

INFORMATION SCRAPS

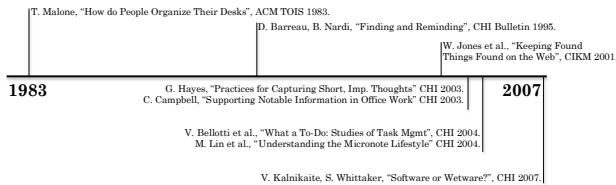
Michael Bernstein
Max Van Kleek
mc schraefel
David Karger
{msbernst, emax,
mc, karger}
@csail.mit.edu
MIT CSAIL and U. Southampton

Next generation interfaces for the lost bits

great ideas	friends' phone numbers	reminders to call home	name of that new song	new year's resolutions	world domination plans
grocery lists	books to read	names of people you met	directions to that place	pasted web snippets	design sketches
web site URLs	shipment tracking #s	a tasty dish you ordered	name of cousin-in-law	incomplete blog entries	places you want to go
bookmarks	recipes	new project ideas	podcast notes	sketches	class notes
favorite quotes	impromptu poetry	how much \$\$\$ you owe	potential jogging routes	records of bill payments	good hotels in SF
to-dos	command line wizardry	where to find good tea	vacation travel itinerary	mental shortcuts	crazy ideas

What are information scraps?

Ethnographic work since the 1970's has seen that knowledge workers often use **short notes of unstructured text**, for example on scraps of paper or in e-mails to themselves, that contain a wide variety of personal information. Despite the evolution of modern personal information management tools, today we continue to see very similar practices, **using post-it notes or free text files** instead of our inveterate information applications. **These 'scraps' of personal information are what we call information scraps.** Our goal is to understand why information scraps exist and to build better tools to support their use.



What information scrap breakdowns exist in current applications?

- Input bottleneck:** the prevalence of forms, required fields and widgets that must be manually manipulated forces the user to slow down in order to carefully formalize thoughts and navigate the interface, making cost of PIM use outweigh advantages
- Inadequate schemas:** PIM applications can currently only handle a small set of fixed schemas as data types, such as addresses, to-dos and calendar events; this ignores a large set of other scrap types
- Fragmentation:** PIM applications segregate data by schema, resulting in data fragmentation; users typically have to input and retrieve information of different data in different schemas by different mechanisms
- Short half-life:** the contextual information people need to remember in order to decipher the meaning of their self-notes fades quickly with time
- Mobility:** information scraps often occur at unpredictable times and locations, when the user is not near a desktop or laptop computer to record the thoughts

What next?

Ethnographic study: A cross-tool study utilizing semi-structured interviews and walkthroughs of users' physical and digital personal information to generate a taxonomy of types of information scraps and to understand the different ways they are used. This includes an investigation of the continued use of unstructured, co-opted digital and physical tools.

Sloppy semantics: Extend jourknow's text analysis to allow for extraction of information by automatic alignment with predefined semantic web ontologies using 'sloppy programming' techniques demonstrated in Little et al.: e.g., "mtg 5pm Karger."

Personal Lifetime User Modeling (PLUM): Incorporation of the user's desktop activity models from the time that a note was made to enable named entity and coreference resolution in information scraps. This will allow jourknow to understand whether the Denny you're mentioning is your skiing friend, your coworker, or the name of a restaurant.

JOURKNOW: INFORMATION SCRAP CREATION, MANAGEMENT AND MANIPULATION



For every note in the journal, jourknow captures a wide variety of contextual information, including **what the user was doing** when he or she wrote the note, **where the user was located**, and **images** of both the user and his or her desktop. These cues are intended as a memory primer to assist later interpretation of notes, as well as to simplify retrieval of related resources.

Information scraps created in a hurry tend to be extremely brief, incomplete sentence fragments and partial phrases, which often contain abbreviations, many named entities, and occasionally omit prepositions or verbs.

To get around challenges, we let users easily define simple grammatical forms we call **pidgin expressions** and apply grammar-based parsing techniques. Users can define their own grammatical forms to express any structured information they like.

Once an expression is recognized, jourknow provides feedback to the user in structured form, as confirmation of how the text was interpreted; this allows users to edit and make corrections.

