

Beagle: Free and Open Desktop Search

Joe Shaw

GUADEC 2006
Vilanova i la Geltrú

26 May 2006



Novell.[®]

Overview

- What is Beagle?
- Prehistory: Dashboard
- Where Beagle is today
- Integration into Nautilus, Yelp, Deskbar, file selector
- A neat new hack by Robert Love
- Google Summer of Code
- The problem of metadata, in particular *implicit metadata*
- Blue-sky: Association browser, file system
- Q&A, T&A

What is Beagle?

Beagle is a per-user service which transparently indexes your data in real-time, while providing applications with an interface for searching that data.

What is Beagle?

Beagle ransacks your *personal information space* to help you find whatever you're looking for.

What is Beagle?

Beagle: Unearthing the porn you thought you had deleted... in front of your boss.

Why is Beagle?

Desktop search is becoming an increasingly important aspect of modern desktop computing.

The Trowbridge Paradox:

“The only way to find information is to know where it is.”

Why is Beagle?

Most desktop searches (Spotlight, Google Desktop, whatever is in Vista whenever it comes out) only focus on finding your data.

Beagle was instead created from the perspective of tying your existing data together. (Although it finds it pretty well too!)

Aside: Good Desktop Search is Hard

The Web is uniquely designed to make searching easier.

A vast number of interconnected documents makes for a (fairly) simple graph and allows things like PageRank.

The volume of data makes a browse interface impossible.

Aside: Good Desktop Search is Hard

Desktop metadata is meager by comparison. Very little data is linked. So much implied metadata is lost.

The Web is pretty much static and done in snapshots; desktop data changes all the time and has to be indexed in real-time.

Desktop applications hoard their data.

The Unix filesystem is inherently racy, and that make things challenging.

Aside: Good Desktop Search is Hard

Lastly, and perhaps most importantly, people *hate hate hate* desktop search.

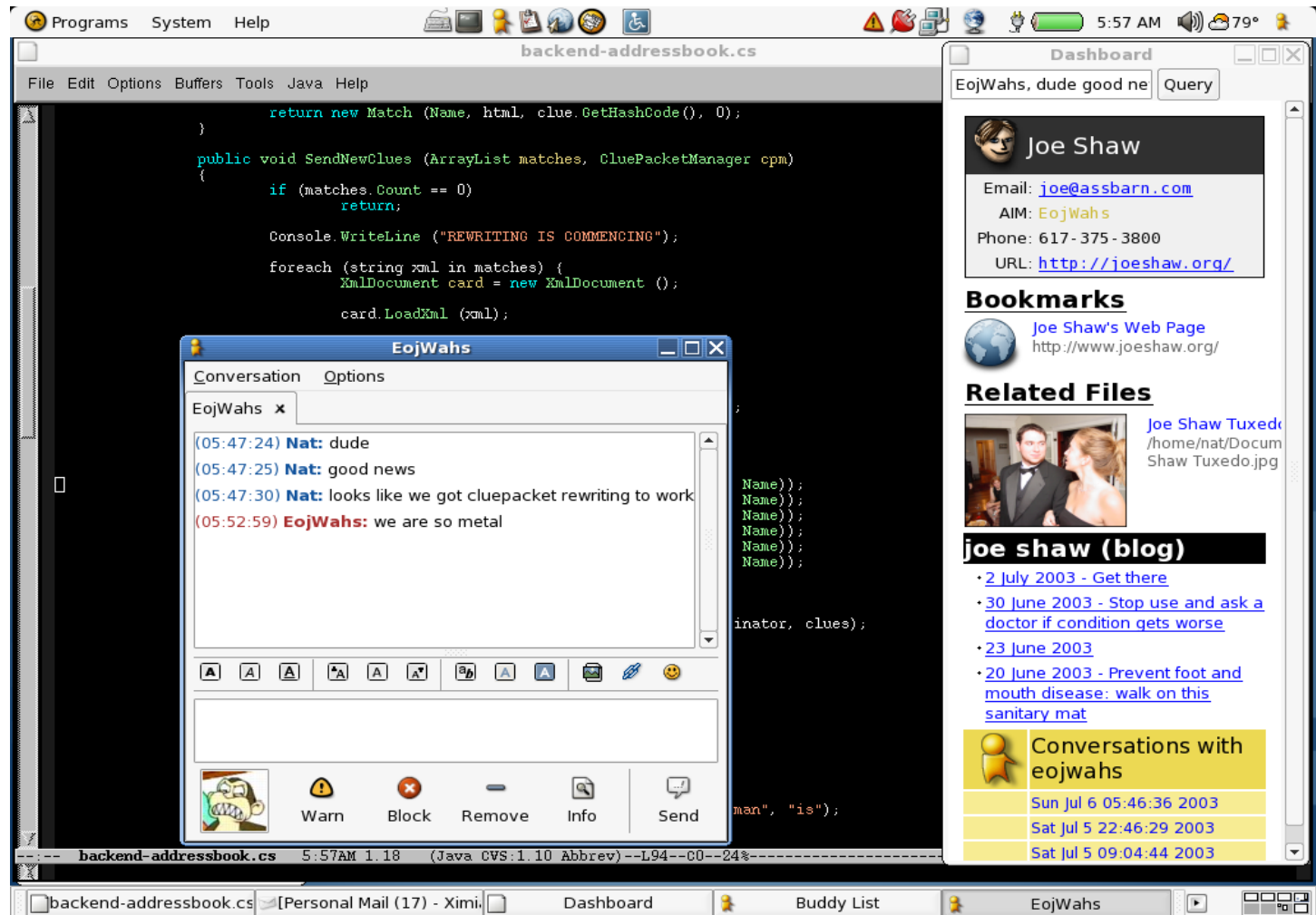
Beagle prehistory: Dashboard

Project started by Nat Friedman in 2003 to provide real-time information related to your current desktop context without any user interaction.

Prototype built in summer of 2003 by Nat, myself, Jim Krehl, Alex Graveley, and many others.

It worked! (Mostly.)

Beagle prehistory: Dashboard



Beagle prehistory: Dashboard

Found that while the basic concept of Dashboard was sound, bringing together all the user's very different data coherently and searching that data was impossible.

This problem is not unique to Dashboard!

Beagle!

Beagle project started in April 2004 by Jon Trowbridge and sponsored by Novell.

Beagle predates the current “Desktop Search” craze. It was first demoed by Jon at GUADEC 2004 in Kristiansand, Norway, hours before Apple first demoed Spotlight.

An Inspirational Dog

Beagle aims to be:

- Able to index every piece of your personal data
- Fast enough to search millions of documents quickly
- Ubiquitous in all desktop applications by providing simple but useful APIs
- Desktop neutral
- Extensible, so that adding new data sources or file types is easy.

Most Importantly,

SEARCH
SHOULD BE
EVERYWHERE.

Sniffing Out Your Data

Today, Beagle can search the following data sources:

- File System
- Evolution and Kmail E-mail
- Firefox, Epiphany, Konqueror web history
- Evolution Calendars and Addressbooks
- Gaim and Kopete Instant Messenger logs
- Tomboy notes
- Blam, Liferea, Akregator RSS feeds
- (Coming soon: Thunderbird E-Mail, Addressbooks)

Rooting Through Your Files

Beagle can parse and extract metadata from over 40 different file formats, including:

- Office documents (OpenOffice, MS Office, RTF, AbiWord)
- PDF
- HTML
- Many audio formats (mp3, ogg/vorbis, etc.)
- Image formats (jpg, png, gif, raw, svg, etc.)
- Many video formats (through Mplayer)
- Documentation (Docbook, man pages, CHM)
- Source code
- RPM, deb, ebuild

Digging Up Metadata

Beagle also pulls some metadata from external sources:

- F-Spot tags
- Nautilus emblems and notes

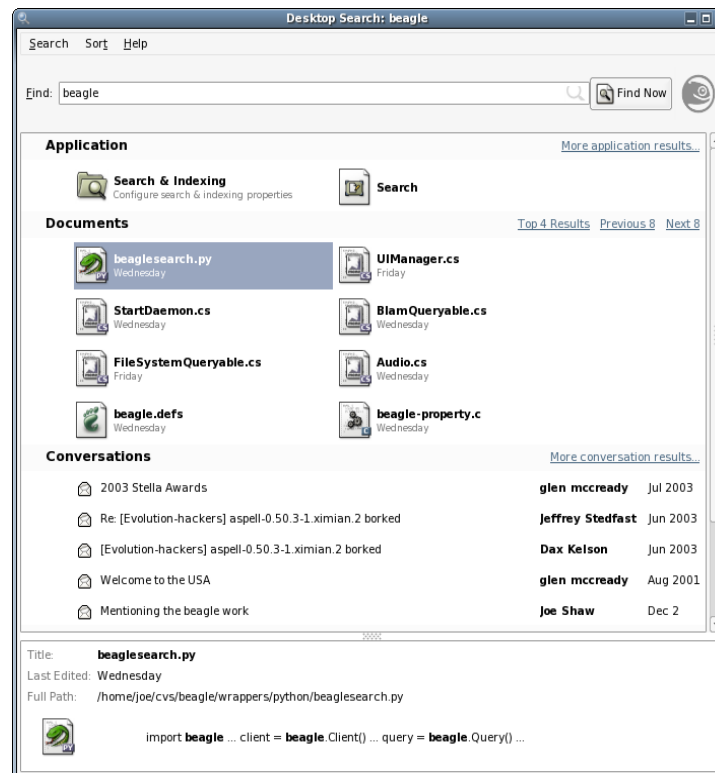
But more work needs to be done:

- Leaftag
- External applications
- Usage?

Winning “Best in Show”

There are several frontends to the Beagle daemon:

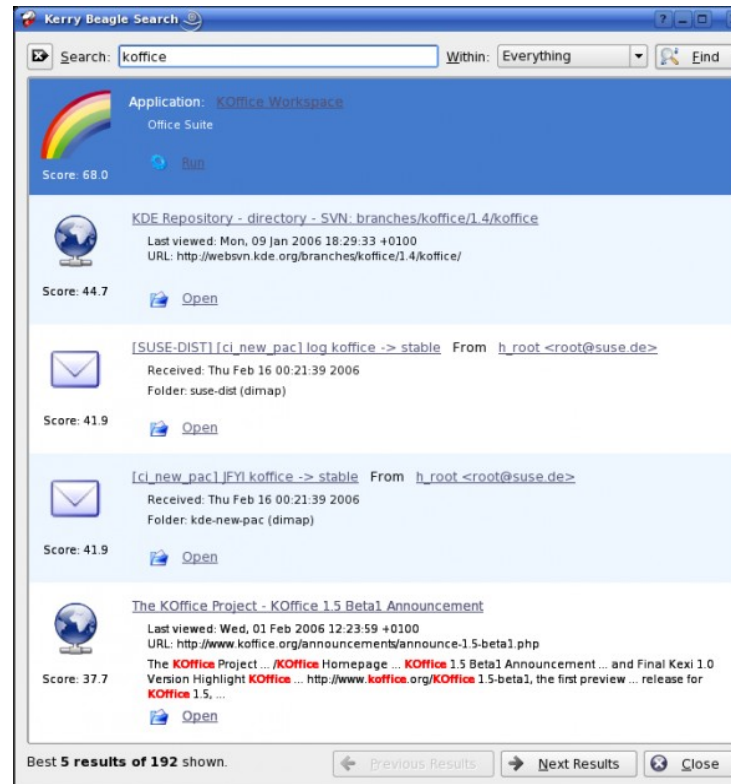
- Beagle Search (the included GNOME UI)



Winning “Best in Show”

There are several frontends to the Beagle daemon:

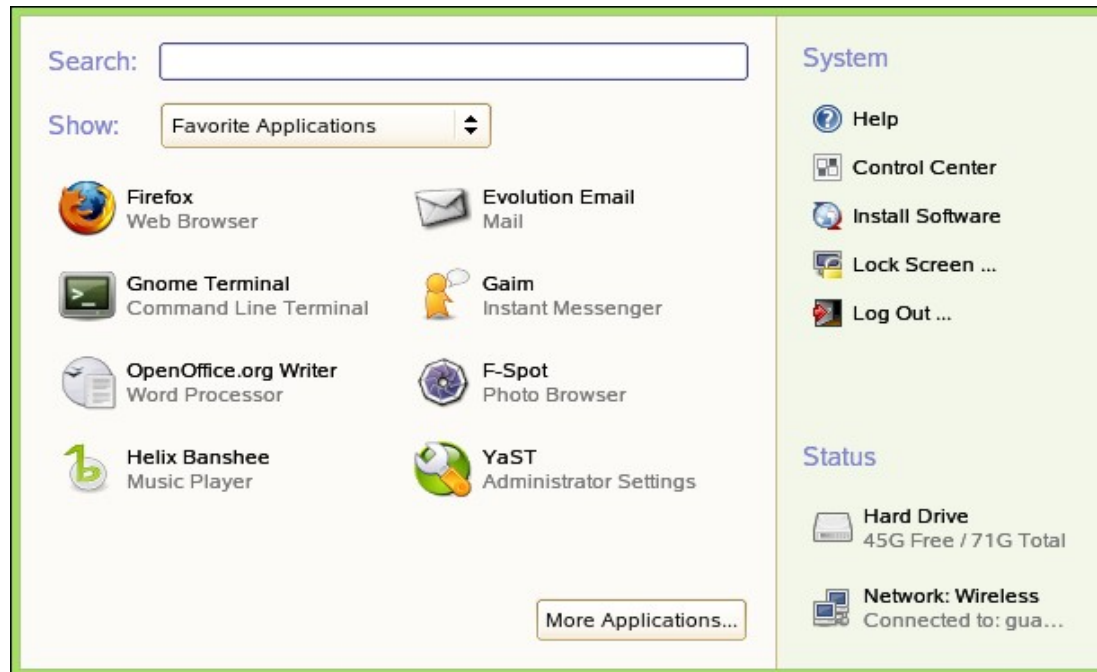
- Kerry (KDE)



Winning “Best in Show”

There are several frontends to the Beagle daemon:

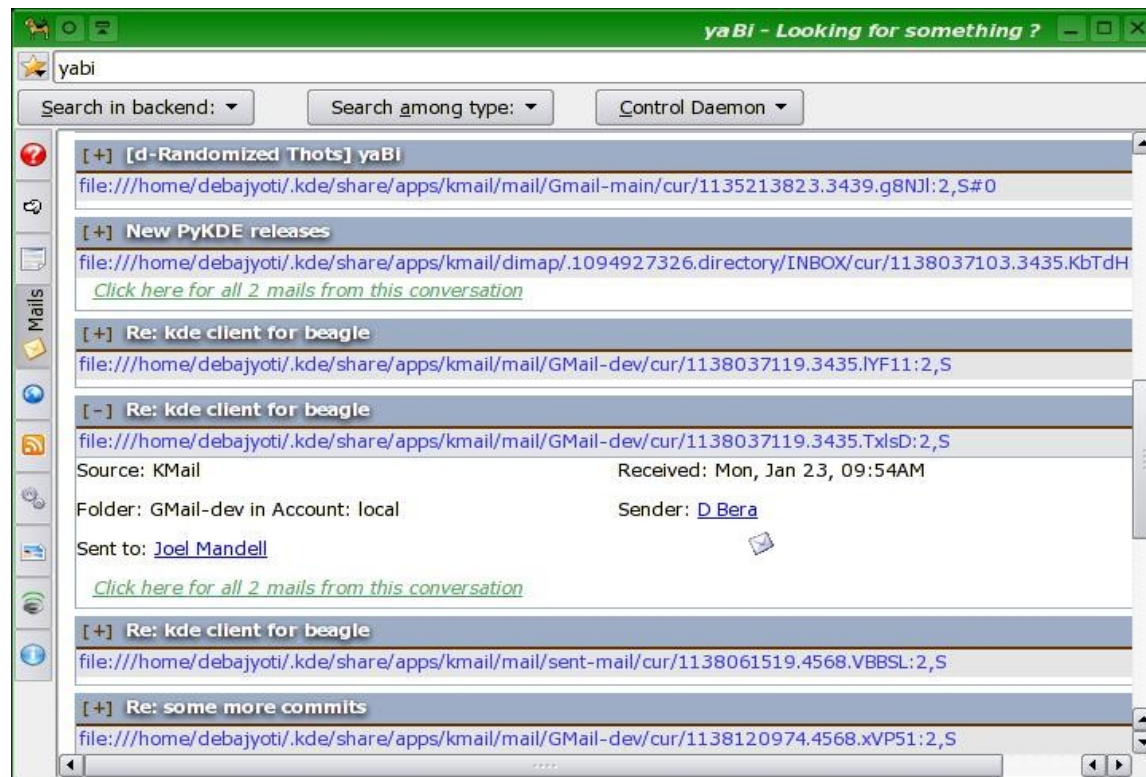
- The GNOME main menu included in SUSE Linux Enterprise Desktop 10



Winning “Best in Show”

There are several frontends to the Beagle daemon:

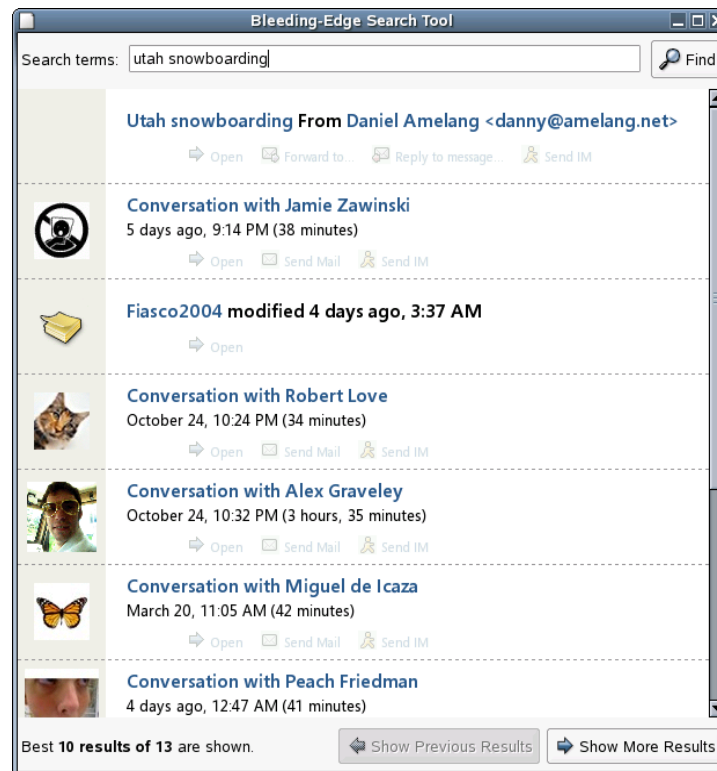
- yaBi (KDE)



Winning “Best in Show”

There are several frontends to the Beagle daemon:

- The original Beagle frontend, Best



Leaving “Surprises” in the Desktop

Today Beagle is integrated into an increasing number of desktop applications:

- Nautilus (file manager)
- Yelp (documentation)
- GTK+ file selector (external patch)
- Deskbar applet
- Banshee (experimental)

A Surprise of our Own

Integrating Beagle search into your application is so easy, Robert Love hacked up a FUSE filesystem called beaglefs in a couple of hours.

Worship him.



Joe Shaw – GUADEC 2006 – Vilanova i la Geltrú – 26 May 2006

beaglefs

- Just under 300 lines of code
- Queries are created using mkdir
- Files are symlinks to the *actual files on your system*
- Every file has all the Beagle-indexed metadata available as extended attributes

With this, we can create desktop-neutral, persistent search folders.

Worship him.

Like the Iditarod

Beagle has driven the development and/or deployment of many technologies, including:

- inotify
- heap-buddy, a profiling tool for Mono apps
- extended attributes
- search infrastructure in desktop apps
- tiles (See Jim Krehl's talk tomorrow!)
- better exposure and interaction with other apps and data

Man's Best Friend

Today Beagle is included in just about every major Linux distribution:

- SUSE Linux
- Fedora Core
- Ubuntu
- Mandriva

And some other interesting ones:

- Symphony OS
- Nexenta (Ubuntu on Solaris)

Bad dog

What we don't do so well:

- Removable media
- Index external metadata
- Memory usage over long periods of time; embedded devices

Where Beagle is Going

Although tweaks, bug fixes, and support for new backends and filters will always be on the radar, Beagle is at a point now where its purpose is (mostly) fulfilled.

We can focus on building some truly innovative technologies, like Dashboard.

Google Summer of Code

Beagle is participating in the 2006 Google Summer of Code, and five projects are being sponsored.

SoC: Networked Searches

Alexis Christoforides is implementing an HTTP-based interface to the Beagle daemon, so that data can be shared across machines.

Kyle Ambroff is adding Avahi integration to the network so that daemons can automatically discover and share information peer-to-peer. Kyle is also integrating networked search into the main user-interface.

SoC: Metadata

Max Wiehle is tackling the issue of how to best integrate Beagle with external metadata. Max is looking into building a metadata store on top of Lucene and SemWeb and indexing with external sources like Leaftag.

SoC: Memory profiling

Dennis Snell is enhancing the heap-buddy tool to help Beagle (and all Mono applications) detect and solve issues with memory consumption.

SoC: DASHBOARD!

Fredrik Hedberg, who interned at Novell last summer and worked on Beagle, is taking on the task of resurrecting Dashboard.

Additional work

Nice new features that would be very nice to add:

- Direct searching of the index so things like autocomplete are quick enough
- Evenly distributed indexes for more uniform search time
- Richer Firefox extension
- Better external indexer
- Implicit metadata

The Metadata Issue

What is the best way to handle metadata? Should Beagle be a centralized store for different kinds of metadata? Or should it simply index existing databases?

Implicit Metadata

Every day in the course of using your computer, you lose all kinds of metadata:

- Email attachments lose the sender, the topic, other files linked to it
- Files downloaded from the web lose their connection to the hosting site.
- When sharing documents, who has ever edited this document?
- Cut-and-paste?

Implicit Metadata

How do we store this information? Every approach has drawbacks:

- Extended attributes don't work on some file systems
- Sidecars have to be maintained and synchronized alongside the actual data
- In-file metadata varies from format to format, and in many cases isn't possible (plain text, for example)
- Centralized databases make change notification extremely difficult.

My vote? Extended attributes.

Blue Sky Projects

With search and data relationships in place, we can build all kinds of interesting tools.

Association Browser

The screenshot shows a window titled "window1" with a menu bar (File, Edit, View, Help) and a toolbar. The main content area is divided into several panels:

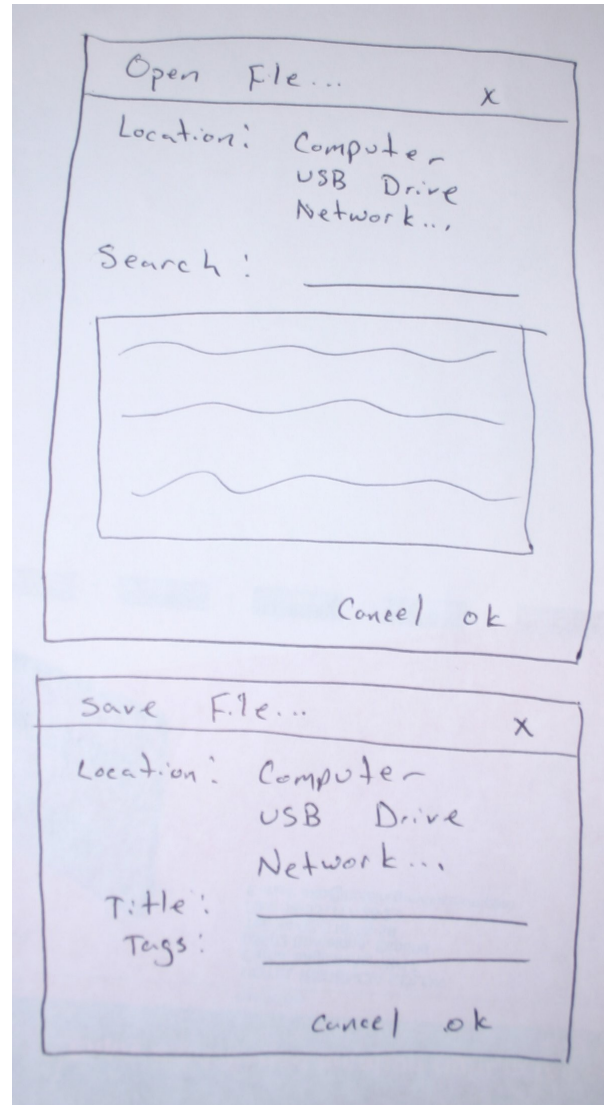
- Brand Grub:** Includes a logo and a "No. of emails: 42" indicator.
- Item info:** Shows "More Info" and "No. of emails: 42".
- User Profiles:** Displays three users: Anna Dirks (anna@ximian.com, Online), Bilbo Baggins (bilbo@baggins.com, Online (invisible1)), and Frodo Baggins (frodo@fellowship.net, Away (mshrmivr22)).
- Tasks:** A list of tasks with checkboxes: "Locate stuff", "Find items for grub splash" (checked), "Do things", and "Look for object".
- Calendar:** A calendar for June 2004 with the 8th highlighted.
- Related Bugs:** A list of four bugs, all titled "Ring needs watching" (Bug #223491, SEVERE, UNCO).
- Email Threads:** A list of email threads, including "RE: [gst-devel] RE: Win 32 build of 0.8.3" and "RE: [gst-devel] Re: [gst-cvs] thomasvs gst-pl...".
- Web:** A list of web links, all titled "The Ballad of Bilbo Baggins", with details like "Spock Sings! Leonard Nimoy..." and "First seen: 07/23/02 2:32am".
- Related Files:** A grid of folders and files, including "CVS", "XD2-installer", "evolution", "gdm", "glade", "html", "windowmanager", "yelp", and "ChangeLog".

Search-based File Dialogs

For the past year, I've had two drawn mockups on my whiteboard.

I would like to see the file system become an implementation detail (like HAL did for /dev nodes), and have all file access be based around search.

Search-based File Dialogs



About Beagle

- Web site: <http://beagle-project.org>
- Mailing list:
 - dashboard-hackers@gnome.org
 - <http://mail.gnome.org/listinfo/dashboard-hackers>
- IRC: [irc.gimp.net #dashboard](irc://irc.gimp.net/#dashboard)

About me

·Email: joeshaw@novell.com

Feel free to ask me any questions after the talk or at any point during the conference!

There is going to be a Beagle BOF on Friday at 10:00!
Let's make search ubiquitous!

Questions?

Bueller?

Novell®

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. Novell, Inc., makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

