

PIM Analysis and Development: Improving Information Visibility In A Micro-Note Taking Tool

by

Wolfe B. Styke
S.B., C.S. M.I.T., 2012

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

Copyright © 2012 Wolfe B. Styke. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
to distribute publicly paper and electronic copies of this thesis document in whole and in part in
any medium now known or hereafter created.

Author:

Department of Electrical Engineering and Computer Science
May 21, 2012

Certified by:

David R. Karger
Professor
Thesis Supervisor
May 21, 2012

Accepted by:

Prof. Dennis M. Freeman
Chairman, Masters of Engineering Thesis Committee

PIM Analysis and Development: Improving Information Visibility In A Micro-Note Taking Tool

by

Wolfe B. Styke

Submitted to the Department of Electrical Engineering and Computer Science
on May, 21, 2012, in partial fulfillment of the requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The digital age has brought about an information overload for individuals. While digital personal information management (PIM) tools are advancing, any one PIM tool generally fails to meet all the information management needs of an individual. Tools that require too much effort to save a new piece of information cause users to record less of their information, which often results in these pieces of information becoming fragmented and/or lost. People frequently lack the right information at the right time, even if the information is kept by one of their PIM tools. Individuals either forget that they previously recorded a piece of information, or are away from their digital devices at the wrong times.

How can people keep better track of more of their information? Micro-note tools, which require minimal time, effort, and distraction for the task of capturing new pieces of information may be part of the answer. This thesis evaluates the List-it micro-note PIM tool, a Firefox add-on that lets users take notes in a browser sidebar. This evaluation has led to a better understanding of how users interact with and view PIM tools.

An analysis of two years of List-it usage led to many insights into PIM tool usage. An examination of List-it users resulted in the discovery of four types of interaction styles that describe how users interact with their PIM tool: Packrats, Minimalists, Sweepers, and Revisers. An examination of notes kept in List-it identified five major categories of personal information: reminders (memory triggers), reference items, journal/diary entries, scratch pad (external cognition), and posterity. The analysis also resulted in several hypothesized unmet needs and the development of new features to meet these needs, which focused primarily on improving the visibility of notes. Finally, a new application was designed and constructed that, as a single codebase, produces a Chrome Extension, a Firefox add-on, and a regular multi-browser compatible website for the new features of List-it. This thesis is one iteration in the understanding of and design for future needs faced by users of PIM tools.

Thesis Supervisor: David R. Karger

Title: Professor

1. Introduction	5
2. Personal Information Management	7
<i>2.1 Information</i>	7
<i>2.2 Personal Information Management</i>	7
<i>2.3 PIM and Situational Awareness</i>	8
<i>2.4 Fundamental PIM Activities</i>	9
3. List-it Background Information	11
<i>3.1 Description of List-it</i>	11
<i>3.2 Motivation for List-it's Creation</i>	13
<i>3.3 Why Individuals use List-it</i>	13
<i>3.4 How and Why Individuals use List-it</i>	16
<i>3.5 Tracking User Activity</i>	20
4. List-it User Activity Analysis	21
<i>4.1 Types of Users</i>	21
<i>4.2 Distinguishing Interaction Styles</i>	26
<i>4.3 Categorization of Interaction Style</i>	27
5. List-it Note Analysis	30
<i>5.1 Note Contents</i>	30
<i>5.2 Note Roles</i>	31
<i>5.3 Manual Categorization of Notes by Role</i>	34
<i>5.4 Automatic Categorization of Notes by Role</i>	35
<i>5.5 MIT Information Scraps Corpus</i>	38
6. New Features for Unmet Needs	40
<i>6.1 Buried and Forgotten Notes</i>	40

<i>6.2 Feature: Pinned Notes</i>	<i>41</i>
<i>6.3 Scanning Note List Chaos with Minimized Notes</i>	<i>42</i>
<i>6.4 Viewing Notes in Context</i>	<i>43</i>
<i>6.5 Context Retrieval for Notes and Automatic Bookmarking</i>	<i>44</i>
<i>6.6 Summary of Features</i>	<i>45</i>
7. New List-it Application	46
<i>7.1 Motivation for New Application</i>	<i>46</i>
<i>7.2 List-it's New User Interface</i>	<i>47</i>
<i>7.3 Design of Chrome Extension</i>	<i>50</i>
<i>7.4 Re-design for Web Compatibility</i>	<i>51</i>
<i>7.5 Firefox Add-on Design</i>	<i>52</i>
<i>7.6 Future Research Potential</i>	<i>52</i>
8. Conclusion	54
9. References	56

1. Introduction

One of the effects of living with electric information is that we live habitually in a state of information overload. There's always more than you can cope with. Our Age of Anxiety is, in great part, the result of trying to do today's job with yesterday's tools and yesterday's concepts.
- M. McLuhan, *The Medium is the Message* (1964).

With the increasing variety of electronic tools for keeping track of personal information, why is it that personal information is still lost, forgotten, or not recorded for later use? Personal information can be useful in carrying out a wide variety of everyday tasks, from the workplace to the home environment and everywhere in between. Any problems a user experiences with a personal information management (PIM) tool can have an impact on just about any aspect of their life. The large variety in types of personal information result in people having fragmented collections of their personal information spread across many tools that each focus on one of their needs.

List-it is a micro-note PIM tool that attempts to be a catch-all for the information scraps that do not fit in the other tools our users utilize. Users can keep different types of notes without the extra cost associated with fitting a note into an existing categorization. By accommodating vastly different types of personal information, List-it helps users reduce the fragmentation of their personal information, since they can find more of their information in one place. Studying the usage of micro-note PIM tools reveals insights into why users choose to store their information in a less structured PIM tool.

There are many important questions about List-it's users' behaviors that provide insight into improving PIM practices. The first of three parts of this thesis explores the primary characteristics that describe how users use List-it in their PIM practices. These characteristics are examined from one angle through a user survey of 255 List-it users. They are also looked at more quantitatively through analysis of the actions individual List-it users take while using the List-it tool. There are many questions examined that revolve around three key themes. What types of notes to users keep? What types of users use List-it and how do different users use List-it over time? How does context, such as time or the website being viewed, play a role in a user

interacting with their personal information kept in List-it? These questions focus on the practices of List-it users to better understand the List-it's users and the notes they keep.

The second part of this thesis describes the design and motivation for new features that will improve a List-it user's ability to keep track of their personal information. From the surveys and analysis work, four new features have been identified that may help users keep better track of their important information scraps. Three of these features focus on improving the visibility of individual notes in different contexts. One feature focuses on the potential relationship between the currently focused website in a browser and the notes that may be relevant to the user while browsing the website. The goal of these features is analogous to the goal of List-it, enabling users to more easily re-find information that they would either not record, or lose track of in another tool.

The third and final part of this thesis is about the creation of a new multi-browser and multi-browser-extension application for List-it. This application started as a Chrome Extension to meet the users' demands for a Chrome version of our Firefox add-on. Through careful system design, the Chrome Extension could be structured to function as a regular website using the same codebase. The new List-it codebase also serves as a Firefox add-on. This allows our research group to maintain only one codebase to support a Firefox add-on, a Chrome Extension, and a regular website. The website has not been optimized for viewing on mobile devices. This wide reach allows our research group to accommodate more users on different browsers, further increasing our ability to collect usage information and better understand how our users interact with List-it.

The work in this thesis provides a better understanding of how users interact with a micro-note PIM tool. Through examining our users, the notes they keep, and how context affects their usage of List-it, several unmet needs are identified. For each unmet need, a new feature is proposed and implemented in the new List-it application for further analysis.

2. Personal Information Management

This chapter describes related work in the area of Personal Information Management (PIM). Views of information and personal information are described, followed by key activities of personal information management.

2.1 Information

Information means something different in every field of science. The information stored in DNA, the information stored by quantum computers, and the information stored by users of PIM tools are all vastly different types of information. Information Science [1] as a field has chosen to use the definition provided by Brooks' "The Fundamental Equation for Information Science" [2] to describe information. His definition of information is "that which changes the knowledge structure of an individual by some amount." Unlike the fairly fixed information in a strand of DNA, this definition allows information to be weighed on a relative scale. A piece of information may be more valuable if it is observed when it is needed, but if observed too late, the same piece of information could be entirely worthless. While the fragment of information didn't change, its value is dependent upon the conditions in which it is observed and recorded.

2.2 Personal Information Management

Personal Information Management systems are used by individuals to capture, store, and re-access pieces of information for personal use. Boardman defines personal information management as the practice of keeping information for your own use [3]. This subset is distinguished from general information management, in which individuals keep information for other peoples' purposes. Jones [4] uses a less strict definition, including information that can be described by the following categories:

1. Information kept for an individual's own use.
2. Information about a person kept by and under the control of others.
3. Information experienced by a person (but outside his or her control).

The first category is similar to Boardman's, but the second and third categories extend past Boardman's definition. The second category includes a wide range of information in both

offline and online environments. From a store's records of what items are bought by which customers in each of their stores, to a website's records showing how users navigate through and use the website, there is an increasingly large set of data stored by external agents about people's personal actions. The third category is very wide in scope, and includes information that may inform an individual but which was not kept track of or managed by that same individual.

There is a fourth category of personal information not mentioned in the above categories, which includes "the set of information that affects or impinges upon an individual, of which the individual is not aware." [5] Similar to the concept of a knowable unknown, this category describes information that is available to an individual but has not been accessed by an individual. There is an overwhelming amount of information available to individuals in society, which consequently means there is an overwhelming amount of information that is not being accessed, but is available for access, by an individual. For example, consider a tourist traveling from one town to another. A bridge along the route has recently been closed, and this information is broadcast over the radio and sent to car GPS units which are notified to plan a route around the obstacle. If the tourist is not listening to the radio station announcing the traffic, or does not have a GPS unit for their car, then they won't find out about the information of the bridge closure, which will cause a delay in their trip. This example shows how important information may be present in the environment but not available due to a lack of the right tools.

2.3 PIM and Situational Awareness

The bridge closure example from the previous section was a piece of information available in the environment of which an individual was not aware. Situational awareness is an abstract term that describes the amount of accessible information, which relates closely to an individual and can be found in the individual's environment, that the individual knows about. Research from human factors [6] has built a theory for situational awareness as it relates to individuals with important decision-making processes, such as pilots and air traffic controllers.

A valuable goal for PIM systems is to increase a user's situational awareness. In section 6, a new feature for List-it called "Notes that Float" increasing a user's situational awareness. When the user opens a new website, the feature shows the user notes that were created while

they were viewing the same website previously. Without this feature, non-recent notes that a user kept while viewing a site would remain buried and out of sight. This feature is not designed to be used for a single purpose. Users who view List-it as a password management tool can let List-it automatically retrieve their account information when they visit login pages. Other users write blogs by first writing summaries of websites in List-it. When this type of user returns to the website they were writing a review about, their review will float to the top of their list. For information that a PIM tool contains about an individual, automatically recognizing which pieces of information are the most relevant at a given time will save the user from making decisions without the information they need. This is a difficult task, and the “Notes that Float” feature is a step in the right direction.

2.4 Fundamental PIM Activities

What basic activities does an individual managing personal information take? Jones [4] suggests that there are four fundamental activities involved in PIM. These activities include:

1. *Finding/Seeking* - Acquiring a piece of information for the first time from an external source, such as from a website or communication channel.
2. *Capture/Keeping* - Storing the acquired information, typically motivated by the perception that the information will be valuable to the individual at some point of time in the future.
3. *Re-finding/Re-searching* - Seeking a piece of information that has been found previously. Either the information is retrieved from the individual’s PIM tool, or is found in the original location where it was located previously.
4. *“M-level” activities* - “Meta-level activities” that help structure an individual’s information to make it easier to capture new information and find previously stored information.

List-it makes it easy to transition from step 1 to step 2 by being available in the sidebar of a browser when the user finds a piece of information from a website they wish to capture. Capturing information in List-it is made as easy and fast as possible, with a single text box and a

save button. Several new features that support re-finding of information stored in List-it are presented in Chapter 6. List-it uses as few “M-level” activities as possible. The original List-it design featured a single list of notes that could be re-ordered. Users could search notes by the text they contained, and could create “saved search” buttons that when clicked would perform the same search that was active when the button was created.

3. List-it Background Information

This chapter describes the motivations behind the old List-it application and information collected from users through surveys. The surveys capture why users value List-it, how they think List-it meets their needs, and what they use List-it to accomplish.

3.1 Description of List-it

List-it is a firefox add-on that provides a sidebar in the user's browser for jotting down notes while browsing the internet. List-it provides fast, lightweight, unstructured note capture with no upfront categorization required. Fast note capture is achieved with List-it allowing a user jot down a note and click save, with no other tasks required to save the note. These properties make creating new notes as effortless as possible, allowing users to capture more notes that they may not have captured if the process were made more difficult or time-consuming.

List-it currently has 20,000 registered users, with potentially more unregistered users. Roughly 10% of our registered users have signed up for our ongoing study, which lets us record roughly 2000 users' interactions with List-it. The more useful results and analysis work from the analysis of these List-it users' interactions with their List-it are presented in this thesis. In addition, the creation of a new platform for expanding access to potential research participants with new features informed from the analysis of our current users is described in a later section.

The old List-it UI, shown in the below screenshot, is kept fairly minimal on purpose. List-it provides the basic actions of creating, searching, editing, and deleting information items. Chapters 6 and 7 describe a new List-it application designed to meet needs and use cases explored in Chapters 3, 4, and 5.

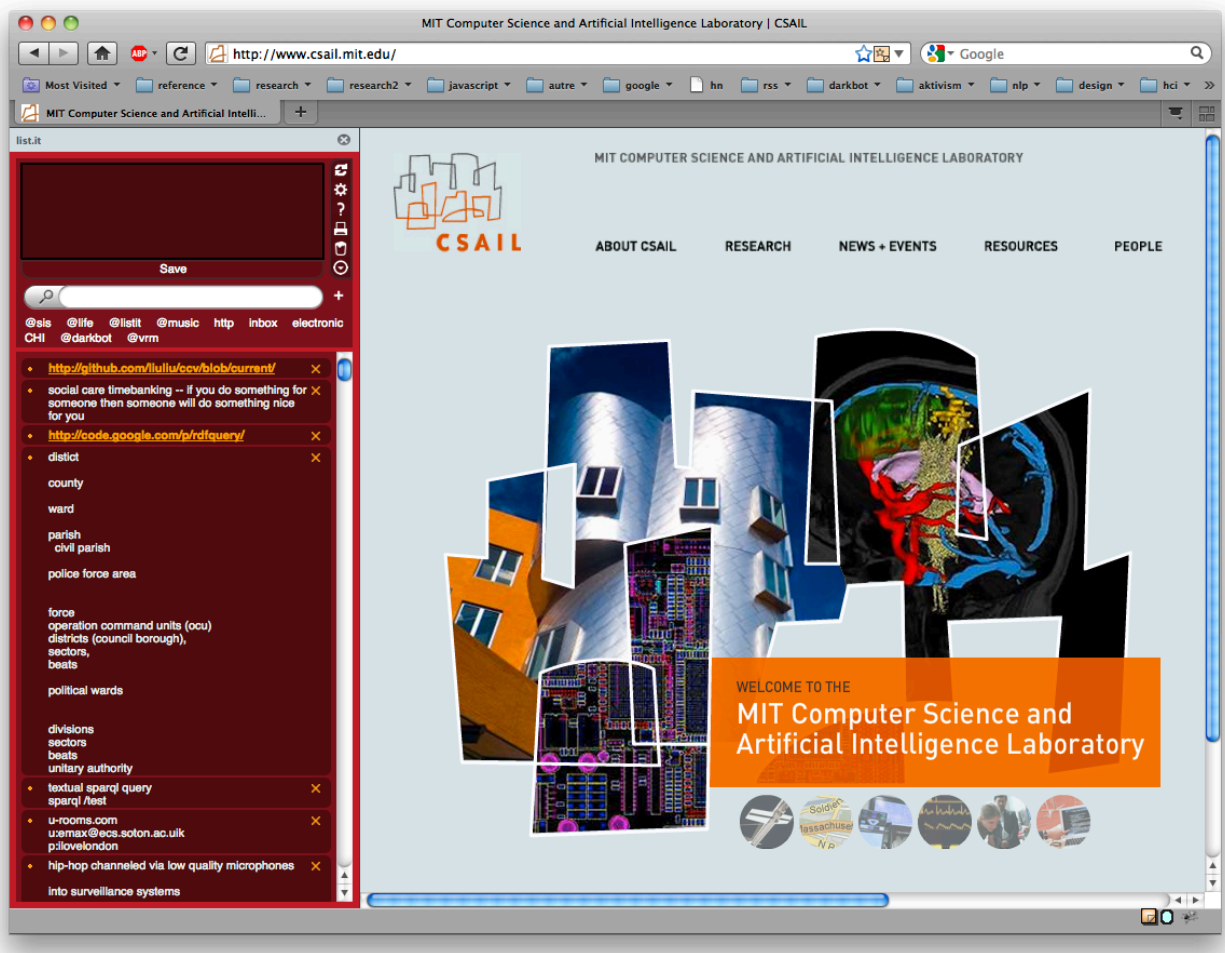


Figure 1. The List-it Add-on is displayed as a sidebar next to the currently viewed website to allow easily using and viewing both List-it and the currently selected website at the same time.

The List-it application presents a limited set of operations necessary to manage a list of notes. By making the main operations for creation, searching, editing, and deletion as easy as possible, List-it attempts to reduce the feeling a user may have that they are “operating” a tool. Instead, List-it should be there when a user needs it without taking too much attention away from what the user is currently doing. At the top of the sidebar is the new note creation box. Below it is the search bar, which provides incremental search as you type, and an X icon for removing the current search term. Below the search bar is a saved-searches bar, which lets users save common searches as a button for faster re-finding later. The list of notes is chronologically ordered, with

the most recently created notes placed on top. The orange dot on each note lets the user drag a note to rearrange their list, and the orange X icon deletes a note.

List-it provides a syncing service for users to be able to backup their notes on our central server. This also allows List-it's users to synchronize multiple Firefox add-ons on different computers so they can have a copy of their notes regardless of which of their machines they are using. In addition to syncing Firefox add-ons, we have a simple website that lets users who sign up gain access to their notes when they are away from their main machine that has List-it installed.

3.2 Motivation for List-it's Creation

List-it was purposely designed as a micro-note tool to help users capture pieces of information that didn't fit into their current note taking strategies with more structured tools. We call these pieces of information "information scraps" [7]. Because List-it is a catch-all for all the information scraps that do not fit into the user's existing PIM tools, users end up storing a wide variety of notes in List-it. This means different users may use List-it for vastly different purposes, depending on the types of information they work with that fall outside the boundaries of their other PIM tools.

As a Firefox add-on, List-it is able to provide a place to store and retrieve a user's information scraps that is visible next to the webpage a user is visiting. This makes it easier for a user to switch between using List-it and a website, compared to switching between a browser and another tab or application. As could be expected, this has resulted in users storing web-related information scraps (such as bookmarks and passwords) in List-it.

3.3 Why Individuals use List-it

In September of 2009 we conducted a survey of 225 users of List-it to better understand more about our users and the value List-it provides to them. We also asked about their general PIM tool usage and how they use List-it to better keep track of their information scraps.

We received interesting and motivating feedback by asking how users kept track of information scraps before they used List-it, and how List-it changed their PIM habits. There

were four trends in responses to how users previously kept track of information and how List-it has helped change their PIM habits. Selected quotes from our survey responses for each of these three trends are shown below.

The first trend was that prior to using List-it, users didn't store the types of notes they kept in List-it in any major tool, or even at all.

- *"I never wrote any of the information that I put in list-it down before"*
- *"I tried to keep them in my mind but i kept forgetting stuff mostly in my head, occasionally on paper"*
- *"Information is available in my browser where I will probably use it instead of paper where I will probably lose it."*
- *"If I remember correctly I used my memory, I think, I can't remember for sure"*
- *"Information I expect to forget if I don't write it down"*

The second trend revealed how the ease of capturing new notes was critical to capturing information scraps that might otherwise be lost to memory.

- *"List-It provides a mechanism that I can (easily -- this is key) record immediate, or very soon, activities and actions to be done."*
- *"List.it is handier to open. This ready availability is a big advantage"*
- *"Also, I write down more things instead of just trying to remember them."*

The third trend revealed how being able to synchronize notes across all of a user's computers was very useful for creating and later re-accessing a user's notes.

- *"List it is great because it allows me to sync lists between computers, and it is fast and I never lost a information on listit."*
- *"mostly convenience across multiple systems as I move around -kept them in sync"*
- *"makes it easier to synchronize the list from home or work so I can add to it from either location. this advantage is why i use it."*
- *"post it notes on monitor. Notes better organized, can see them from both work and home."*

The fourth insight revealed a change in browser preference for some of our List-it users, from Mozilla Firefox to Google Chrome.

- *“I have pretty much stoped using List.it, though, because I migrated from FF to Chrome a few months ago. I would go back to List.it though if it would run there.”*
- *“It must work with a multiplicity of browsers, Please make Chrome your next!”*
- *“But these days I also use Chrome. Therefore, it becomes little limited in use.”*

List-it users use a variety of other tools for managing their personal information. In our survey we asked users of List-it which other tools were used by our users. We found a general preference for web-based tools over the desktop application alternatives as the following graph shows.

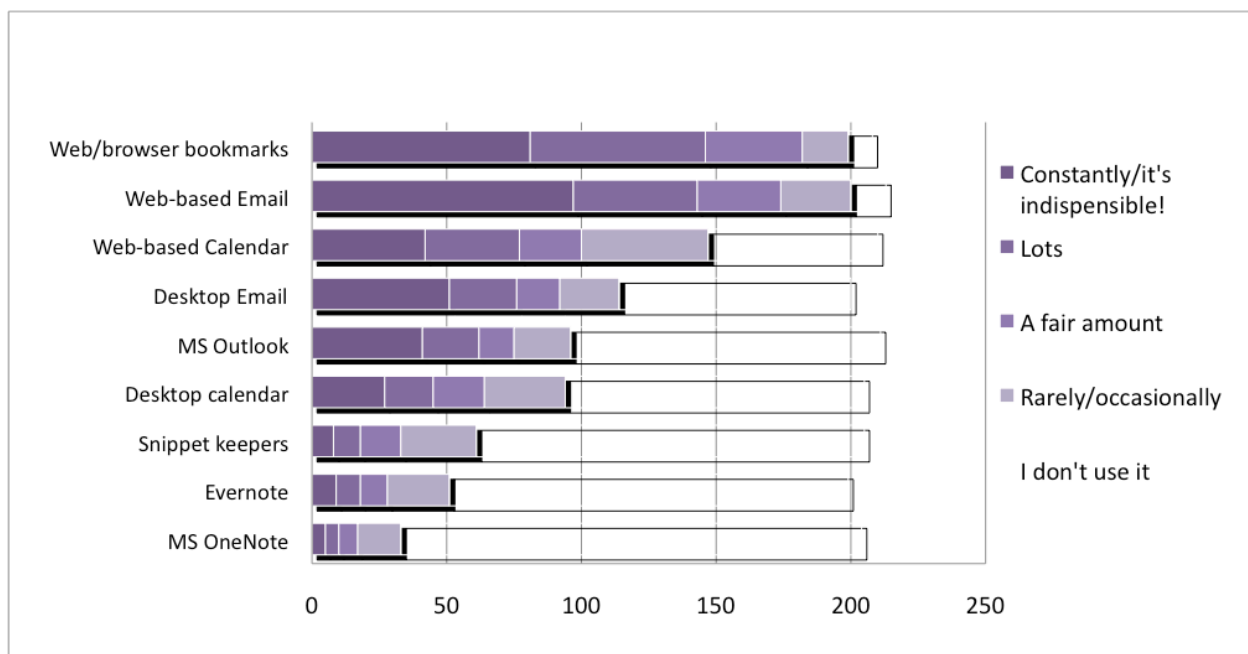


Figure 2. The number of survey responses (shown on the x-axis) for each frequency of using the different PIM tools. Web-based PIM tools were used more frequently than their offline counterparts.

We also asked general statistics about our users, including their age, education level, and the average time they spent working on their computers. We discovered that our average user's age was higher than we expected. Additionally, a higher proportion of highly educated (Masters/ PhD) individuals used List-it. Since List-it adoption has spread almost entirely through word of mouth, starting with a PhD student and a professor, this older and more educated collection of users makes sense.

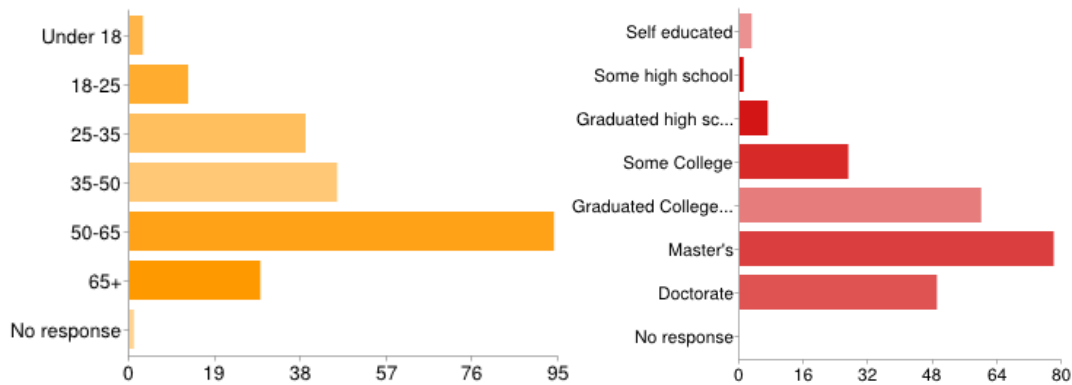


Figure 3. (above left): Age distribution of List-it users from survey responses.

Figure 4. (above right): Education distribution of List-it users from survey responses.

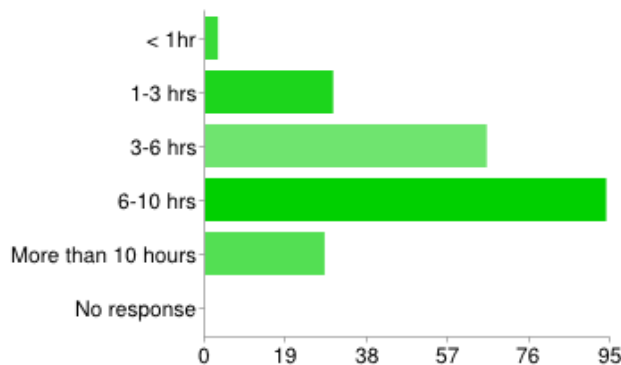


Figure 5. (above left): Average reported time List-it users spend in front of a computer each day from survey responses. This shows that List-it users tend to be heavy computer users in general.

3.4 How and Why Individuals use List-it

We hypothesized that List-it's users would store a wide variety of notes both at the level of the individual user and across all users. One potential reason for this is that users would use List-it to keep track of any or all information scraps that didn't fit in their existing PIM tools, and that these scraps would be necessarily different from the main categories of information that had particular tools for each user. From our survey we discovered that List-it's users were indeed keeping a large number of different types of notes.

One survey question focused on the different types of notes users keep in List-it. Looking at the combination of "All" and "Lots" responses for how much each category was kept in List-it, the top six categories in order are: Look up some day, Ideas & Brainstorming, Immediate To-Dos, Long-Term ToDos, pasted from the web, and web bookmarks.

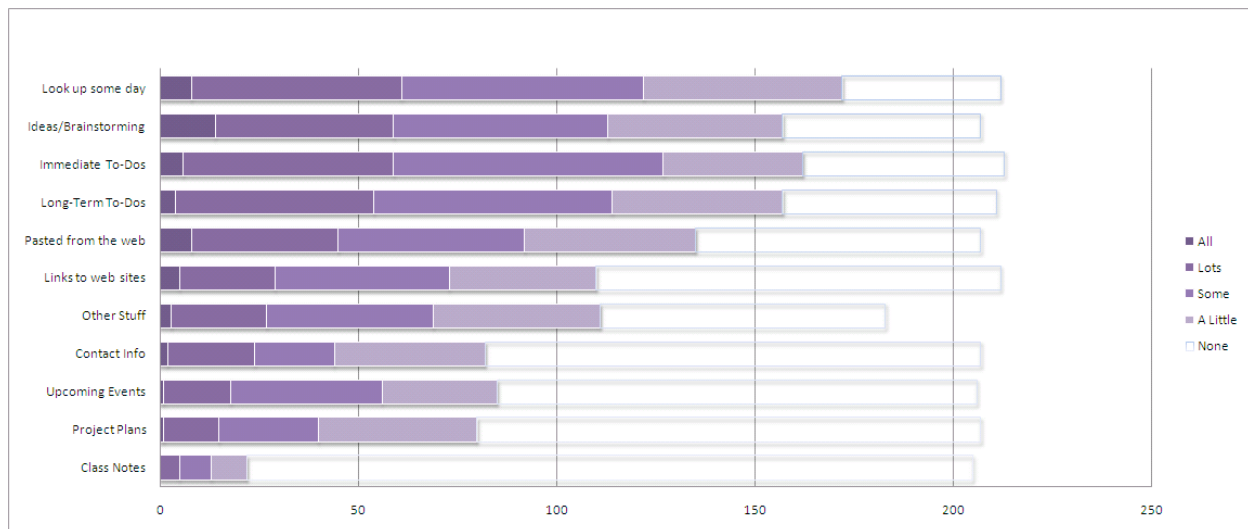


Figure 6. Number of survey responses (x-axis) for the frequency of the types of notes kept in List-it.

From the survey we also asked how often users both write things down and look things up in List-it. Combining the responses shows for each frequency of writing things down in List-it, the distribution of responses for the frequency of looking things up in List-it. The graphs show that the largest group of users both look things up and write things down about once a week. One interesting result is that there are a sizable population of users who report writing things down between once a week and a few times a day who also report never or almost never looking things up in List-it.

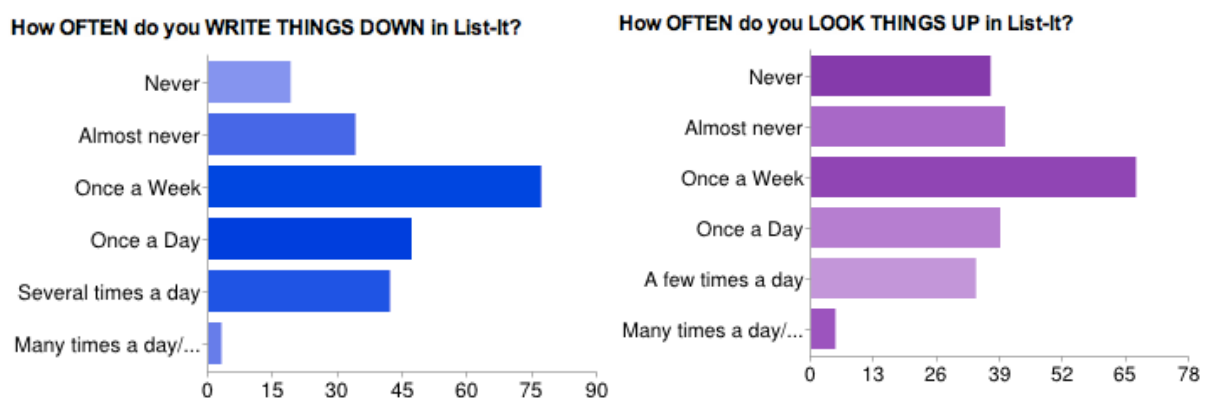


Figure 7 (Above left). The distribution of survey responses for how often users wrote things down in List-it.

Figure 8. (Above Right). The distribution of survey responses for how often users looked things up in List-it.

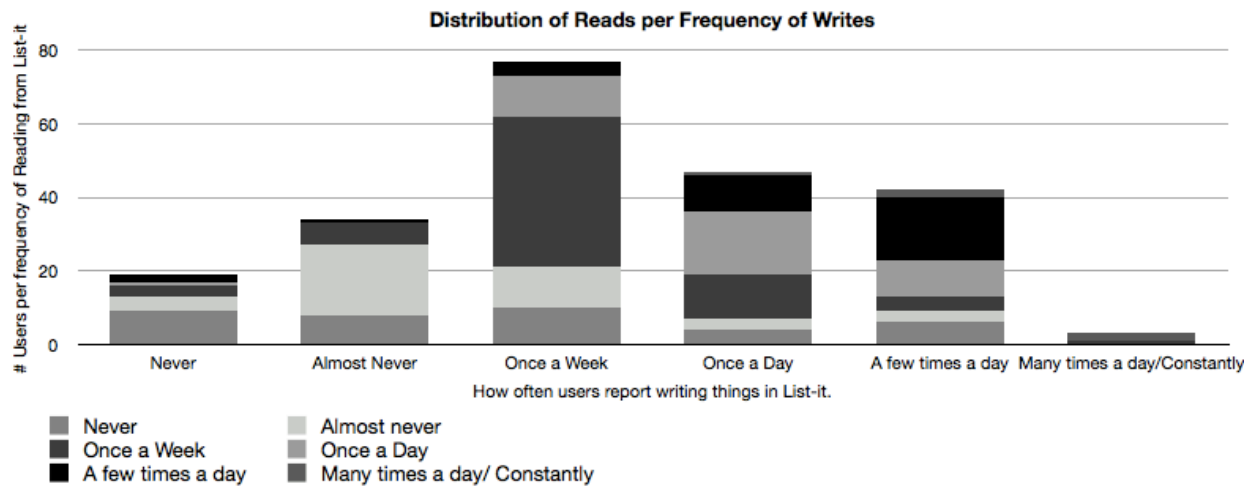


Figure 9. Distribution of survey responses for how often users look things up in List-it by how often they write things down in List-it.

Comparing survey responses to our user's activity logs (discussed more in the next chapter) of how frequently users opened the sidebar to write something down, we found that in general users opened the sidebar without creating or editing a note. The following two figures shows the number of times a sidebar was open for a particular range of durations. The darkened portion of the bar represents the number of opened sidebars that had at least one note created or edited.

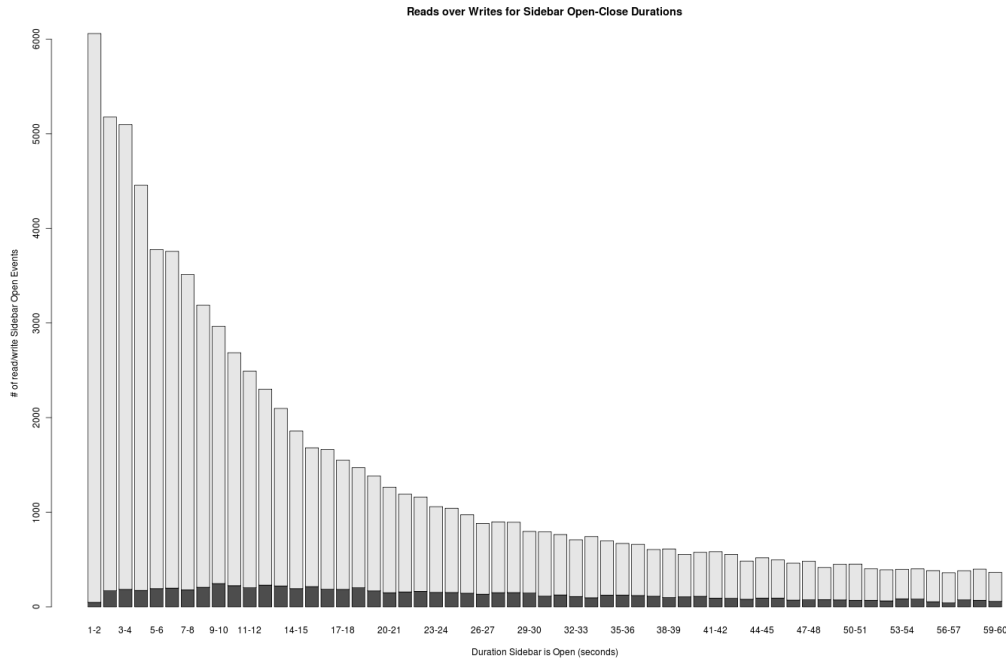


Figure 10. For each 2-second bucket of the amount of time a user opened the List-it sidebar, the darker bar represents the proportion of sidebar open events that included a note creation or deletion. The lighter bar represents the number of sidebars opened for each range in duration for which no writing activity was performed.

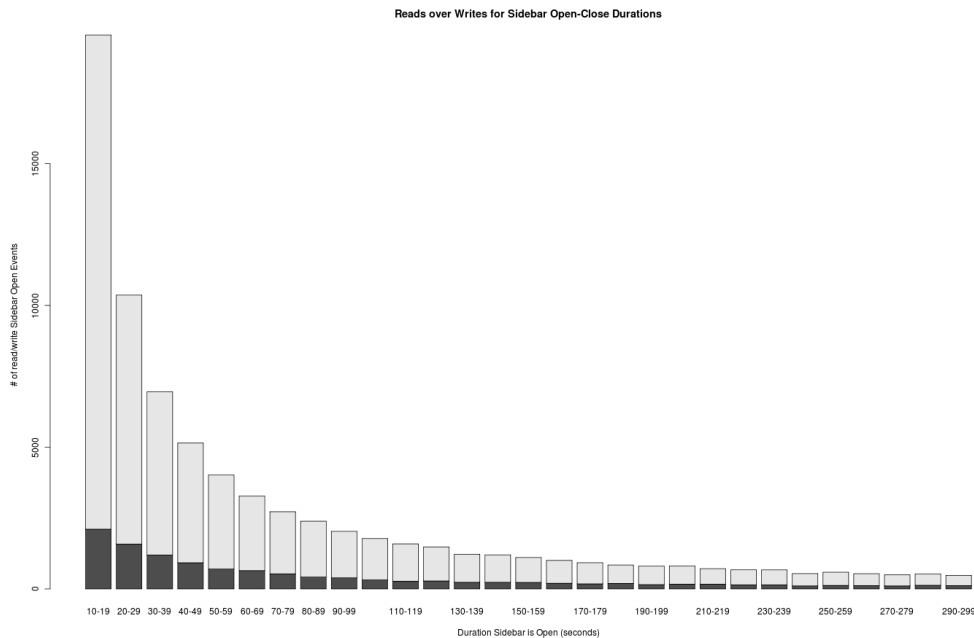


Figure 11. For each 10-second bucket of the amount of time a user opened the List-it sidebar, the darker bar represents the proportion of sidebar open events that included a note creation, edit, or deletion. The lighter bar represents the number of sidebars opened for each range in duration for which no writing activity was performed.

The above two graphs show that users generally open the List-it sidebar for relatively short amounts of time. The percent of opened sidebars that have a note created or edited is fairly low. This indicates that users are doing something else shortly after opening their List-it sidebar, such as reading their notes.

3.5 Tracking User Activity

List-it is instrumented with activity logging to record in fine detail the interactions of individual users with the tool. Roughly 10% of our 20,000 users have joined our research study, letting us analyze the activity logs that reveal how they use List-it. One pair of logs describes the creation of new notes, both the start event just before a note is typed in and the end event of creating the new note. Another pair of logs describes editing and saving changes to existing notes. Note deletion along with sidebar opening and closing events are also logged. Search logs describe not just what terms were used for the search, but which notes matched the search at that time. Through the analysis of these activity logs, we have gained a better understanding of how List-it users keep track of their information scraps.

4. List-it User Activity Analysis

An analysis of the activity logs that describes in fine detail a user's interactions with List-it over their lifetime of using List-it has revealed insights into the different interaction styles of types of users and the categories of information that are stored in List-it.

4.1 Types of Users

Through analyzing how users create, edit, and delete their notes over a two year timeline, four basic types of interaction styles were discovered. While many users could be clearly described by only one of these types, there were also several users who exhibited some combination of the four styles of interaction.

1. **Packrats** take lots of notes and rarely delete their notes from the past. This causes their list of notes to grow continually while they bury the old notes below the new.
2. **Minimalists** take lots of notes, but unlike packrats, are constantly deleting their older notes, keeping their list small and manageable.
3. **Sweepers** collect piles of notes like packrats, but then clean up in large sweeps by deleting many notes at a time.
4. **Revisers** keep fewer notes which they repeatedly edit.

The differences between the four types of users can best be described with the following note lifeline graphs. The x and y axes show approximately two years of time, with the origin being two years ago and the top right corner being the time the graph was created. Each grid line represents one week. Dots on the diagonal represent notes being created at a point in time. Above the dots are lifelines describing what happens to the note at later points in time. A vertical line above a green dot means the note created at that time has not been deleted, while an X icon at the top of a black line indicates that the note was deleted at that point in time marked by the X icon on the y-axis. Triangles on a line represent an edit made to the note it is vertically above.

The main benefit of this type of graph is that it clearly shows which notes from the past are affected by events that occur near each other at a time in the future.

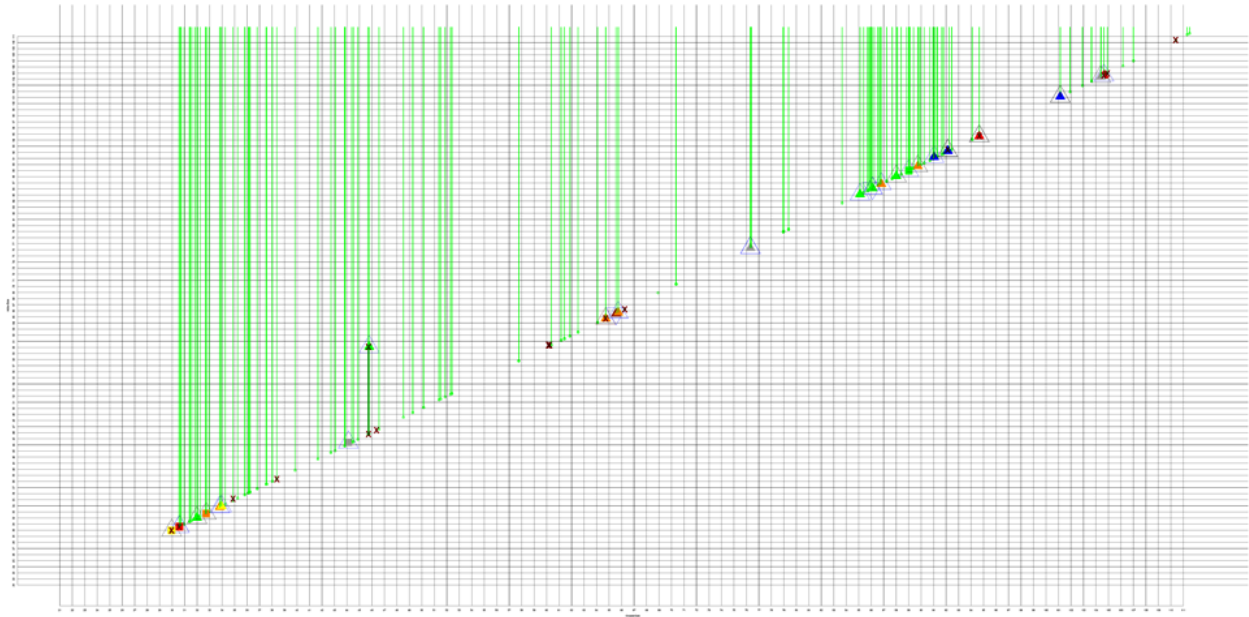


Figure 12. A typical Packrat, the long vertical lines represent the many notes that are never deleted by this user.

The above graph is representative of the *packrat* interaction style. Packrats create many new notes and avoid deleting their old notes, resulting in the many long vertical lines above the diagonal note-creation line. This causes packrats to grow increasingly large piles of notes as they use List-it. As a packrat adds new notes to their list, older notes are pushed down and eventually out of view. To view the older notes, a packrat can either search or scroll. Some packrats keep hundreds of notes in List-it.

The next interaction style, a *minimalist*, is shown in the following graph. Note how unlike the packrat, a typical minimalist deletes their old notes, as you can see by the lack of tall note lifelines above the diagonal note-creation line.

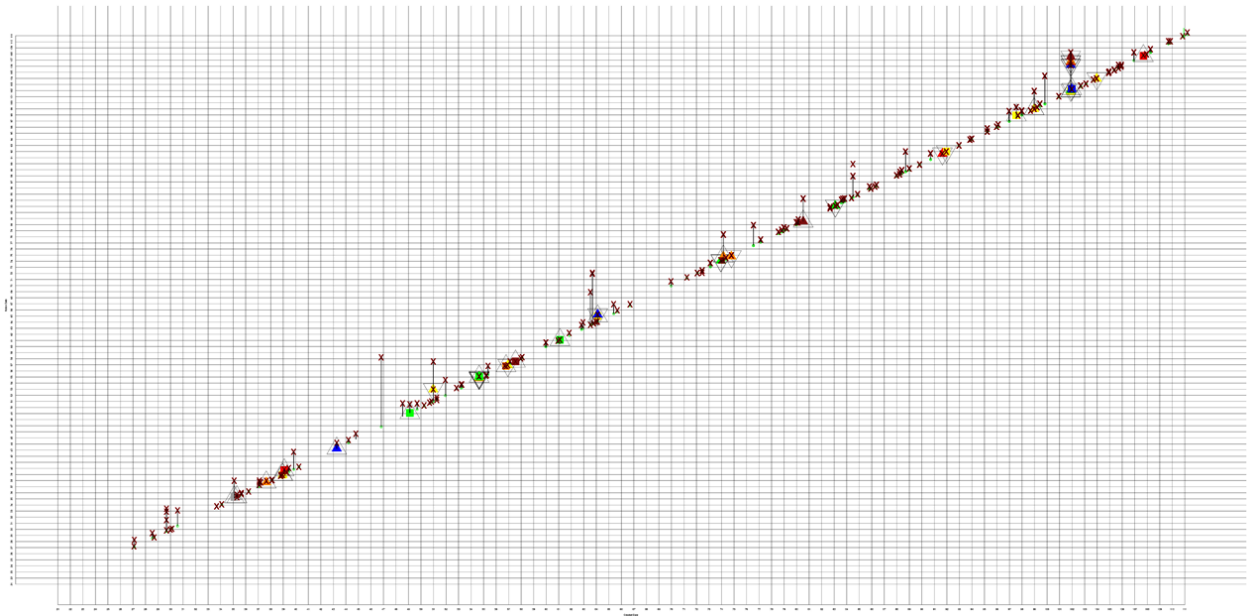


Figure 13. A typical Minimalist, the many X marks close to the diagonal represent notes being deleted soon after their creation.

A typical *minimalist* is represented by the above graph. Unlike packrats, minimalists are constantly deleting their old notes. This means they often have relatively few notes compared to the packrat. With only a handful of notes, a minimalist has easy access to all of their notes, since they don't have to scroll or search through 50 notes to find the one they want.

The next interaction style, that of the *sweeper*, is shown in the following graph. Note how similar the graph is to the packrat. The major difference is the rows of deletions, when at a particular time the sweeper decided they no longer wanted dozens of old notes and went through their list deleting many of them.

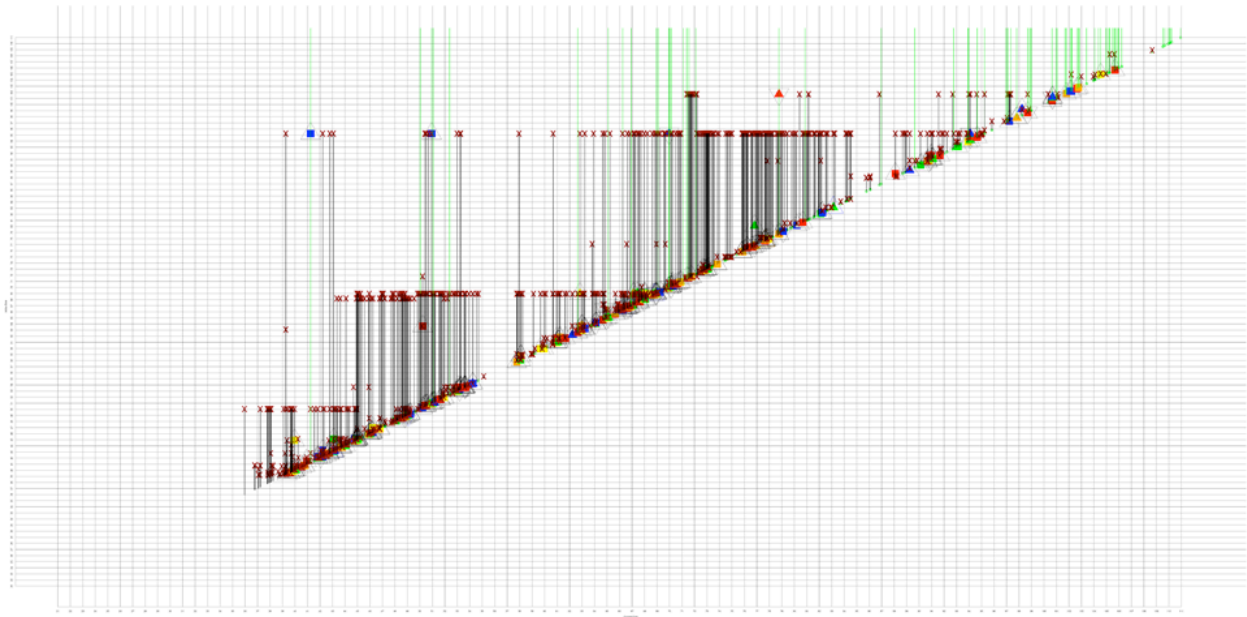


Figure 14. A typical Sweeper, the rows of X marks show when this sweeper deleted many dozens of notes that were created in the past.

There is a continuum of interaction styles that users fall in between pure packrat and pure minimalist. The sweeper is a special combination of the packrat and the minimalist in that instead of deleting recently created notes incrementally and keeping others forever, the sweeper keeps most of their recent notes and later goes on massive purges of all their notes. The above graph shows horizontal rows of red X marks where at that point in time on the Y axis, the user deleted most of their notes from the past several months.

The next interaction style, that of the *reviser*, is shown in the following graph. Note how sparse the graph is with few new notes being created on the diagonal line. The majority of a reviser's actions are edits made to notes, which are represented as triangles on the lifelines above the individual note's creation dots. Revisers rarely create new notes compared to the other interaction styles, preferring instead to constantly revise their existing notes. For revisers, List-it is more of a scratchpad with a few notes that are constantly being updated. Closely examining the notes of revisers participating in the List-it study revealed that the primary type of heavily edited note kept by revisers contained lists of things.

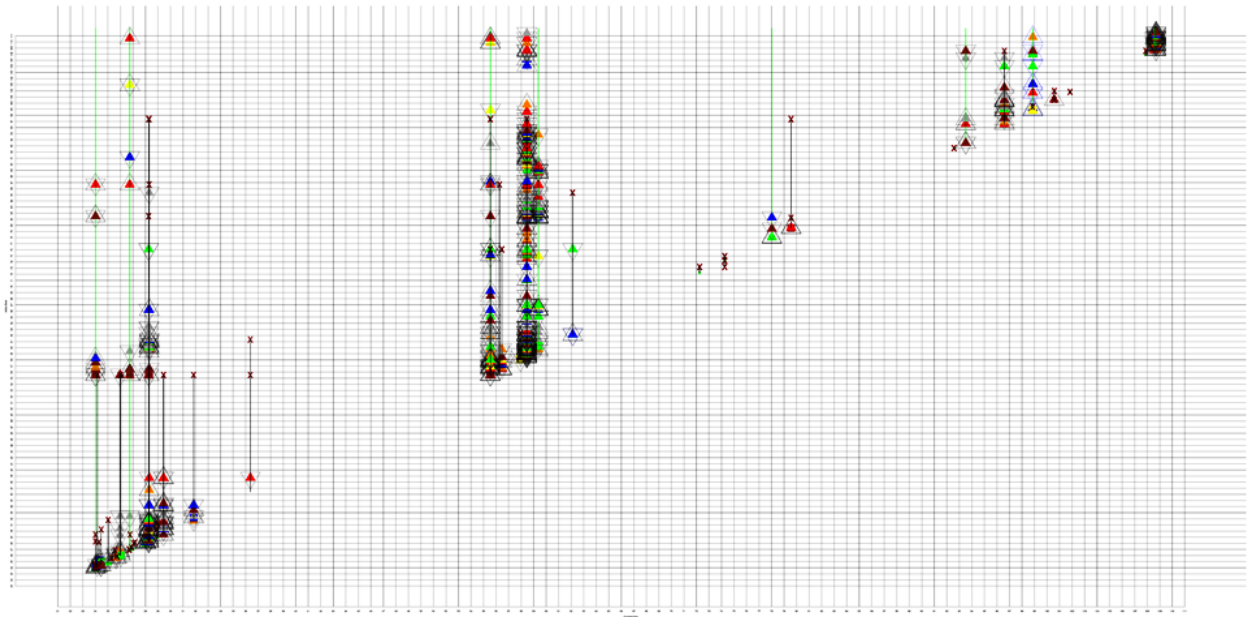


Figure 15. A typical Reviser, the many triangles on the lifelines above the notes show that this user frequently edited many of their notes.

Examining three pure revisers who showed no signs of being a packrat, minimalist, or sweeper led to the following observations. One reviser kept several heavily edited notes with details of their progress through quests in a video game. Another reviser kept a note of a list of things to print out that was revised 41 times, a note of a list of people to email that was revised 38 times, and a note containing a list of tasks that was revised 38 times. A third revisor kept a note with a list of songs that was revised 54 times, a note with a list of grocery items that was revised 11 times, and a note with list of things to do that was revised 13 times. These three pure revisers generally kept notes containing lists of things that were constantly changing.

Not all users could be cleanly categorized by one of the interaction styles above. Many users exhibited a primary category and a secondary category. Fewer users exhibited three categories, and very few users were hybrids of all four categories. The following graph shows one such user who keeps many notes without deleting them like packrats but also deletes many notes fairly close to their creation time. Additionally, this hybrid user shows a few sweeps where they primarily delete their old notes, in addition to having several of their notes which were never deleted receiving many edits.

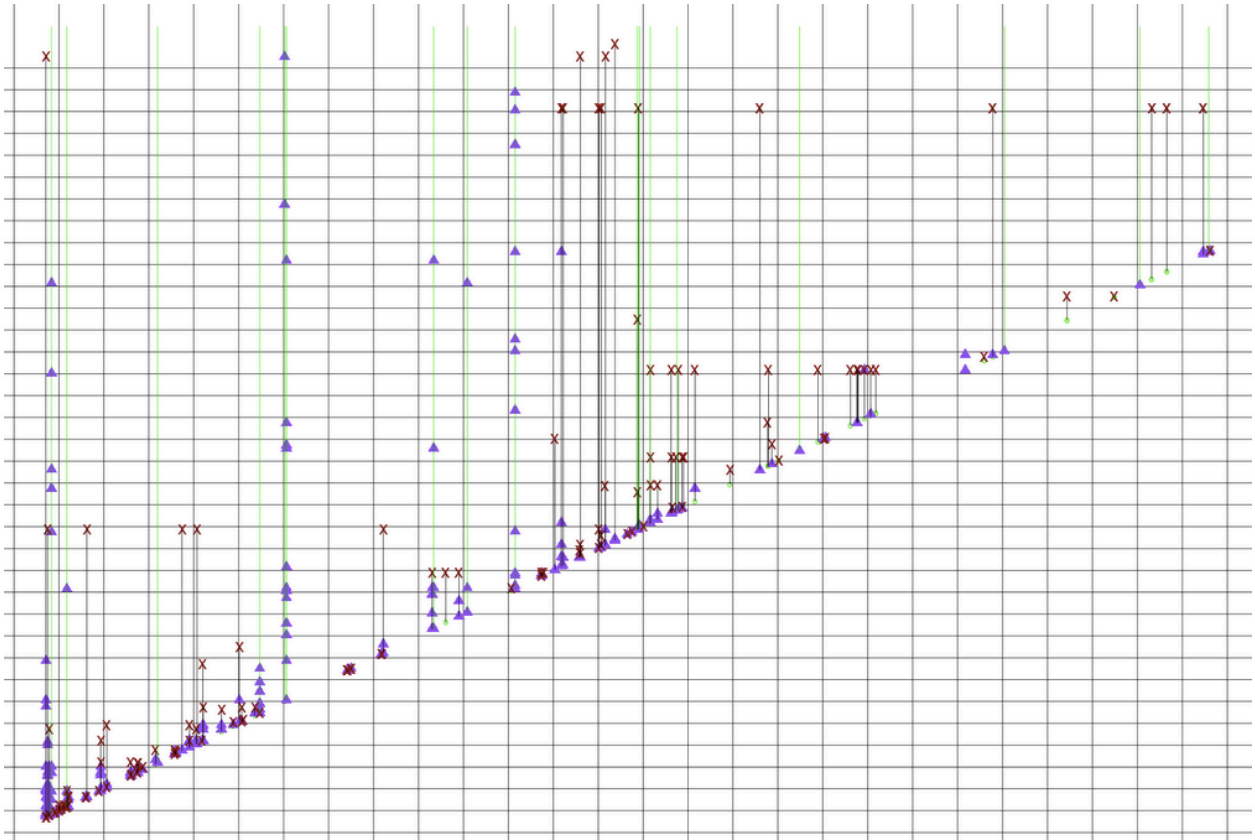


Figure 16. This part of a user's interaction style shows a rare hybrid of the previously discussed four types of interaction styles. Very few users exhibited a combination of all four interaction styles, with most users exhibiting a primary and possibly a secondary interaction style.

4.2 Distinguishing Interaction Styles

Our statistical analysis of the differences between users' interaction styles from our CHI paper [7] show significant differences in a set of metrics that describe the different ways users interact with List-it.

Packrats were found to have a statistically significantly ($p < 0.0001$) larger average number of words per note than the average user's notes. Packrats created more notes each day they used List-it, along with deleting fewer notes per day of List-it usage. The average change in the packrat's number of notes was 1.06 new notes per day, between roughly 10 and 100 times more growth in note collection size than sweepers, revisers, or minimalist experienced per day. As expected, packrats were found to never delete 67% of their notes, compared to the average of

33% for everyone and 13% for minimalists. Packrats used the search feature an average of once every four days, which was significantly more than minimalists searching at an average of once every 10 days.

Minimalists were found to be actively deleting notes, preventing the growth of their note collection's length. Interestingly, we found minimalist's average collection size change to be slightly negative at -.005 notes per day. This is because List-it users start with an introductory set of about a dozen notes, so minimalists shrink their note collection over time by deleting these notes and then staying below the number of initial notes they started with. Interestingly, minimalists were found to use the tool more than the other interaction styles, using the tool an average of 1 out of every 3 days, compared to 1 of 5 days for revisers and 1 of 4 days for packrats.

Sweepers have similar statistical metrics as minimalists, except that their notes tend to live longer and are more likely to be deleted in a large removal of many notes. Sweepers were found to keep about twice as many notes as minimalists, but half as many as revisers, and only one-third as many as packrats.

Revisers were the most unexpected category to come from the analysis. With the fewest note creations per day of using List-it, at one add per three days of use, compared to packrats creating almost two notes per day of use, revisers also experienced very little collection size growth similar to sweepers and minimalists. Revisers made up for their few note additions by editing an average of 1.48 notes per day, compared to .26 per day for minimalists and .19 per day for packrats.

4.3 Categorization of Interaction Style

With these statistical descriptions and general descriptions of the behavior of packrats, minimalists, sweepers, and revisers, it should be possible to automatically classify List-it users into a category that best describes their usage patterns. However, a particular user may have more than one of the four interaction styles. Most users exhibited a primary and sometimes a secondary interaction style. Very few of the users we classified were observed to very weakly show all four styles. In these few cases, users showed none of the interaction styles very

strongly. In the cases where a user exhibited several interaction styles, usually only one or two styles dominated, with the remaining observable styles only accounting for a small minority of the user's interactions. The interaction styles that dominate for a particular user will be described as a major interaction style, while lesser prominent styles will be described as minor interaction styles.

Packrats can be identified by their tendency to not delete a significant percentage or number of their notes. A high percentage of undeleted notes could indicate the packrat style is a major interaction style for a user. A user with a lower percentage of notes that have never been deleted (ie. not a primary packrat) combined with a relatively high number of notes that have never been deleted spread out through the user's history (showing packrat tendencies for a significant number of their notes) could indicate that they are a minor or weak packrat. In this case, the user is deleting a large percentage of their notes, so they are likely to have their primary style identified as a minimalist or a sweeper.

On the opposite end of the spectrum, minimalists can be identified by their tendency to delete a significant percentage or number of their notes fairly recently after their creation. Users that delete a significant percentage of their notes fairly soon after their note's creation are likely to have minimalist be their primary interaction style. Users with a smaller percentage of their notes deleted but with significant numbers of notes that are deleted fairly soon after their notes are created are likely to have minimalist as a secondary style.

Sweepers pile notes like packrats, and delete significant portions of their notes like minimalists. Their distinguishing characteristic is that they delete their notes in bursts, unlike minimalists who continuously clean up their list by deleting notes fairly soon after creating them. Major sweepers can be identified by the deletion of a significant percentage of their notes in bursts. Minor sweepers can be identified by the deletion of a significant number, but not percentage, of their notes from bursts of deletes.

Revisers are quite different from the note creation and deletion heavy styles mentioned above. Revisers make and delete few notes, but are constantly editing their old notes. Major revisers can be identified by a significant percentage of their notes being edited several times. Minor revisers may have a significant number, but not percentage, of their notes that are heavily

edited. Upon exploring three pure revisers, users who exhibit none of the other interaction styles, it was discovered that the primary type of heavily edited note consisted of a list of things. Examples include lists of songs, tasks, grocery items, sheet music to print, todos, and progress and requirements in video game quests.

Most users display a primary and sometimes a secondary interaction style. A few users show three of the interaction styles, with very few users showing a weak version of all four styles. Since a user cannot both delete and never delete a majority of their notes, users can't be primary packrats and primary minimalists. Users who display weak primaries may either use multiple interaction styles or lack a well defined interaction style.

5. List-it Note Analysis

The previous chapter discussed findings related to the types of user behaviors discovered through analyzing the fine-grained activity records of List-it's study participants. This section describes the types and roles of notes kept by the different users of List-it.

5.1 Note Contents

To begin to understand the role that notes played for List-it users, we began by examining what users placed in the contents of their notes. Four expert coders began this process by finding categories to describe a randomly chosen set of 540 notes which were created by volunteer participants of the List-it study. Each note was assigned one category. These initial categories were collected and reduced to 48 categories which are shown in the following figure.

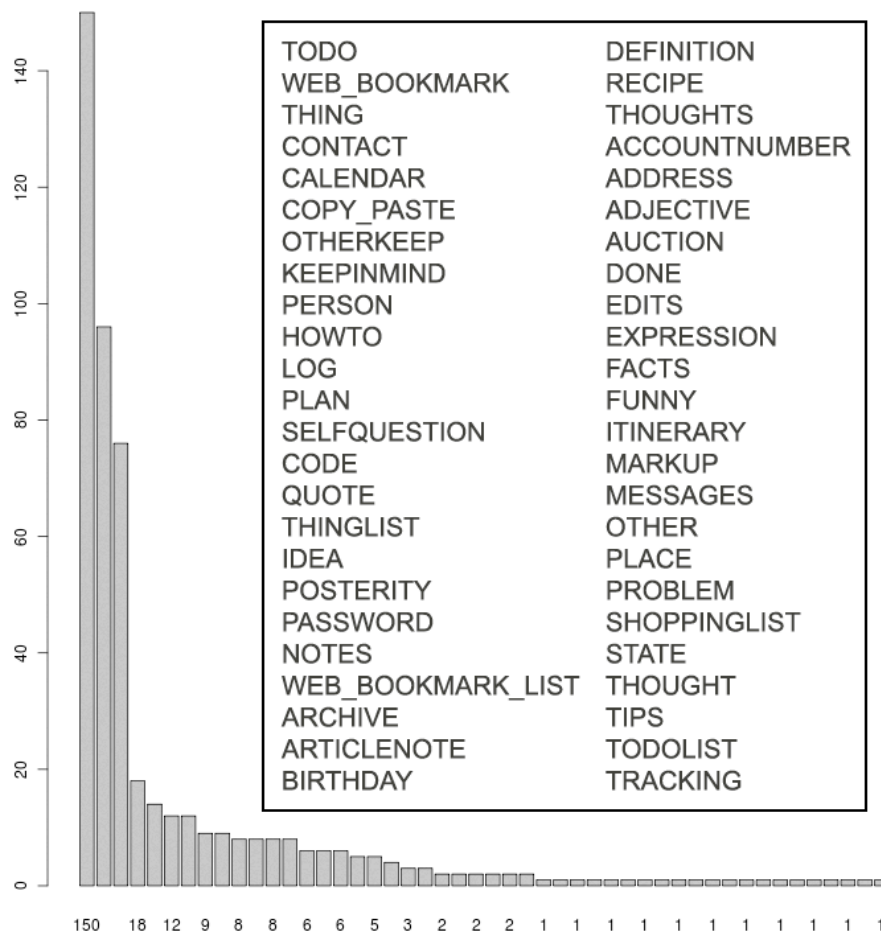


Figure 17. Counts (y-axis) of the final 48 categories (x-axis) of notes from four expert coders' classification of 540 notes from List-it's study participants.

The above figure shows an exponentially distributed set of categories. Todo, bookmark, and generic things were the top three categories, followed by contacts, calendar, and copy_paste. There are many types of notes that make up only a small part of the total notes examined. This lends evidence toward our hypothesis that List-it is suitable for capturing the pieces of information scraps that fall between the cracks of more structured PIM tools.

5.2 Note Roles

With roughly 10% of our 20,000 users participating in our research study, we were able to look through many notes from many users to better understand and categorize the different types of notes that our users kept in List-it. Through reading notes and trying to understand the intent of the user for keeping a note, five major roles for notes were uncovered.

1. Memory Trigger
2. Reference
3. External Cognition
4. Journaling
5. Posterity

These categories were used in the MIT Information Scraps Corpus (MISC) (described in the next section) to allow List-it users to categorize their own notes. From the 2,327 categorized and redacted notes that our users submitted to the MISC, we can see clear examples of notes our users describe as belonging to one of the above five categories. Several examples of notes for each note role are provided from the MISC. The examples are of the redacted notes, in which redacted phrases have had lower/upper-case letters turned to x/X, numbers to ‘9’, and symbols to ‘*’.

Memory triggers generally served as a mental bookmark to remind the user about a more detailed memory from the generally short note. Todos, tasks, reminders, names of things, and upcoming events are types of notes we classified under the memory trigger category. Listed below are examples of notes that were categorized as memory triggers by their owners using the MISC tool.

- lookup new health bill, 19/1
- write a few scenes
- (todo) Index cards for work.
- TAXES

In a sample of notes categorized by members of our research group [7], memory triggers were statistically shorter than the other note types, averaging 55.56 characters, 8.53 words, and 1.49 lines per note. Memory triggers were almost twice as likely to be deleted (70%) as the other types of notes.

Reference notes generally served as a repository of knowledge to be used in completing a future task. Passwords, codes, urls, recipes, instructions, and contact information were classified under the reference category. The following is a sample from the notes categorized as references by their owners in the MISC.

- 2010 taxes filed 2/3/10 (accepted date); deposit to checking: 2/19
- Copier code: 999999
- Source for rolled felt furring strips: whitecapdirect.com
Source for PhoenixBoard strips: eepdistributing.com
- http://www.metacafe.com/watch/593360/flower_coloring_experiment/

External cognition notes are used to augment the working memory of the user. Plans, outlines, agendas, drafts, and notes to help with problem solving were classified under external cognition. To avoid confusing users in the MISC, we called the external cognition category “Temporary / Scratch Pad”. The following notes were described as belonging to this category from the MISC.

- Oyster sauce #shop
- <http://www.schooltube.com/video/4b18f3134568895b378d/Theories-of-Learning>
- In addition, write a brief statement describing what each member of your team contributed and how your team worked together.
- THANKS for all your work @ #ISTE10 - because of you it's #SS (Saturated Saturday). Almost as good as being there?!

Journalling describes notes that record things that happened in the past. Diaries and log entries for exercise, food consumption, project status, general status, and meeting notes were classified under the journalling category. For the MISC, we called this category “Diary / Journal”. Here are some notes labeled under this category from the MISC.

- ^^^

6/99 went biking with ^^; also went a little shopping; bought cute dragonfly straw thing!

- had 2 take 4th oxy at 7:15---gruelling, burning MS pain plus sciatica...think weather related

- Bicycle commute times:

9-14-09 37:30

10-5-09 36:26

10-14-09 37:50 (cold)

6-23-10 36:00 (new speedo, doesn't count stopped time)

7-2-10 36:43 (new), 38:28 (old)

- MSOffice2007/Upgrades: boy, I may have really screwed myself by issuing this challenge (upgrading to office 2007). just as I was mastering certain components & programs in 2003...

Posterity notes are similar to reference notes, but with less of an expectation that they will be useful for completing a future task. Jokes, quotes, poetry, and song lyrics are examples of the types of notes classified under the posterity category. Posterity notes were found to be deleted least; only 30% were deleted in our sample of analyzed notes. In the MISC tool, we labeled posterity as “Just for Keeps / Posterity”. The following notes provide a sample of the notes that were categorized by their owners as part of the Posterity category.

- @videos @links

<http://www.youtube.com/watch?v=Cbk980jV7Ao> -

"Validation" is a fable about the magic of free parking.

- <http://flowingdata.com/2011/05/30/time-lapse-of-the-night-sky-with-the-very-large-telescope/>

- Beauty, purposeless,
dawning on us from the world,
disinterested love.

- Feedly

<http://www.xwiki.com>

5.3 Manual Categorization of Notes by Role

A set of 600 notes were categorized by four expert coders to classify notes by their primary and secondary categories. Each coder scored notes from 1 to 5 for each of the five categories. Additionally, coders were asked to propose new categories that might describe notes that couldn't fit into one of the existing categories. No new categories were suggested. The coders worked on the same notes, and were found to have moderately strong agreement (Fleiss = 0.615) for categorizing the primary category of the notes. When secondary categories with a score of either 4 or 5 were included, there was stronger agreement (Fleiss = 0.85). A total of 473 (79%) of the 600 notes were marked with at least one category, with an average of 1.44 roles per note. From this exercise we learned that external observers could agree upon the role of another user's note with moderately strong agreement.

type N (%)	co-occurs	%	forms	%
mem trg: 303 (66%)	reference	29	todo	42
	ext cog	12	noun	15
	pers arch	8	web bkmk	7
	journal	8	contact	3
ref: 215 (47%)	mem trg	41	web bkmk	41
	ext cog	10	noun	7.5
	pers arch	12	contact	6.9
	journal	10	copypaste	3.7
ext cog: 59 (12%)	mem trg	65	todo	18
	ref	39	question	8
	pers arch	20	draft	4
	journal	19	idea	3
pers arch: 61 (13%)	mem trg	38	copypaste	32
	ref	43	idea	14
	ext cog	11	quote	9
	journal	13	joke	8
journal: 43 (9%)	mem trgg	51	todo	37
	ref	51	mtg mins	1
	ext cog	26	noun	1
	pers arch	18	plan	1

Figure 18. For each primary note role, the distribution of roles that co-occur and the distribution of note forms for that role is shown. Each note (n=473) may be labeled as several roles, so percentages do not total 100%.

5.4 Automatic Categorization of Notes by Role

Machine learning applications of text classification have affected several areas of our lives. Automated spam detection keeps our inboxes free of junk mail. Sentiment analysis of movie reviews at IMBD help us find better movies to watch. Analysis of product reviews at Amazon help us find the best comments about a product. Text classification at Yelp helps us find the best places to eat by showing us the best comments about restaurants we might be interested in visiting. Why do structured PIM tools require us to go through menus and dialogs to specify what we want out of a note? How can a micro-note tool that lacks the structure and user provided directions for what to do with a note provide similar functionality?

A micro-note tool that can automatically recognize different types of notes can better assist a user's ability to keep track of their notes. Consider a note serving as a memory trigger. A user takes this note so they will be reminded about something in the future. There are several problems that prevent List-it from showing this note to the user at the right time in the future. The first issue is that List-it doesn't know that a user wants to see a particular note in the future. If List-it could figure out which notes a user was more likely to want to see in the future, the next issue to overcome would be figuring out when to show each selected note to the user. One potential solution is to randomly show a small sample of notes from the notes a user wants to see. This could be done on a daily basis, such that the user would be constantly reminded about the notes they wish to remember. So how do we identify the notes a user wants to be reminded about at some later point in time?

In an attempt to identify notes a user would like to be reminded about in the future, a machine learning approach was used. Of the note roles described at the beginning of this chapter, the "Memory Trigger" note role best fit the requirement for finding notes a user wants to be reminded about.

A maximum entropy model was selected for this note role categorization task, as this model is suggested for text classification in Using Maximum Entropy for Text Classification [8]. The maximum entropy model was selected for its ability to easily utilize features of the text of each note, along with the possibility of using features about the context of a note, such as when or where it was created. Another potentially useful aspect of the maximum entropy model is its

ability to provide a probability that its classification is correct for each note, as an application of this model in a PIM tool would likely be able to use the probability that a note is correctly classified to its advantage when selecting memory trigger notes to show a user.

A total of 438 unique notes were categorized by four expert coders into the five note roles discussed at the beginning of this chapter. These notes were collected from List-it study participants who had agreed upon signing up with List-it to allow List-it's researchers to view and analyze their notes for research purposes. The five categories the raters used were: "memory triggers," "references," "external cognition," "logging/journaling/chronicling," and "emotional/personal archive." Raters gave a note a primary label, in addition to assigning strengths to possible secondary labels that might describe a note. Of the 438 notes, 248 (56.7%) were given the "memory trigger" primary label, the other 189 notes were given one of the other note roles for a primary label.

The maximum entropy model used a wide range of features to classify notes as "memory trigger" or "not memory trigger". Part of speech (POS) information was used to distinguish notes that contained each type of a POS, such as whether a note contains a verb or a noun. In addition, positional POS was used, such as to identify whether a note's first word was a verb, or whether a verb was found within the first three words. Due to the lack of grammar or sentence structure exhibited by many notes in List-it, and particularly for the generally shorter memory triggers (see figure below), many words were mis-categorized when determining their part of speech. The largest problem was notes that began with verbs like "Email ..." being classified as starting with a noun. Unlike most text categorization tasks, stop words (such as: a, an, the, but) were included in the feature set because memory triggers were observed to be so short and concise that they often left out these types of words. In addition to part of speech unigrams, part of speech bigrams and trigrams were also used. Other features examined the counts of numbers, symbols, and white space features. For example, punctuation such as question marks and commas were mainly absent in memory triggers. Automated features were created to look at ranges of features, such as whether a note contains more than n lines of text or n period marks, for n from 1 to 10. The following is a selection of some of the features used:

- First word is noun? First word is verb? First word is stop word? Second word is verb?

- Note contains (url, verbs, determiners, adjectives, adverbs).
- Note contains (day of week, stop words, todo terms).
- Note contains punctuation (period, comma, question mark, semi-colon, colon, and symbol, hashmark, dash).

Note Attributes	Memory Trigger (average/std. dev.)	Not Memory Trigger (average / std.dev.)
# Characters	43.7 / 62	127 / 241
# Words	8.3 / 12	21.5 / 49
# Lines	1.5 / 2.2	2.5 / 3.7
# Punctuation	1.0 / 2.0	4.7 / 7.9

Figure 19. Memory triggers were shorter than their complement in number of characters, words, and lines. Memory triggers also contained fewer punctuation.

The maximum entropy classifier was trained on a set of 80 tokens from each category of notes: “memory trigger” and “not memory trigger”. The test set contained 100 notes from each category. A total of 20 models were trained and tested (taking only a few minutes) simultaneously on different sets of training and test notes, with their performance scores averaged together to provide more robust accuracy statistics.

The maximum entropy classifier was successful at distinguishing memory triggers from the remaining notes. The model with the best performance includes the following features: part-of-speech (POS) unigram, bigram, and trigrams, a combination of number, symbol, and whitespace character unigrams, and unigrams for the first word of each newline, in addition to the features mentioned above. The best feature selection showed a precision of 85.4%, a recall of 87.8%, a true negative rate of 83.4%, a specificity of 80.3%, and an F-1 measure of 86.6%. The high recall means the algorithm can recognize 87.8% of a user’s memory triggers. The high precision means that of the notes categorized as memory triggers, 85.4% are actually memory triggers. For comparison, if the algorithm randomly assigned one of the two categories to a note with one-half probability for either category, on average only 50% of the memory triggers would be labeled correctly (50% recall). For precision, on average half the memory triggers and half the non-memory triggers could be categorized as memory triggers, 50% of the notes labeled memory trigger would actually be memory triggers (50% precision). The maximum entropy model does a much better job of identifying memory triggers than random chance.

This maximum entropy model could likely be improved by using a larger number of accurately categorized notes to train upon. The MIT Information Scraps Corpus (MISC) described in a previous section, is one source of categorized notes that meets both of these demands. Users know better than a third party what category their note is best described by. We also gain far more notes for much less work by collecting categorized notes from the MISC than we could if we hand categorized notes one at a time using expert raters.

Of the many things we write down, memory triggers are among the ones we wish would come back to remind us of critical information pieces at various times in our lives. The ability to recognize these notes is one important part of building better PIM tools. The successful ability of maximum entropy models to classify List-it users' personal notes under the memory trigger intent category suggests that it may also be possible to select features to distinguish other categories of intent for people's personal information scraps. If a PIM tool could reliably attain knowledge of how a user intended for a note to be useful, then user interfaces could be designed to increase the benefit to the user of the PIM tool from smart utilizations of this knowledge.

5.5 MIT Information Scraps Corpus

The MIT Information Scraps Corpus (MISC) was developed to provide an open and free corpus of user-redacted and user-categorized personal notes. The MISC, implemented as a website where List-it users can view, redact, categorize, and submit their notes, has had a total of 2,327 categorized and redacted notes submitted as of May 10, 2012. The MISC was created with two primary goals:

1. To gather more authoritative note categorization data.
2. To provide access to personal notes for researchers outside of our small research group.

The first goal, gathering more authoritative note categorization data, was motivated by the difficulty of categorizing notes that are not your own. Certain categories of notes, such as Memory Triggers, are much more difficult to categorize than other categories, such as References. Instead of using a team of categorizers to find agreement on a note's category, the MISC lets the note's owner categorize their notes. From the five categories we discussed in the previous section, we renamed several categories to be more understandable to users who

categorized their notes. We chose the following six categories (with the mapping to the categories described in the previous section in parenthesis), plus an optional other input box where the user could create new categories:

1. To-Do / Reminder (Memory Trigger)
2. Ideas / Thoughts (Memory Trigger/External Cognition)
3. Reference (Reference)
4. Diary / Journal (Journalling)
5. Temporary / Scratch Pad (External Cognition)
6. Just for Keeps / Posterity (Posterity)

The second goal, providing broader access to personal notes, was motivated by the difficulty of running a longitudinal study of personal information practice. For a research group at another institution to study micro-note PIM practices, there are hurdles to overcome. First they would have to build a note taking application, which may take up to a year. Second, they would have to find a way to acquire users for the application. Third, they would have to get some of their users to volunteer to be study participants so their usage of the tool could be analyzed. Lastly, they would have to get enough study participants to be statistically useful in analyzing their usage behaviors. This process took our research group roughly 2 years, and each step was a potential failure point that could have caused the entire project to fail. The categorized and redacted notes data is available at <https://welist.it/MISC/view/>. By providing an free and open corpus of redacted and categorized personal notes, we are moving one step towards reducing the work required by other research groups to study people's personal information.

6. New Features for Unmet Needs

One goal of both our user surveys and our more intense graphical and statistical analysis of their activity logs was to find out what List-it was good and bad at in terms of helping users manage their information scraps.

From our user surveys, we heard from many users how they used List-it to keep information that they would have lost. Users keep notes in List-it because they want to be able to have the information available at a later time.

Users identified the problem of losing track of their notes if they don't keep them in List-it. For these notes, visibility is critically important for List-it to be able to remind the user about their note later. For minimalists and revisers this is likely not a problem, as both user types have few notes that are constantly being edited and/or deleted.

6.1 Buried and Forgotten Notes

“it would be cool if there would be a way to prioritize some points rather than having them all appear on a list.”

- A user's survey response.

Packrats who dabble in minimalist or reviser tendencies, keeping notes that they would normally forget about but want to keep track of for reading or editing at a later time, are more likely to experience the problem of “out-of-sight, out-of-mind” compared to the other interaction styles. In general, any user who wants to keep track of a note they will likely forget about but who is also constantly adding new notes to their list will likely experience this problem. The problem is that these notes become worthless if they are forgotten, because the user won't actively search for them if they don't remember they exist and have no external cues to remember their existence. Clearly the notes are not actually worthless, because the user took the effort to write them down. Our previous solution to this problem allowed users to reorder their notes. For these types of notes, users could drag them to the top of their list each time they get buried when enough new notes are added. The problem with this is for all notes that are relevant over a long time period, the user must continually reorder each of the important notes each time a

new note is added. Alternatively, the user could drag each newly created note below the important notes. Pinned notes makes this the default experience.

6.2 Feature: Pinned Notes

List-it previously provided a single list of notes, sorted in chronological order with the most recently created note on top. Users were, and still are, allowed to drag their notes to a new ordering. But there are some notes users may wish to always stay visible. The pinned notes feature lets users keep track of a few of their notes by separating the main list of notes into two stacked lists. The top list contains pinned notes in the same chronological ordering as the bottom list, which contains the normally created notes. All of the pinned notes are shown above the non-pinned notes. The pinned notes can be re-ordered just like the non-pinned notes, and notes can be dragged from one list to the other. Pinned notes repeatedly remind the user about their existence by being visible at the top of the list of notes each time the List-it sidebar is opened.

Every time a user opens the List-it sidebar, they see their pinned notes at the top of their list. As such, the user is reminded that they have important information every time they open the sidebar, making it harder to forget that they have important information and easier for them to find the important information when it is needed. By making these types of notes easy and useful to store, we expect users will find that they can keep these notes that they might otherwise have trusted to memory or written down elsewhere. This transition is similar to how people found they could keep information scraps in List-it that they previously didn't record elsewhere.

What types of notes fall into this category? Memory triggers are worthless if they fail to remind the user about something at some future point in time. Weekly progress reports such as recording hours worked become harder to use if you have to go digging through increasing piles of notes to update it each day. Task lists for the coming week or month also get buried by new notes, making them harder to find. One possible solution is creating a new saved search that finds the particular note a user wants to keep easily accessible. While this may reduce clutter at the top of the user's list of notes, it requires finding a search that matches the note they want without including too many other notes. Pinning notes helps the user remember and easily re-

access individual notes that will otherwise either be completely forgotten about or frequently reaccessed.

6.3 Scanning Note List Chaos with Minimized Notes

"I would describe my organization as "constantly scanning chaos". I have a pile of notes on list it and some paper notes that I keep scanning"

- A user's survey response.

Packrats and other users who keep lots of notes have another problem that other users with fewer notes don't experience. For a user with many notes, the majority of their notes are hidden out of view, hidden below the few visible notes at the top. There are two major ways to view these notes: search and scroll. Search is great at letting a user find notes they can remember exist. But not everyone can remember the existence of all their notes. This means a user who wants to look at previous notes must scroll through a long list of differently sized notes.

There are really two problems that are cleverly disguised to look like one problem. The first problem is that when a user with many notes opens the List-it sidebar, only the top few notes, some of which may be large, are visible. The second problem is that scrolling through unevenly sized notes slows down the process of identifying and examining individual notes.

For packrats and other users that have acquired lots of notes, the minimized note feature lets the user shrink all their note to only show the first line of text in each note. This feature addresses both of the problems described above. Retrieval is made easier as the user can quickly skim over about 20 one-line notes at a time instead of only a few or one large note which takes up all the list space on the screen. While the user can't immediately see the full text of any one note (except for notes that are 1 line long), they can click notes that catch their interest to easily expand the note to show its full text.

This feature has another benefit besides improved search efficiency. When a user with many notes opens the List-it sidebar, instead of seeing only the last 3-5 notes they created, they can easily see the first line of the last 20 minimized notes. This means that an increased number of recently created notes are visible to the user for a longer period after their creation, which may help the user remember about the note's existence if it isn't a pinned note.

6.4 Viewing Notes in Context

“the only thing that i would find relevant and needing to keep close are passwords ... basically, it has been my solution to a password manager”

- A user’s survey response.

Some users view List-it as a password manager, keeping passwords, account information, and other website-specific information in their list of notes. For these users, logging in to a website involves searching their list of notes for the note containing the site’s account information. Other users write snippets about web pages, to be compiled into a blog post later. Users previously needed to copy and paste the site’s url into the text of their note to keep a reference to which site they were reviewing. If they didn’t finish their note about a website, later they would have to retrieve the note when visiting the site again to finish their writeup.

This feature surfaces notes that were created at the currently focused website, such that when a user changes which website they are viewing, either by changing the focused window or tab, or navigating within a page, notes that were created at the currently viewed website are floated to the top of their list of normal notes. This helps the user view previous notes, some of which may be long buried, that were created at the same site they’re currently viewing and which may be relevant in some way to their current activities.

The benefit of this feature for password management is clear. As long as you jotted down a site’s login information while looking at the login screen of the site, when you visit the login page again, your previously recorded account information will be surfaced from the depths of your note list for easier re-access.

The pinned notes feature complements this feature nicely. The pinned list of notes stay on top of the notes that float to the top of the normal list section. This allows the user to maintain a set of notes that are constantly at their fingertips ready to meet their needs. Without a pinned list, users would likely be upset to lose the notes at the top of their list as floated notes are placed on top while they view different websites.

6.5 Context Retrieval for Notes and Automatic Bookmarking

“I find it's a better place to put URL's that I want to check out later than just adding them to my bookmarks (where they are often forgotten)”

- A user's survey response.

From our user survey of 225 List-it users, we identified bookmarks as the 6th most popular category of note stored in List-it, after the categories ‘look up some day’, ‘short & long-term todos’, ‘ideas/brainstorming’, and ‘pasted from the web’. For bookmarks, the previous version of List-it featured an icon that allows users to save a note with the url of the currently viewed website. If users use the minimized note feature described previously, the only visible text for a note with a bookmark is likely to be the url of the site. In the older Firefox add-on, notes transitioned between two views. One view used a textarea to allow editing of the text of the note. The other view displayed the note in a div so that links would be clickable. In the new application, these two views were merged into one view, so the user doesn't have to wait between clicking the note and the note changing from one view to the other. To visit a site from the new application, a user must highlight and copy the url and then mouse over to the address bar to paste the url before hitting enter to view the site.

Context retrieval is a useful feature for micro-note PIM tools. For bookmarks and other notes that discuss the website a user is viewing while creating the note, being able to open the website without having to store the url in the text of the note may be a useful feature. Some users report writing reviews of sites for their blog posts. For these notes, being able to get back to the site the note is about would likely be useful.

Automatic bookmarking associates each note with the website that was visible when the user created the note. This feature is implemented but is not visible to the user yet, as we are searching for a good way to allow the user to easily discover the bookmark without cluttering the List-it user interface. This feature is essentially the inverse of the notes that float feature, in that notes bring websites into view instead of websites bringing notes into view. An interesting side-effect of having both notes that float and this feature active is that opening a note's bookmark causes that note and other likely similar notes created at that page to float to the top, where the user can view not only the note they're editing now but also other notes made while viewing the same webpage.

6.6 Summary of Features

The main focus of these features is to make the most relevant notes more visible at the right times. Pinned notes make a user-selected subset of notes always visible by keeping the notes at the top of the list. Notes that float make a context-selected subset of notes visible based on the current and previous websites a user views. Minimized notes enable a user to both initially view (the first line of) more notes than they previously could, in addition to allowing easier scanning of the large collection of notes through scrolling. Automatic bookmarking allows users to describe the website in the text of the note without requiring the extra step of copying and pasting a url which may take up the entire first line (the entire viewable text in minimized mode) of the note.

7. New List-it Application

The chapter discusses the motivation behind creating a new List-it application that supports a Chrome Extension, Firefox Add-on, and website using a single codebase. The new application features a modified user interface to support the features discussed in Chapter 6.

7.1 Motivation for New Application

The current List-it application is a sidebar for Firefox that allows users to capture notes while simultaneously viewing websites in tabs within the browser. Our research group has received requests through emails and user surveys for List-it to become more available for users who use Google Chrome and other browsers. In considering the construction of an extension for Chrome, it was discovered that a sidebar like design, while not a native feature of the Chrome Extension API, could be effectively mimicked to provide a similar sidebar experience like the Firefox add-on.

The codebase supporting the Chrome Extension's unique architecture has been designed to work as normal website. This sharing of a single codebase across both a Chrome Extension and a website reduces the effort required to implement new features for List-it on all its applications. A Firefox add-on has also been created that loads the List-it web application in a browser sidebar. This results in three applications, a Chrome Extension, Firefox Add-on, and regular website all using the same shared codebase. This greatly reduces the amount of work to implement new features, and provides more choice to users who would like to use their favorite browser with List-it.

The new applications will also encourage more users to sign up who prefer other browsers, likely resulting in a larger number study participants. The research presented in the previous sections of this thesis was conducted using the data collected by our current List-it study participants who were using the older Firefox extension described in Chapter 3. As more study participants sign up to use List-it, our capability to better understand our users will increase.

7.2 List-it's New User Interface

The screenshot shown below shows the List-it application as a sidebar Chrome Extension. The List-it sidebar is shown on the left side of the browser window, and the browser window is automatically resized such that the sidebar fits on its left side without overlapping.

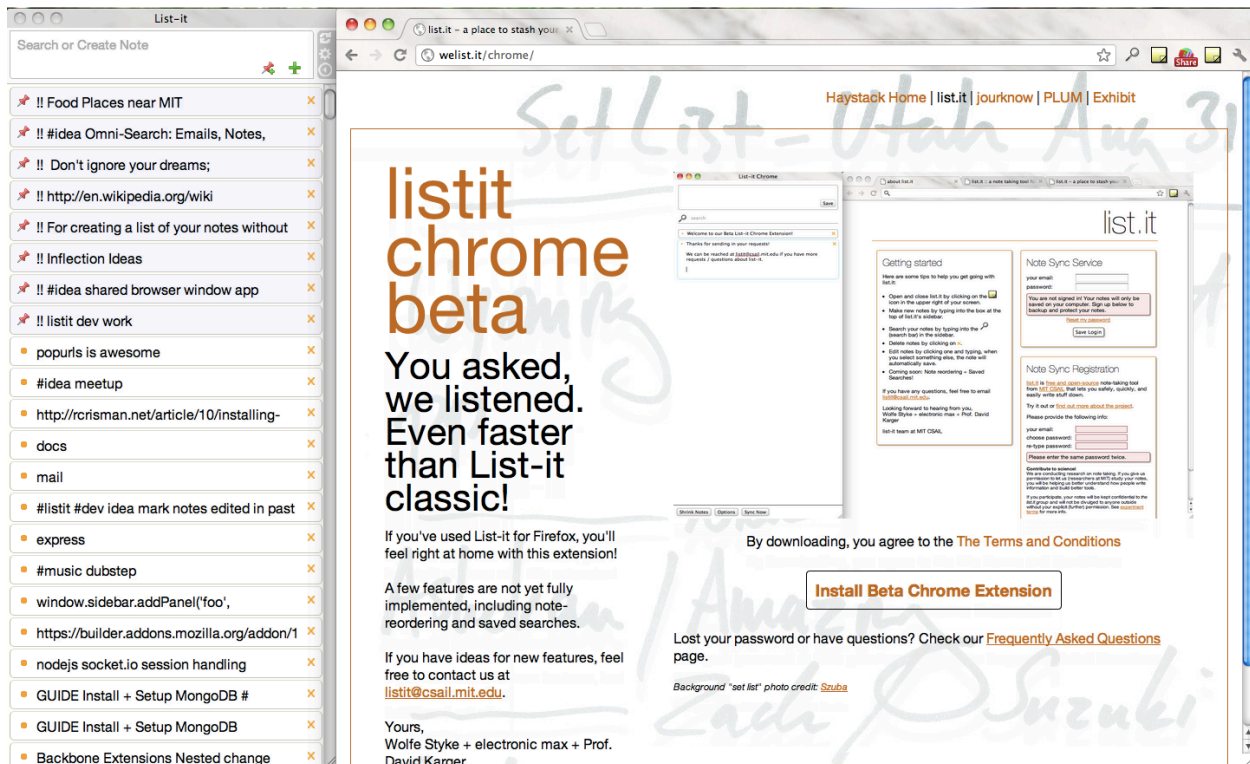


Figure 20. The List-it Chrome Extension sidebar is visible to the left of the browser. The browser automatically resizes to fit the sidebar on its left side when the user opens the sidebar by clicking the note icon or using the hotkey.

Clicking the yellow note icon to the right of the address bar, or using the hotkey command (default of “Ctrl-Shift-F”) opens the List-it sidebar and automatically resizes the browser window such that the sidebar fits on its left side without overlapping with the browser window or showing up off the monitor on the left side. At the top of the sidebar is the combined note creation and search box. The first line of the note creation box is used as a search bar. Users can type as much as they want into the top box before saving the note as either the regular or pinned note list. Pinned notes appear above normal notes, and help the user keep important notes at the tips of their fingers.

The screenshot below shows the List-it application as a Firefox Add-on. The sidebar is opened within the browser, unlike the Chrome Extension. The currently focused tab's page is automatically resized by the browser to make room for the sidebar.

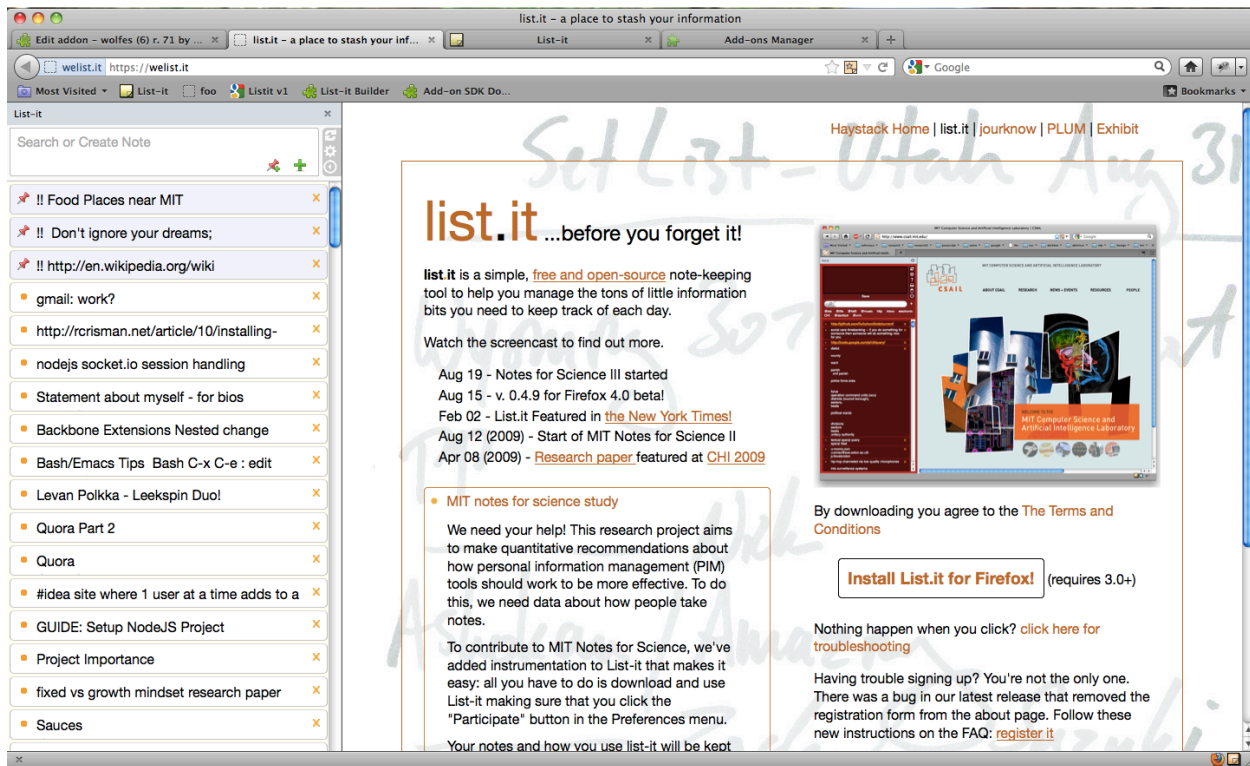


Figure 21. The List-it Firefox add-on sidebar is visible within the browser to the left of the currently focused page. Switching tabs keeps the sidebar in place.

Clicking the note icon in the lower right of the browser, in the add-ons toolbar, or using the hotkey, opens the sidebar within the browser next to the currently focused page. The currently focused page is automatically resized to make room for the sidebar. Changing tabs keeps the sidebar visible on the left side, with the website on the right switching to the new tab. Unlike the chrome extension which can get the full height of the screen to show its UI, the Firefox add-on is displayed under the toolbars and tabs of the Firefox browser, which is less ideal for smaller screens. The browser window can be resized to only show the List-it sidebar, such that a user can place only the sidebar next to another application window on their computer, such as a document or code editor.

The screenshot below shows the List-it application as a regular website in the Chrome browser. The UI does not fully support mobile browsers such as for the iPhone, Android phones, or the iPad. There are two types of modifications that will need to be made to support mobile phones. The first to work on is adding support for features of the major browsers that mobile browsers lack, such as the native JSON object for parsing objects. After these features have been added and the code isn't missing anything it needs to run, the second modification is modifying the user interface to be more friendly for use on a smaller screen.

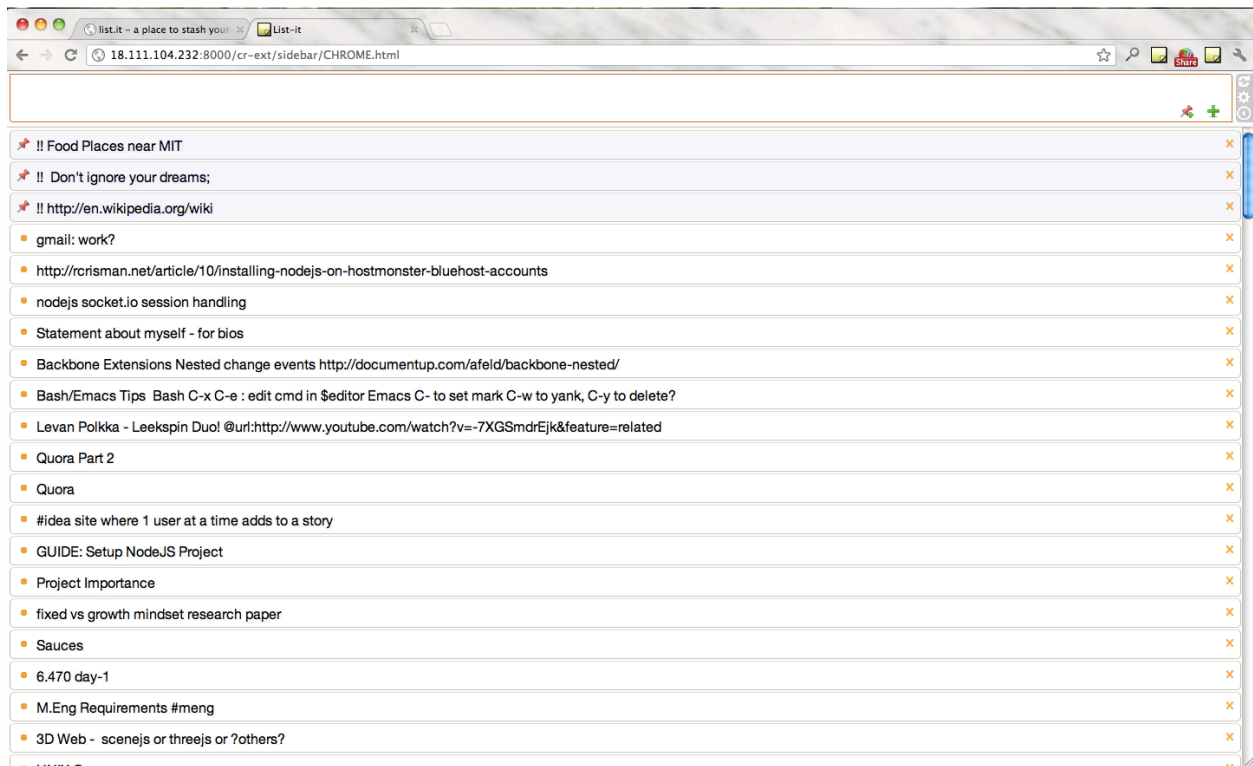


Figure 22. The List-it Application in the Chrome browser as a regular website.

Unlike the Firefox Add-on or the Chrome Extension, the regular website version of the application does not have the browser's permission to keep track of the last focused tab a user has looked at. This means the new Floating Notes feature that floats notes created on the website a user is viewing will not work for the website version of the List-it application. The main benefit of the website version is that users can access their notes from any computer with a browser, such that they don't have to install List-it on a public or friend's computer. The website

version also provides more horizontal space for creating and editing larger notes. The main drawback is that users have a harder time viewing their notes and using other applications, or viewing other tabs, at the same time.

7.3 Design of Chrome Extension

The new List-it application has changed architectures several times during its creation. The structure required for web applications, where a single HTML page shows a website, is different from the Chrome Extension and Firefox Add-on structures. The current application uses the JavaScript libraries jQuery, Backbone, Underscore, and the doT.js templating library to provide a combination of publisher/subscriber in the form of a global event aggregator and model-view-controller (MVC) software design patterns.

In a Chrome Extension, there are two main HTML pages. The background page is a persistent program that is continually running, and can even continue running while the browser is closed. When a user asks to see a sidebar for their notes, the background page creates and places a new popup window next to the focused browser window that loads a different HTML page. The background page is an intermediary between the sidebar page and the server, and is responsible for many tasks, including:

1. *Syncing* notes every 10 minutes and activity logs every 30 minutes with the main List-it server.
2. *Saving* new and updated notes and activity logs in the browser's HTML5 localStorage.
3. *Managing* the appearance of the many sidebars, to be able to resize a browser window when its sidebar is closed.
4. *Messaging* all sidebars with updates when any one sidebar creates/edits/deletes a note, or causes a change that updates the background page's model and that the other sidebars need to show.

The HTML page that displays the sidebar is the other half of the Chrome Extension. This page is responsible for displaying a user's notes, providing note creation and search features, allowing a user to synchronize their notes, opening the settings page, in addition to a selection of

other features. A user can open one sidebar per Chrome window, which means there may be multiple sidebar HTML pages running and communicating with the single persistent background page. The sidebar page consists of models and views for the following major components:

1. Combined note creation / search box.
2. Collection of pinned and normal notes.
3. Options in column next to note creation box.
4. Account registration/login.
5. Status message display.
6. Message center for connecting to the persistent background page.

7.4 Re-design for Web Compatibility

To make the codebase work as a website, the two HTML pages that make up the Chrome Extension need to be combined into one HTML page. The first attempt was to create two sidebar HTML pages, one for the chrome extension, and one for the regular website view that included the additional javascript files from the background page. This meant edits to the HTML page responsible for showing the sidebar needed to be reflected in both pages, which was less than ideal. A solution was found to let a single sidebar HTML page work as both the Chrome extension sidebar and the regular website application. To make this work, the best solution was to limit all communication between the background and sidebars to two channels, because the sidebar code loaded and replaced the same modules loaded from the background page. One channel restricted the sidebar page to only accessing the background's controller object. The other channel allowed a proxy to re-publish the events published by background page's model in the sidebar page. This turned out to be a positive feature by helping enforce that the model-view-controller (MVC) design pattern was followed, where the sidebar was considered the view by the background page. All communication from the sidebar page was sent to the background's controller namespace. The sidebar uses this line of code when it is loaded as a chrome extension to replace its version of the controller object with the controller object from the real background page:

```
controller = chrome.extension.getBackgroundPage().controller;
```

In the opposite direction, all communication from background pages were sent to the sidebars through a publisher/subscriber proxy system. The only two ways for the background page to send a message to the sidebar page is for the background page's model object to publish a message, or for the sidebar to ask the background page's controller object to return a value or pass a value back to a continuation function. For the Chrome extension, all such messages are taken and sent directly to the sidebars using the Chrome Extension messaging API. The sidebars then re-publish the message from a proxy model object, and the rest of the sidebar page listens to this proxy publisher instead of the background's model's publisher.

7.5 Firefox Add-on Design

The same codebase that powers the website described above also supports a Firefox add-on. The Firefox add-on is created by having a small additional javascript program that controls the opening and closing of the List-it sidebar in the Firefox browser. The Firefox add-on code uses a third-party library to create a sidebar and load the List-it web application. The Firefox add-on can then message the List-it web application loaded in the sidebar of the browser about the Firefox API supported events that occur in the Firefox browser. An example of one such event is when the user changes which tab is currently focused. The add-on code messages the sidebar that a new tab has been focused, so new notes that are created will know which page the user was looking at and allow the user to open the page through the automatic bookmark feature. Additionally, it provides an extra resource for logging data related to how a user interacts with their List-it application in the sidebar of their Firefox browser.

7.6 Future Research Potential

The new List-it application allows users to use the same application across many platforms. It allows future List-it developers to implement new features across all the applications without writing their new feature multiple times. This makes it easier to test new features across all three platforms. Users benefit by having access to List-it as a Chrome Extension, Firefox add-on, and regular website. Users can install the application on their personal and work machines, and still gain access to their information from a public computer.

The new List-it is instrumented to collect activity logs for the old and the new features, and will provide further information about how users use micro-note PIM tools for future research projects.

8. Conclusion

Information overload is a large problem in this digital age. Effective PIM tools minimize the amount of perceived effort required by an individual to capture and re-access their personal information. PIM tools that are difficult to use discourage information capture, reducing their effectiveness later when the information is needed for future tasks. Few people can say that they have never forgotten anything without any later consequences for their forgetfulness. The more that PIM tools can alleviate this problem, the less anxiety individuals will experience due to forgotten pieces of information.

Micro-note PIM tools address several aspects of the problems with current PIM tools. Information can be easily captured by a user by requiring no structure at capture time, letting the user jot down just what they want to say and then saving the new note and moving on with their activities. A challenge for micro-note tools is that while an individual may be encouraged to capture more of their pieces of information, there is less structure for the tool to help the user re-access their information. This is less of a problem for the two types of interaction styles discussed in this thesis that maintain smaller note collections, Minimalists and Revisers. For the interaction styles with more notes at any given time, Packrats and Sweepers, this may present a challenge for a micro-note tool to be able to help a user find relevant notes for their needs.

Analyzing two years of List-it usage led to many insights into PIM tool usage. Examining users resulted in the discovery of four types of interaction styles that describe how users interacted with their PIM tool: Packrats, Minimalists, Sweepers, and Revisers. Interestingly, some users were best described as a combination of these strategies. Examining notes led to five major categories of personal information: reminders (memory triggers), reference items, journal/diary entries, scratch pad (external cognition), and posterity. However, there was also a long tail of different information forms that users kept in List-it. From this understanding of different types of users and the types of notes they keep, hypotheses were formed for four unmet needs faced by certain types of users for particular types of notes. For each hypothesized need, a new feature was included in the new List-it application to explore possible solutions. These features have also been instrumented for recording their usage. The

new List-it application will provide a piece of the groundwork for future studies about how micro-note PIM tools benefit users.

The addition of new features to any application is not a complete solution, as users can experience feature overload in the same way they currently experience information overload. Google's Gmail uses a labs approach for adding features that are likely only useful to some fraction of the total users, where features are turned off by default, but can be turned on by users. A next step I propose for PIM tools is an automatic matching system where features can be matched and advertised to the users they would be more relevant to. For example, pinned notes helps users keep some notes on top to prevent them from being buried. This feature has the potential to be much more useful to users with dozens or hundreds of notes than for minimalists who have all their notes visible without needing to scroll.

9. References

1. H. Borko. Information science: What is it? *American Documentation*, 19(1):3–5, 1968
2. B.C. Brookes. The fundamental equation of information science. *Problems of Information Science*, FID 530:115–130, 1975.
3. Richard Boardman. Improving tool support for personal information management, September 2004.
4. William Jones. Personal information management. *Annual Review of Information Science and Technology*, 41(1):453–504, 2007.
5. Max Van Kleek, 2011. Effort, Memory, Attention and Time: Paths to More Effective Personal Information Management. PhD Dissertation, Massachusetts Institute of Technology.
6. Mica R. Endsley. Measurement of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37:65–84(20), March 1995.
7. M. G. Van Kleek, W. Styke, m. schraefel, and D. Karger. Finders/keepers: a longitudinal study of people managing information scraps in a micro-note tool. In *CHI '11: Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 2907–2916, May 2011
8. A.: Using maximum entropy for text classification – Nigam, Lafferty, et al. - 1999
9. Card, S.K., Robertson, G.G., Mackinlay, J.D., *The Information Visualizer* (1991), CZ-ZI '91 Conference Proceedings, New Orleans, 181-188.
10. Richard Boardman and M. Angela Sasse. Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management. In *Proc. CHI '04*. ACM Press, 2004.