. Semantically Rich Recommendations in Social Networks for Sharing and Exchanging Semantic Context</td>
</tr>
<tr>
<td>more ..</td>
</tr>
<tr>
<td>Wolfgang Nejdl</td>
</tr>
<tr>
<td>Raluca Paiu</td>
</tr>
<tr>
<td><strong>Conference:</strong></td>
</tr>
<tr>
<td>In Proceedings of the 2nd European Semantic Web Conference, Heraklion, Greece</td>
</tr>
<tr>
<td><strong>Year:</strong></td>
</tr>
<tr>
<td>May, 2005</td>
</tr>
<tr>
<td><strong>Cited papers:</strong></td>
</tr>
<tr>
<td>more ..</td>
</tr>
<tr>
<td><strong>Citing papers:</strong></td>
</tr>
<tr>
<td>1. W. Nejdl and R. Paiu. I know I stored it somewhere - Contextual Information and Ranking on our Desktop.</td>
</tr>
<tr>
<td>more ..</td>
</tr>
<tr>
<td><strong>Location:</strong></td>
</tr>
<tr>
<td>Email: Bob</td>
</tr>
<tr>
<td>Sent: Bob. Received: Today, 2:56 PM</td>
</tr>
</tbody>
</table>
Integrating Keyword and Metadata Search

- Search text and metadata on the desktop
- Search efficiently in a user-friendly way
- Simple query language
- No complete schema knowledge necessary
Documents / RDF Fragments

- Metadata stored as RDF graphs, each document has a corresponding RDF fragment
- Extended documents consisting of both full-text and metadata properties
- Query model supports the operator selection, projection and union, intersection and set difference

- Support for approximate and imprecise metadata queries
- Separation between metadata statements is ensured by positional indices
Scenario

• Bob, Alice and Tom exchange resources via email
• They do not only exchange documents, but also context information using the Beagle++ Thunderbird extension
• Alice trusts Bob more than Tom
Peer-Sensitive ObjectRank [1]

- Step 1: start with PageRank formula – random surfer model
  \[ r = d \cdot A \cdot r + (1 - d) \cdot e \]
  - \( d \) = dampening factor
  - \( A \) = adjacency matrix
  - \( e \) = vector for the random jump

- Step 2: distinguish between different kinds of objects
  - ObjectRank variant of PageRank
Peer-Sensitive ObjectRank [2]

\[
\begin{pmatrix}
YingMa & Kleinberg & Balmin & P & Q & J & H & B & WWW & VLDB \\
YingMa & - & - & 0.2 & 0.2 & - & 0.2 & 0.2 & - & - \\
Kleinberg & 0.25 & - & - & - & 0.73 & 0.23 & 0.23 & 0.7 & 0.1 \\
Balmin & - & 0.25 & - & - & - & - & 0.1 & - & 0.1 \\
P & - & 0.25 & - & - & 0.23 & 0.23 & - & - & - \\
Q & - & - & - & 0.23 & - & - & - & - & - \\
J & - & - & 0.73 & - & - & 0.73 & 0.1 & - & - \\
H & - & - & - & - & 0.23 & - & - & 0.1 & 0.1 \\
B & - & - & - & - & - & 0.1 & 0.1 & - & 0.1 \\
WWW & - & - & - & 0.23 & 0.1 & 0.1 & 0.1 & - & 0.4 \\
\end{pmatrix}
\]
Peer-Sensitive ObjectRank [3]

- Step 3: Take provenance information into account
- → Peer-Sensitive ObjectRank
- Represent different trust in peers by corresponding modifications in the $e$ vector
- Keep track of the provenance of each resource

originates $(r_i, P_n) = \begin{cases} 1, & \text{if } r_i \text{ is in the initial set of } P_n \\ 0, & \text{otherwise} \end{cases}$

$\text{trust}(P_i, P_j) \in [0,1]$, the trust value of peer $P_i$ for $P_j$

$e_k(P_i) = \max_{j=0}^{N} \{ \text{trust}(P_i, P_j) \cdot \text{originates}(r_k, P_j) \}$

Beagle++ Demo
**Open Source Search Engines**

Build your own search engine!

<table>
<thead>
<tr>
<th>Search Engine</th>
<th>Indexing Time (h:m:s)</th>
<th>Index Size (%)</th>
<th>Searching Time (ms)</th>
<th>Answer Quality P@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ht://Dig</td>
<td>(7)</td>
<td>(10) 104</td>
<td>(6) 32</td>
<td>-</td>
</tr>
<tr>
<td>Indri</td>
<td>(4)</td>
<td>(9) 63</td>
<td>(2) 19</td>
<td>(2) 0.2851</td>
</tr>
<tr>
<td>IXE</td>
<td>(8)</td>
<td>(4) 30</td>
<td>(2) 19</td>
<td>(5) 0.1429</td>
</tr>
<tr>
<td>Lucene</td>
<td>(10)</td>
<td>(2) 26</td>
<td>(4) 21</td>
<td>-</td>
</tr>
<tr>
<td>MG4J</td>
<td>(3)</td>
<td>(8) 60</td>
<td>(5) 22</td>
<td>(4) 0.2480</td>
</tr>
<tr>
<td>Swish-E</td>
<td>(5)</td>
<td>(5) 31</td>
<td>(8) 45</td>
<td>-</td>
</tr>
<tr>
<td>Swish++</td>
<td>(6)</td>
<td>(3) 29</td>
<td>(10) 51</td>
<td>-</td>
</tr>
<tr>
<td>Terrier</td>
<td>(9)</td>
<td>(7) 52</td>
<td>(9) 50</td>
<td>(3) 0.2800</td>
</tr>
<tr>
<td>XMLSearch</td>
<td>(2)</td>
<td>(1) 22</td>
<td>(1) 12</td>
<td>-</td>
</tr>
<tr>
<td>Zettair</td>
<td>(1)</td>
<td>(6) 33</td>
<td>(6) 32</td>
<td>(1) 0.3240</td>
</tr>
</tbody>
</table>
Selecting an Appropriate Ranking Function


<table>
<thead>
<tr>
<th>Feature</th>
<th>$\text{MRR}(r, S_{2-50})$</th>
<th>Feature</th>
<th>$\text{MRR}(r, S_{&gt;50})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.54</td>
<td>SVM</td>
<td>0.26</td>
</tr>
<tr>
<td>LexORD</td>
<td>0.53</td>
<td>LexORD</td>
<td>0.18</td>
</tr>
<tr>
<td>Selective</td>
<td>0.5</td>
<td>Selective</td>
<td>0.17</td>
</tr>
<tr>
<td>UserBest</td>
<td>0.47</td>
<td>AccessDate</td>
<td>0.16</td>
</tr>
<tr>
<td>UpdateDate</td>
<td>0.43</td>
<td>UserBest</td>
<td>0.16</td>
</tr>
<tr>
<td>Name</td>
<td>0.43</td>
<td>UpdateDate</td>
<td>0.15</td>
</tr>
<tr>
<td>AccessDate</td>
<td>0.4</td>
<td>CreateDate</td>
<td>0.12</td>
</tr>
<tr>
<td>CreateDate</td>
<td>0.39</td>
<td>Name</td>
<td>0.1</td>
</tr>
<tr>
<td>Size</td>
<td>0.39</td>
<td>Path</td>
<td>0.1</td>
</tr>
<tr>
<td>Content</td>
<td>0.38</td>
<td>Size</td>
<td>0.08</td>
</tr>
<tr>
<td>NormalizedSize</td>
<td>0.36</td>
<td>QueryLog</td>
<td>0.07</td>
</tr>
<tr>
<td>Path</td>
<td>0.34</td>
<td>DirRank</td>
<td>0.06</td>
</tr>
<tr>
<td>QueryLog</td>
<td>0.34</td>
<td>Content</td>
<td>0.06</td>
</tr>
<tr>
<td>DirRank</td>
<td>0.33</td>
<td>NormalizedSize</td>
<td>0.06</td>
</tr>
<tr>
<td>Level</td>
<td>0.31</td>
<td>Level</td>
<td>0.03</td>
</tr>
<tr>
<td>Random</td>
<td>0.28</td>
<td>Random</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Lucene-based DS prototype 
19 volunteers.
In total 1219 queries
188 queries had a single result,
916 queries has 2-50 results
115 queries had over 50 results.

$$\text{Selective}_q(f) \triangleq \sum_{\text{Feature} \in \{\text{Name, Path, Content, QueryLog}\}} \frac{\text{Feature}_q(f)}{\text{nz(Feature}_q)}$$
Research prototypes and Semantic Desktops (continues)

- Beagle++ *(extended open source DS)*
- Semex *(includes Malleable Schemas)*
- Haystack and Magnet *(Semantic Web approach)*
- Stuff I’ve Seen *(Phlat predecessor)*
- Phlat *(was used as a basis for Windows DS)*
- PIA *(semantic desktop solution from DB area)*

Some more:
  - Gnowsis, CALO
Personal Information Management with Semex, Yuhan Cai, Xin Luna Dong, Alon Halevy, Jing Michelle Liu, and Jayant Madhavan. In SIGMOD 2005
Semex Features

- Highly database oriented approach
  - Resources connected through *Reference Reconciliation*
  - On-the-fly integration with external sources
  - *Malleable Schemas*  

- Interesting visualization, though a bit too complex for everyday users

- Search
  - Keyword search – IR
  - Domain restricted search (i.e., Organization) – Recent IR
  - Association queries (i.e., triples) – DB

- Less special things, but not very common:
  - Basic PIM ontology used as a *Domain Model*
  - All associations are stored in a database
Search Semex

3 Conferences for publishing Semex papers

105 Images in Semex papers

2398 Messages
2 Presentations
65 Articles

15 Persons working on Semex (though they are not named Semex)
User: Do I know this paper of Susan Dumais?
Semex: Yes, you once cited it.

User: I got to know Susan Dumais by citing her paper.

Latest Lineage
The last time we mentioned Susan Dumais is in an email.

Earliest Lineage
I got to know Susan Dumais by citing her paper.

Shortest Lineage
User: Do I know this paper of Susan Dumais?
Semex: Yes, you once cited it.
Semex: PIM Reference
Reconciliation: Challenges

1. Multiple Classes
2. Limited Information
3. Multi-value Attributes
4. Lack of training data

Article: $a_1$ = (“Distributed Query Processing”, “169-180”, $\{P_1, P_2, P_3\}$, $c_1$)
$a_2$ = (“Distributed query processing”, “169-180”, $\{P_4, P_5, P_6\}$, $c_2$)

Venue: $c_1$ = (“ACM Conference on Management of Data”, “1978”, “Austin, Texas”)
$c_2$ = (“ACM SIGMOD”, “1978”, null)

Person: $p_1$ = (“Robert S. Epstein”, null)
$p_2$ = (“Michael Stonebraker”, null)
$p_3$ = (“Eugene Wong”, null)
$p_4$ = (“Epstein, R.S.”, null)
$p_5$ = (“Stonebraker, M.”, null)
$p_6$ = (“Wong, E.”, null)
$p_7$ = (“Eugene Wong”, “eugene@berkeley.edu”)
$p_8$ = (null, “stonebraker@csail.mit.edu”)
$p_9$ = (“mike”, “stonebraker@csail.mit.edu”)
Haystack (1)

- Lots of separate info, Haystack stores in central repository.
- Easy to separate info from its form, easy to connect related info.
- Many people could share a single repository.

Haystack: Per-User Information Environment Based on Semistructured Data. David Karger, in “Beyond the Desktop Metaphor” edited by Victor Kaptelinin and Mary Czerwinski. 2007
Haystack (2)
Stuff I've Seen (SIS)

S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff I've seen: a system for personal information retrieval and re-use. In SIGIR'03
Phlat

Phlat is written in Microsoft Visual C# and uses the Windows Desktop Search indexing and search engine.
Personal Information Application

A layered framework supporting personal information integration and application design for the semantic desktop, Isabel F. Cruz, Huiyong Xiao, in VLDB Journal 2008

Using RDQL (RDF Data Query Language)
PIA: Ontology
PIA: Smart Browser
Just-In-Time Retrieval

• “Just-in-time Information — Proactively offering a user information that is highly relevant to what s/he is currently focused on” (Pattie Maes)
JIT Approaches

- Watson
- Remembrance Agent
- Jimminy

All approaches aim to suggest relevant information snippets when the user writes a document or an email

Some more:
QUESCOT, MarginNotes, Letizia, WordSieve, CALVIN, Kenjin
WATSON

- supports just-in-time access to task-relevant information
- a system gathers contextual information as a text of the document the user is manipulating
- proactively retrieves documents from distributed information repositories
- Potential problems:
  - managing interruptions
  - ranking suggestions

Figure 2: Watson is suggesting documents as a user is writing a paper.
Watson Architecture
Remembrance Agent (RA)

- Remembrance Agent (‘96) / RADAR later for Word

Rhodes, B. and Starner, T. The Remembrance Agent: A continuously running information retrieval system, in PAAM’96
“Jimminy provides information based on a person's physical environment: her location, people in the room, time of day, and subject of the current conversation.”

“Processing is performed on a shoulder-worn "wearable computer," and suggestions are presented on a head-mounted display.”
What is context?

• Synonyms for context: (user/application) environment, situation, state, scenario, task, …

• Elements of context:
  – Location
  – People
  – Activities (tasks)
  – Time of day, season, temperature
  – Objects and changes to objects
  – Emotional state
  – Focus of attention
Context on a Desktop

Resource as context

Interaction with resource as context

Sequence of access

Genre

Time windows

GPS location

Sender

TFxIDF

Reading time

Reference

Bookmarking

Web address

Printing document
Using Context to Improve Desktop Search

- Connections (HITS and PageRank on File traces)
- Confluence (HITS and PageRank on File traces and Window focus)
- SeeTrieve (TFIDF variant on text snippets graph)
- Method by P. Chirita and W. Nejdl, (PageRank on File traces)
Connections

- Tracing file system calls
- Temporal relationships between files
- Used to reorder content search results
- Relation window of N seconds
- Number of occurrences of a sequence of files

Figure 1: Architecture of Connections. Both applications and the file system remain unchanged, as the only information required by Connections can be gathered either by a transparent tracing module or directly from existing file system interfaces.

C. A. N. Soules and G. R. Ganger. Connections: using context to enhance file search. In SOSP ’05
**Confluence** is an extension to **Connections**

- **Confluence** records *window focus* events within the GUI, which are generated each time the user activates a different application window. These events are used to infer *task*.

- Contextual relationships can be used to augment traditional search methods with additional, conceptually related files that do not match the text query.

- *For example, if documents A and B are frequently accessed at similar points in time, this suggests a task commonality. Searches that return "A" now return "B" as well.*
SeeTrieve

- A personal document retrieval and classification system
- Considers only the text presented to the user.
- Identifies information about the task associated with a document.

Figure 1. SeeTrieve architecture.
Method by P. Chirita and W. Nejdl


Algorithm 3.1. Ranking Desktop Items.

Pre-processing:
1: Let $A$ be an empty link structure
2: Repeat for ever
3: If (File $a$ is accessed at time $t_a$, File $b$ is accessed at time $t_b$) AND ($t_a - t_b < \epsilon$),
4: Then Add the link $a \rightarrow b$ to $A$

Ranking:
1: Let $A'$ be an additional, empty link structure
2: For each resource $i$
3: For each resource $j$ linked to $i$
4: If ($\#Links(i \rightarrow j) > T$) in $A$
5: Then Add one link $i \rightarrow j$ to $A'$
6: Run PageRank using $A'$ as underlying link structure
Context Detection

- Lumiere (Bayesian User Models)
- Nepomuk (K-Medoids and TFIDF)
- TaskTracer and TaskPredictor (Naïve Bayes/SVM)
- SWISH (Probabilistic Latent Semantic Indexing)
- CAAD (GaP probabilistic model)

Some more:
  QUESCOT, EPOS, MyLifeBits, Lifestreams
Lumiere

E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of soft. In UAI'98

Goal:
- help assistant for MS Office 97
- predict if help is needed, if yes, what is the problem?

Tools:
- Bayesian User Models

Lessons learned:
- advise capabilities are of limited utility
- recommendations can be annoying
More or less organized folder hierarchy

Applications for supporting knowledge work with proprietary formats

Current desktop

Important/real files

Desktop Area

Knowledge work support by file organisation

Temporary storage

-> R&D in Personal Information Management (PIM)
Nepomuk (2)

Desktop with Nepomuk

• *Semantic Desktop:* Information layer on top of the desktop content (personal semantic web) allowing machines to process information and provide intelligent services

• *Social:* Exchange between desktops
Nepomuk (3)

Goal:
- task-based document clustering

Tools:
- mixture of TFxIDF and K-Medoids clustering

The final goal is CONTEXT-AWARE INFORMATION RETRIEVAL
TaskTracer and TaskPredictor

J. Shen, L. Li, T. G. Dietterich, and J. L. Herlocker. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In IUI'06

Goal:
- associate resources with user activities

Tools:
- adaptive file open/save dialog box
- Naïve Bayes/SVM classifiers for task prediction

Lessons learned:
- precision is about 80%
- data is very noisy, users forget to change a task
Goal:
- task-based windows clustering for intelligent interfaces

Tools:
- unsupervised learning: Probabilistic Latent Semantic Indexing

Lessons learned:
- precision is about 70%
- data is very noisy due to occasional windows' switches
CAAD

T. Rattenbury and J. Canny. Caad: an automatic task support system. In CHI ’07

Goal:
- task-based windows clustering

Tools:
- GaP probabilistic model for Context Structures
- concatenated filenames for labels

Lessons learned:
- relevance is useless, if novelty is important or information changes quickly
- user models are too broad or too narrow
Ontology-based user interaction context model (UICO) automatically derives relations between the model's entities and automatically detects the user's task.
Current State

- Automatic Task Detection is under active development
  - most publications are within 2006-2009 time interval
  - no perfect solution so far

- Task Detection is based on machine learning
  - Naïve Bayes, PLSI, SVM

- Training data is missing
  - Activity-Logging can be used for data gathering
Towards Requirements for Logging Desktop

- **Automatic**
- **Cross-application**
- **Implicit Feedback**
- **Privacy preserving**
- **Extensible**
Desktop Logging Framework

Sergey Chernov, Gianluca Demartini, Eelco Herder, Michal Kopycki, and Wolfgang Nejdl.
Evaluating Personal Information Management Using an Activity Logs Enriched Desktop Dataset in PIM 2008 Workshop

Timestamp, Google queries and result pages, URL, ...

Timestamp, subject, sent time, attachment, recipient, ...

Timestamp, application name, window title, created/activated/destroyed, ...

Start Logger
Stop Logger
Options
About
Exit Logger

4:05 PM
Thursday
9/13/2007
Supported notifications

<table>
<thead>
<tr>
<th>Notification</th>
<th>Email</th>
<th>Instant Messenger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conversation (start, active, finish)</td>
<td>Email (receive, reply, delete, move, print)</td>
<td>MSN, TIS, AIM, ICQ, Windows Live Messenger, Yahoo Messenger, Skype</td>
</tr>
<tr>
<td>Address book entry (create, modify, delete)</td>
<td></td>
<td>TIS, AIM, ICQ, Windows Live Messenger, Yahoo Messenger, Skype</td>
</tr>
<tr>
<td>Email folder (create, rename, delete)</td>
<td></td>
<td>TIS, AIM, ICQ, Windows Live Messenger, Yahoo Messenger, Skype</td>
</tr>
<tr>
<td>Submit Web form</td>
<td>Firefox</td>
<td></td>
</tr>
</tbody>
</table>
Collected Data

- 21 participants
- Average of 170 active logging days
- 2,828,706 Events
- Average of 2,815 distinct emails per user
- Average of 9,337 distinct URLs per user
- Average of 902 events per user per day
- Average 5 hours of active interaction per user per day
A glimpse into user behavior (1)

Sergey Chernov, Gianluca Demartini, Elco Herder, Michal Kopycki, and Wolfgang Nejdl. Evaluating Personal Information Management Using an Activity Logs Enriched Desktop Dataset in PIM 2008 Workshop

### Email reaction time

**Instant reader**

- 60.00%
- 30.00%
- 0.00%

**Moderate reader**

- 8.00%
- 4.00%
- 0.00%

**Graphs:**

- Time [minute]: 0, 10, 20, 30

---
A glimpse into user behavior (2)
Evaluation

• Evaluation frameworks:
  – Naturalistic (one-time evaluation in a natural environment with own data)
  – Longitudinal (studies over extended period of time with measurements at fixed points)
  – Case study (in-depth picture of few individuals behavior)
  – Laboratory (controlled scenarios)

• Could and should be combined with each other

• Challenges:
  – Lack of control over environment (unpredictable interactions)
  – Appropriate time intervals and study duration
  – Narrow scope of evaluation task
Evaluation Components: Participants, Collections, Tasks

• Participants
  – Compared to Web Search: harder to recruit, data is too sensitive, prototype must be more robust, more involvement is required, limited generalization, using “personas” – simulated users

• Collections
  – Users should provide their own data, it is a mixture of documents, photos, emails, contacts, etc.

• Tasks
  – Tasks are broad, user-centric and situation-specific
  – Different granularity level (doing email vs. search for a piece of text in email)
  – Different types of tasks (planning a travel, reading the news, finding information about X)
Evaluation Components: Baselines

- Solomon four group design

<table>
<thead>
<tr>
<th>Time</th>
<th>Period 1 (pre)</th>
<th>Period 2 (post)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental group</td>
<td>O₁</td>
<td>X</td>
</tr>
<tr>
<td>Control group</td>
<td>O₃</td>
<td></td>
</tr>
<tr>
<td>Experimental group</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Control group</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- O: Observation. X: Intervention

- Caveat: *Trained Incapacity* – users create unique ways of using tools that the original designers may not have intended.
Evaluation Components: Measures

• Measures could be defined in two ways:
  – Nominal – what is it? (Learnability is defined by a grade on a 5-point Likert scale)
  – Operational – how exactly it should be measured? (Learnability is a length of time it takes for a user to learn to use an interface)

• Standard usability measures:
  – Effectiveness, Efficiency, Satisfaction, Usefulness, Ease of use, Ease of learning

• Usability measures in PIM context:
  – Performance (recall/precision), Adoption and Use, Flow, Quality of Life
<table>
<thead>
<tr>
<th>Question</th>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Overall, I am satisfied with how easy it is to use this system</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>2. It was simple to use this system</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>3. I can effectively complete my work using this system</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>4. I am able to complete my work quickly using this system</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>5. I am able to efficiently complete my work using this system</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>6. I feel comfortable using this system</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>7. It was easy to learn to use this system</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>8. I believe I became productive quickly using this system</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>9. The system gives error messages that clearly tell me how to fix problems</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>10. Whenever I make a mistake using the system, I recover easily and quickly</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>11. The information (such as online help, on-screen messages, and other documentation) provided with this system is clear</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>12. It is easy to find the information I needed</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>13. The information provided for the system is easy to understand</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>14. The information is effective in helping me complete the tasks and scenarios</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>15. The organization of information on the system screens is clear</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>16. The interface of this system is pleasant</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>17. I like using the interface of this system</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>18. This system has all the functions and capabilities I expect it to have</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
<tr>
<td>19. Overall, I am satisfied with this system</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly disagree</td>
<td>strongly agree</td>
</tr>
</tbody>
</table>
Usability Questionnaire Example 2

Step 1: Read over the following list of words. Considering the product you have just used, tick those words that best describe your experience with it. You can choose as many words as you wish.

- Unattractive
- Fun
- Distracting
- Inconsistent
- Friendly
- Effective
- Bright
- Counter-intuitive
- Patronising
- Exciting
- Simplistic
- Organised
- Fresh
- Secure
- Irrelevant
- Consistent
- Easy to use
- Predictable
- Useful
- Satisfying
- Efficient
- Creative
- Annoying
- Accessible
- Dated
- Logical
- Inadequate
- Stimulating
- Comprehensive
- Time-consuming
- Intuitive
- Confusing
- Awkward
- Effortless
- Understandable
- Frustrating
- Expected
- Usable
- Dull
- Desirable
- Advanced
- Unpredictable

Step 2: Now look at the words you have ticked. Circle five of these words that you think are most descriptive of the product.
Summary and Challenges

• Desktop Search research just started 😊
• Main future directions are:
  – Logging of user activities and creating context-aware DS
  – Integration of metadata and fulltext search in personal repositories
  – Building social semantic desktop - collaboration, recommendation and knowledge sharing functionalities should extend basic information access on the desktop
  – Better understanding of user needs
  – Seamless integration of search and browsing behavior
We are hiring!

• Relevant Areas
  – Search and Information Retrieval
  – Information and Concept Extraction
  – Data Mining and Statistical Analysis
  – User Interface Engineering and Interaction Design
  – Semantic Technologies and Web 2.0
  – Multimodal Communication and Analysis
  – Social Software for Technology Enhanced Learning

• Phd and PostDoc positions
  – See handouts or [http://www.l3s.de/web/page23g.do](http://www.l3s.de/web/page23g.do)

• 6-months internships for Master Students
  – Send your CV (1-3 pages) and Research Statement (1-2 pages) to Prof. Wolfgang Nejdl (nejdl@L3S.de) or most relevant person from L3S

  – Further questions – come and ask now or write to chernov@L3S.de
References: Research DS prototypes


References: Just-In-Time Retrieval


References: Context-based DS


- Activity put in context: Identifying implicit task context within the user’s document interaction, Karl Gyllstrom, Craig Soules, Alistair Veitch. In IIiX 2008.


References: Context Detection Tools


- N. Oliver, G. Smith, C. Thakkar, and A. C. Surendran. Swish: semantic analysis of window titles and switching history. In IUI '06.

