

# Evolution and Evaluation of an Information Scrap Manager

Michael Bernstein<sup>1</sup>, Max Van Kleek<sup>1</sup>, mc schraefel<sup>2</sup>, David R. Karger<sup>1</sup>

<sup>1</sup>MIT CSAIL

32 Vassar Street, Cambridge MA  
{msbernst, emax, karger}  
@csail.mit.edu

<sup>2</sup>Electronics and Computer Science

University of Southampton  
Southampton, UK, S017 1BJ  
mc+chi@ecs.soton.co.uk

## ABSTRACT

The Jourknow project addresses the question of freeform notes known as *information scraps*. We are focused on unifying the user’s PIM tools, providing lightweight input and enabling support for uncommon types of personal information. In this position paper we describe our current directions and lessons learned from work on the Jourknow system. We have redesigned our lightweight input language for capture of arbitrary data types and developed a mobile phone client. We report on feedback from a weeklong user study, generating design recommendations for systems like Jourknow. Our study identified the importance of supporting common PIM data in addition to uncommon information types, of providing a mobile solution for capture and retrieval of notes, and of integrating with current user tools and practice. With respect to evaluation, we reflect on the large scale of personal information needed to evaluate our tools.

## Author Keywords

Personal information management, lightweight input, information scraps

## ACM Classification Keywords

H5.2. User interfaces.

## INTRODUCTION

Since the development of the Jourknow system described in [12], our research group has continued to iterate upon and develop our vision of the next generation of PIM tools – tools that unify the user’s PIM experience, provide lightweight input, and support uncommon types of personal information. We began by investigating unstructured personal information called *information scraps* via a 27-participant study involving semi-structured interviews and artifact examinations [4]. We have since redesigned the Jourknow system’s interface and our command input language, and introduced a mobile phone client to capture

thoughts while away from the computer.

In this position paper, we reflect on knowledge gained from our user study and refine our notions of how to best approach the design and evaluation of new PIM systems. We have found it insufficient to build tools focused only on consolidating uncommon PIM information types – common PIM data types like appointments and bookmarks are a *sine qua non*. We have verified the importance of supporting mobile capture and access, though specific phone affordances such as keyboard layout have a large impact. Finally, in the evaluation space, we have recognized the importance of scale: techniques designed to operate with large amounts of personal information can have trouble scaling down to the amount of information generated in a weeklong longitudinal study.

To follow, we begin by outlining the overarching goals of our research agenda. We then describe novel features of the Jourknow system, and outline our user study. We describe our results, and reflect on implications for the PIM field at large.

## VISIONS OF PERSONAL INFORMATION MANAGEMENT

Jourknow is working toward the following research goals: a unified data experience, lightweight input, and support for uncommon information types.

### Unified Data Experience

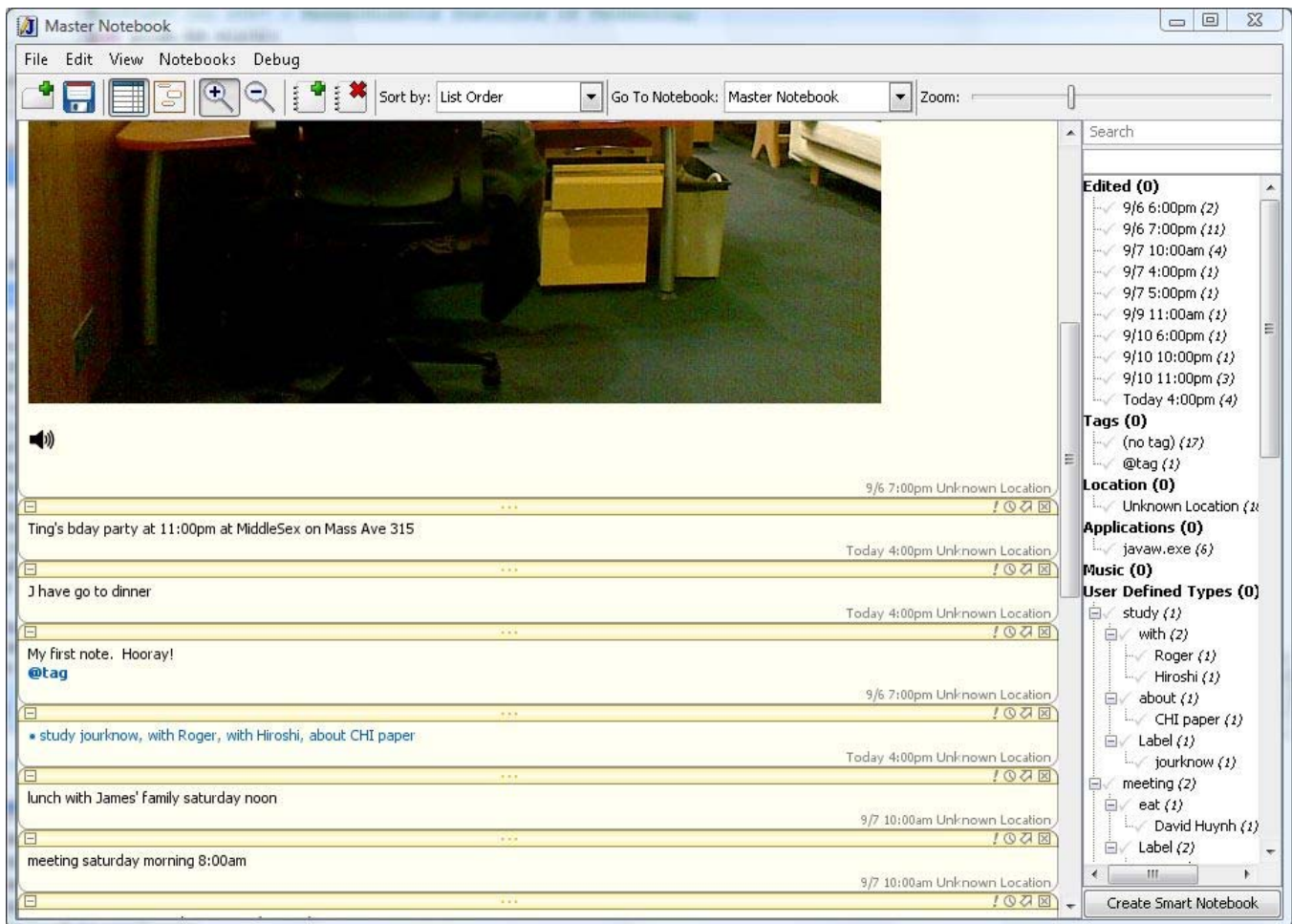
One of Jourknow’s central tenets is that the user should interact with *data* rather than *applications* [8]. Our investigations have revealed that one of the most desired qualities of information scraps is the flexibility of their representation: a text file can record a book title as easily as a phone number. We wish to extend this concept to the interactive world, invoking one interaction path that can intelligently handle all of a user’s data, regardless of type.

### Lightweight Input

We often generate information scraps when we associate too high a time or effort cost with navigating the user interface of our PIM applications [4, 12]. Thus, the Jourknow project focuses on enabling as lightweight an interaction as possible. We envision the recording of new PIM information to be as simple as scribbling “mtg w/ Sanjay @ 5” onto a Post-It note.

### Support for Uncommon Information Types

Our most common PIM applications handle a large percentage of our personal information, including e-mails,



**Figure 1.** The redesigned Jourknow client. Notes, including cameraphone pictures and recorded audio, are on the left. To the right is the faceted browsing panel, with context information at the top and Pidgin information at the bottom.

contact information, and calendar appointments. However, information scraps hold an extremely wide variety of data beyond this – information that we capture commonly but is uncommon for others, for example: expense reports, how-to guides, and fantasy football lineups. We envision the next generation of PIM applications to support these uncommon types as smoothly as they do the most common PIM information types.

### RELATED WORK

Our explorations span several research domains; we have built our knowledge on top of a wide variety of ethnographic research investigating various aspects of information scraps, especially notes [10], to-dos [2], and e-mails sent to oneself [3]. We take special note of the inherent tension between a need for lightweight entry and a desire for structured representation later [7]. Studies of remembrance habits then informed us of the various mechanisms our users might utilize to re-find information, such as relevant people and situations [5]. Here we drew on systems such as Stuff I've Seen [6] for design inspiration as well as existing tools in support of lightweight note-taking [1].

### ETHNOGRAPHIC INVESTIGATION

We conducted a study consisting of 27 semi-structured interviews and artifact examinations of participants' physical and digital information scraps, at 5 different organizations. Details are available in [4]. We found information scraps to play five major roles in our participants' lives: temporary storage, archiving, work-in-progress, reminding, and capture of uncommon data types. We also synthesized five desired affordances of future information scrap tools: lightweight entry, freeform contents, cognitive support, visibility and reminding, and mobility.

### JOURKNOW

Jourknow is an information scrap manager built around a notebook metaphor. It is primarily intended to be a capture interface for thoughts and bits that might otherwise be lost or fall between the cracks of other PIM applications. Jourknow is also a research platform for us to investigate novel interaction techniques for information scrap management, such as context-based re-finding, concise data entry and ubiquitous capture.



**Figure 2.** Through a right-click interaction, the user can gather together all information Jourknow knows about an entity.

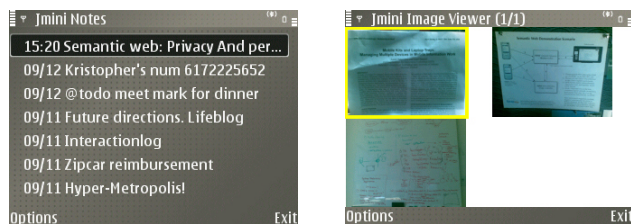
The original incarnation of Jourknow focused on providing a lightweight text input mechanism for structured information entry and context capture for re-finding purposes, centered around a notebook-style design [12]. Based on the results of our study, our most recent efforts have redesigned the Jourknow client (Figure 1), iterated on the *Pidgin* input language, and added a mobile Jourknow client. Here we describe the new version of *Pidgin* as well as the mobile client.

### Lightweight Language for Uncommon Data Types

As part of Jourknow, we have designed various incarnations of *Pidgin* languages bridging the barrier between unstructured human text, which is easy to write but hard for computers to interpret, and structured application input, which may be easy to interpret but can impose unacceptably high requirements on users. In our original work, we designed a language that could handle common PIM types in forms close to natural language, as well as a second language with more complex syntax that could handle arbitrary types.

In our current work, we redesigned the latter language for arbitrary data types to one that we believed would be more usable. The resulting language, called TurtleDove, is comma delimited. Users begin a line with a period, marking that Jourknow should interpret the following text. The next word is understood as the information type (e.g., meeting, todo, shoppingList), and the rest of line up to the next comma is treated as the name of the item. The rest of the line is a series of property-value pairs, delimited by commas. For example:

*. meeting organizing payment plans, with david, at 5pm*  
*. todo buy adi birthday present*



**Figure 3.** The JourMini mobile phone client.

*. shoppingList this week, item dairy, item milk, item OJ*

By a right-click interaction, the user can inquire for details on a pidgin phrase (Figure 2). TurtleDove statements are also added to a faceted browsing panel (Figure 1, right panel), so that the user can perform searches for notes by TurtleDove phrases. For example, the user can search for all class notes where Cain was the lecturer, or for this week's shopping list.

### Mobile Information Scrap Management

Our ethnographic research indicates that ubiquity is a common failure of both pen-and-paper and digital information scrap tools. Bothering to record information is useless if that information isn't available later when the user needs it. Specifically, the laptop form factor misses out on many opportunities for information scrap collection; as Oulasvirta notes, the time taken to open the laptop, wait for it to boot, open the program, and type the information may simply not be worth it [11].

To begin to address this issue, we have created JourMini, a mobile phone Jourknow client (Figure 3). JourMini synchronizes its notes with Jourknow, so any notes created on the desktop are available on the mobile device and visa versa. In addition, it allows users to attach photos from the phone's camera and audio recordings directly to notes.

### USER STUDY

We performed a week-long evaluation of the Jourknow system, including TurtleDove and JourMini, as well as the context capture elements from the original prototype. In this section, we detail the study and our results.

### Method

We recruited 14 participants from our university (ages 18-41, median 26). 7 were students at the business school, 1 was visiting computer science faculty at our university, 2 were undergraduates, and 3 were graduate students in computer science. There were 10 men and 4 women. We randomly divided the group into seven participants who received just the desktop version of Jourknow, and seven participants who received both the desktop and the mobile version of Jourknow enabled. Approximately half of the participants with phones received a phone with a typical 12-

key numeric keypad (the Nokia N95), and the other half received a phone with a full QWERTY keyboard (the Nokia E91i) – this was done to investigate any effect of mobile keypad type.

Participants were instructed to introduce Jourknow into their everyday practice, and to make extra effort to use the software to capture their thoughts and notes. They then used the Jourknow client for a period averaging eight days, including one weekend. Jourknow was instrumented to record a variety of participants' interactions with the software: when it gained focus in the operating system, when new notes were created, when notes were edited, when faceted browsing and search were used, and so on.

### Results

Here we focus on the feedback we received, centered on three aspects of Jourknow: context, TurtleDove, and JourMini.

*Context-based Re-finding.* We found that our participants made very little use of the contextual features, rarely viewing the context associated with a note or navigating via the associated context to re-find a note. Participants pointed out that much of the contextual information was incomprehensible, for instance wireless router *ssid*'s – and since Jourknow was logging a great amount of context, such information overwhelmed the facet panel. Most participants generated a very small number of notes by the end of the experiment, and could thus linearly scan the list more quickly than might locate the correct facet values.

*TurtleDove Language.* Our participants generally found the comma-delimited syntax of TurtleDove too complicated to use. Engineers and programmers understood the format, but others found it abstruse and gave up after a small number of attempts. We found few instances of uncommon PIM type capture beyond shopping and to-do lists – both examples given during the training period. Participants tended to forget syntax quickly after the training session, likely contributing to our null result. Several participants requested that we re-introduce Jourknow's functionality to push TurtleDove expressions into applications such as the calendar.

*Mobile Client.* Mobility was our most successful venture – reactions to JourMini were generally positive. Notes taken on the mobile client included many text scraps, as well as several cameraphone pictures. Participants using phones with full QWERTY keyboards were in general more positive about the mobile experience than those with traditional 12-key number pads, due to the ease of typing. Furthermore, participants who lived primarily digital lives and used little or no paper also found the system more useful than those who used paper regularly for such notes, as they already had a functioning mobile solution in scraps of paper.

### DISCUSSION

Our lightweight languages were originally designed to serve as lightweight entry of two data classes: common PIM types (using simple language), and uncommon data types (using more complicated syntax). With TurtleDove we disabled the former by removing the functionality to push Pidgin expressions into common applications, thinking that the opportunity to record uncommon data types would provide enough of a benefit to our users. Our participants indicated that these common PIM types would be at least as important as the uncommon item capture we enabled.

Testing with a population heavy on business students underscored the importance of integration with users' critical work pathways. Several of these participants had extremely well-defined information handling routines that we were unable to penetrate with a system that did not directly integrate into their tools. Whether this result is an issue of adoption for evaluation purposes (as Kelley and Teevan suggest in [9]) or a more general design critique is an open question.

Finally, we reflect on the importance of data scale in studies of systems like Jourknow. The context information in particular proved unbeneficial to our participants over such a short period of time, due to both the small number of notes they accumulated and our participants' still-intact memory of notes' contents. To better stress-test the context features of Jourknow, our participants must gather notes over a long enough period of time that these mechanisms may become useful, or instead seed the application with existing notes.

### CONCLUSION

In this position paper we have focused on lessons learned from our continued research on the Jourknow project – lessons that we believe are applicable to PIM tools attempting to break out of the traditional application model. The issue of scale is an important one: we had both too small a scale (users did not generate enough personal information during the week of our study to make use of some features) and too large a scale (Jourknow overloaded the user with a bevy of contextual information surrounding all notes). Mobile access seems a fruitful direction for future tools, especially for users with QWERTY-keypad cell phones and who have given up paper as a primary recording mechanism. Finally, we have underscored the importance of command languages' handling of common PIM types in addition to creating support for uncommon PIM data types.

### REFERENCES

- 1 Bederson, B. Interfaces for staying in the flow. *Ubiquity* 5(27). 2004.
- 2 Bellotti, V., Dalal, B., Good, N., Flynn, P., Bobrow, D.G. and Ducheneaut, N. What a to-do: studies of task management towards the design of a personal task list manager. In *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems*. pp. 735-742, 2004.

- 3 Bellotti, V., Ducheneaut, N., Howard, M., Smith, I. and Grinter, R.E. Quality Versus Quantity: E-Mail-Centric Task Management and Its Relation With Overload. *Human-Computer Interaction* **20**(1). pp. 89-138, 2005.
- 4 Bernstein, M., Van Kleek, M., Karger, D.R. and schraefel, mc. Information Scraps: How and Why Information Eludes our Personal Information Management Tools. To appear in *ACM Transactions on Information Systems*. 2008.
- 5 Czerwinski, M. and Horvitz, E. An Investigation of Memory for Daily Computing Events. In Proceedings of *HCI: International Conference on Human-Computer Interaction*. pp. 230-245, 2002.
- 6 Dumais, S., Cutrell, E., Cadiz, J.J., Jancke, G., Sarin, R. and Robbins, D.C. Stuff I've seen: a system for personal information retrieval and re-use. In Proceedings of *SIGIR: ACM Conference on Information Retrieval*. pp. 72-79, 2003.
- 7 Kalnikaite, V. and Whittaker, S., Software or wetware?: discovering when and why people use digital prosthetic memory. In Proceedings of *CHI: ACM Conference on Human Factors in Computing Systems*. pp. 71-80, 2007.
- 8 Karger, D.R. Unify Everything: It's All the Same To Me. In Jones, W. and Teevan, J., eds. *Personal Information Management*, University of Washington Press, Seattle, WA, 2007, pp. 127-152.
- 9 Kelley, D. and Teevan, J. Understanding What Works: Evaluating PIM Tools. In Jones, W. and Teevan, J., eds. *Personal Information Management*, University of Washington Press, Seattle, WA, 2007, pp. 190-205.
- 10 Lin, M., Lutters, W.G. and Kim, T.S. Understanding the micronote lifecycle: improving mobile support for informal note taking. In Proceedings of *CHI: ACM Conference on Human Factors in Computing Systems*. pp. 687-694, 2004.
- 11 Oulasvirta, A. and Sumari, L., Mobile kits and laptop trays: managing multiple devices in mobile information work. In Proceedings of *CHI: ACM Conference on Human Factors in Computing Systems*. pp. 1127-1136, 2007.
- 12 Van Kleek, M., Bernstein, M., schraefel, mc. and Karger, D.R. GUI--Phooey!: The Case for Text Input. In Proceedings of *UIST: ACM Symposium on User Interface Software and Technology*. pp. 337-346, 2007.