

# Popcorn: the personal knowledge base

Stephen Davies

University of Colorado, Boulder  
430 UCB  
Boulder, CO 80309-0430  
+1 (303)439-8561

stephen.davies@colorado.edu

Scotty Allen

University of Colorado, Boulder  
430 UCB  
Boulder, CO 80309-0430  
+1 (303)546-0251

scotty.allen@colorado.edu

Clayton Lewis

University of Colorado, Boulder  
430 UCB  
Boulder, CO 80309-0430  
+1 (303)492-2844

clayton.lewis@colorado.edu

## ABSTRACT

People often use powerful tools to manage the documents they encounter, but very rarely to store the mental knowledge they glean from those documents. Popcorn is a *personal knowledge base*: an experimental interface and database designed to store and retrieve a user's accumulated personal knowledge. It aims to let the user represent information in a way that corresponds more naturally to their mental conceptions than simply text would, in part by making heavy use of transclusion: sharing items among multiple contexts. This paper describes the design rationale for the system, contrasting it with related efforts, and presents the results of deploying it to a group of volunteers who used it in real-world settings. The results, while revealing some limitations in the tool, and some challenges in coping with knowledge reorganization, suggest that the analysis underlying the design is useful, and that Popcorn is a powerful and effective tool for a variety of intellectual work.

## Categories and Subject Descriptors

H.5.2 [User Interfaces]: Interaction styles; H.5.4 [Hypertext/Hypermedia]: User issues; H.3.3 [Information search and retrieval]: Search processes.

## General Terms

Design, Human Factors.

## Keywords

Personal knowledge bases, concept maps, transclusion.

## 1. MOTIVATION

The so-called “information explosion” has triggered many efforts to help people cope with the data that daily bombards them. Most of these efforts have been focused on improving access to *documents*: the innumerable Web pages, e-mail messages, and files that we work with.

But what about the knowledge that is generated from these documents? Reading, after all, is a process in which an input –

natural language sentences – is transformed into an output – a “mental model” of useful understanding inside one’s own mind[3, 15, 20, 29]. As one studies and interprets a document, one forms a mental representation that is related to but separate from the verbatim words. The mindful reader will bring in background knowledge, form opinions, question conclusions, and extend their current understanding. It takes hard work to fully comprehend a document and form a clear mental impression of it. And this mental impression has lasting value, often greater than that of the document itself.

It can be argued that our knowledge is our most precious resource. But it is rarely materialized outside the mind. Every day we work hard to understand and relate information from multiple, complex sources, but then we most often leave the product of our efforts subject to the whims of our erratic memories. We use powerful, high-tech tools to search for and retrieve documents, to store structured, tabular data, and to communicate with remote colleagues, but very low-tech tools (or no tools at all) to manage the knowledge that all of these avenues generate.

Computers are good at storing information reliably, retrieving it faithfully, and representing flexible data structures. If we can exploit these strengths to create a tool for externalizing knowledge, we could provide a new mode of computer support for intellectual work. Designing such an application is the goal of the Popcorn project.

## 2. RELATED WORK

Popcorn is not the first attempt to design an “intimate supplement to the human memory”; in fact, that phrase first appeared in 1945 in Vannevar Bush’s prophetic “As We May Think” article[7]. Systems that attempt to perform this function can be classified as “personal knowledge bases” (PKBs); that is, they are *personal*, they contain *knowledge*, and they are *bases* of knowledge, rather than scattered, independent files:

- *personal*: a PKB is intended for private use, and its contents are custom-tailored to the individual. It contains trends, relationships, categories, and personal observations that its owner sees but which no one else may agree with.
- *knowledge*: a PKB contains distilled knowledge representations that its owner has mentally generated, not merely the original information sources from which they were derived.
- *base*: a PKB is a consolidated, integrated knowledge store. Because our ideas can be associated in arbitrary ways, the

PKB cannot be partitioned into isolated segments that cannot interrelate with each other.

We refer the reader to [12] for a survey of attempts to design personal knowledge bases. Here we touch on only a few, in order to contrast Popcorn with prior developments.

## 2.1 Note-taking tools

A large number of note-taking and outlining applications are on the market today, offering, for instance, “a trusted thought container”[17], or “an extra brain.”[27] These systems are typically based on natural language. They store free text chunks that the user types or pastes in from other sources. The user can then organize these snippets to reflect a personalized, nonlinear structure, typically a tree hierarchy.<sup>1</sup> (see, e.g.,[5].) Some researchers, recognizing the inherent limitations of the tree model, have extended or replaced it with a paradigm in which items can belong to multiple user-defined categories.(e.g.,[19]) This allows a snippet of text to appear in more than one context, effectively introducing what hypertext pioneer Ted Nelson has termed “transclusion.”[24] Transclusion means that a single item can be referred to in many places, increasing the possibilities for interconnecting related information.

Popcorn borrows from these systems the notion of an ever-growing personal database of information, but diverges sharply from them in that the primary way to encode knowledge is not via natural language sentences, but through informal knowledge structures.

## 2.2 Mind-mapping and concept-mapping tools

The past thirty years have seen a growing enthusiasm for pictorial knowledge representations: thoughts are decomposed into individual elements whose relationships are shown in a diagram. Claims have been made that this is an excellent way to communicate complex ideas, enhance learning, and even to “unlock the potential of the brain.”[8] The two most popular techniques are the tree-based Mind Maps of Tony Buzan, and the graph-based Concept Maps pioneered by Joseph Novak.[25] These techniques are supported by electronic tools (e.g., [6, 9]) with which users can generate and manipulate the knowledge diagrams. Unlike the note-taking tools mentioned above, a user expresses their knowledge not in streams of text, but in pictorial structures where nodes represent concepts.

As we will see, Popcorn shares much with these knowledge mapping tools, since it too depicts knowledge elements and the relationships between them. It differs from them in two important ways. First, tools in this category are all “file-based”; that is, they require the user to create and name a new file for each diagram, and to store that file somewhere on their disk. Second, because elements in different files cannot refer to one another, this structure prevents these tools from supporting an integrated

---

<sup>1</sup> We will refer several times in this paper to the difference between a *tree* and a *graph*. Briefly, a graph is a data model in which any element can be connected or linked to any other. An example would be the World Wide Web, in which any page can refer to any other page. A tree (or *hierarchy*) is a restricted form of a graph, in which each element is linked to another element called its *parent* (or is a *root* element, with no parent), and may be linked to a number of elements called its *children*.

knowledge base. Popcorn overcomes this limitation by allowing the same elements to be shared among different diagrams. This sort of transclusion is very rare among PKB tools, and emerged as an important factor in Popcorn’s success.

## 2.3 NoteCards

The NoteCards project, designed “to help people formulate, structure, compare, and manage ideas,”[16] solved some of the problems of diagram tools. It still promoted the partitioning of a user’s knowledge into separate divisions (“NoteFiles”), but users could store large amounts of information in a single file and treat it as an integrated knowledge base. Also, in addition to the “notecards” that formed the basic unit of information in the system, users could create “browser cards” that contained links to sets of other notecards, and depicted the relationships between them as a graph. Although not emphasized in the NoteCards literature, this supported the transclusion property. An individual notecard could appear on multiple browser cards, thereby showing it in different contexts and related to different sets of cards.

Popcorn resembles an updated NoteCards in many respects, but its data model is a bit different. Rather than distinguishing elements that refer to other elements (browser cards) from elements that do not (notecards), Popcorn allows *all* elements to refer to or include others.

This flexibility leads to a difference in the way content is expressed. In NoteCards, relationships among ideas are often expressed in text on a notecard, since this is easier than creating a separate browser card to capture the relationship. This leads to notecards often containing quite a bit of text. In Popcorn it is more natural to use diagram structures to interrelate elements, and the elements therefore contain less text, often just a name. More content is thus represented diagrammatically than textually.

## 2.4 Compendium

Finally, we mention Compendium[28] and its earlier incarnation gIBIS[11], knowledge representation tools specifically for capturing design rationale and arguments. Compendium is close in spirit to Popcorn, in that it expresses knowledge as a graph, and arbitrary subsets of the graph are held inside individual elements. One important difference between Compendium and Popcorn is that Compendium requires a *fully-connected* graph. That is, in a Compendium knowledge base, all elements must be reachable from each other via the connections in the graph. The moment that a portion of the graph is severed from the “root” (the original node), that portion is abruptly lost. Popcorn relieves this constraint by allowing the user to maintain multiple “islands” of information in their knowledge base and to easily traverse between them.

(Note that we have omitted mention of tools whose purpose is primarily to facilitate access to documents (ie., information sources) rather than formulation of knowledge from those documents. Haystack[2], Memory Extender[18], Presto/Placeless Documents[13], Stuff I’ve Seen[14], and WebTop[31] are all in this category.)

## 3. POPCORN

### 3.1 Design goals

Our effort to develop Popcorn as a useful personal knowledge base was organized around a number of key design goals.

Collectively, these goals form an analysis of the tasks of personal knowledge management.

1. Recording knowledge should be quick and painless. Building knowledge from information is difficult, and requires focus and concentration. A tool won't be used if it adds more than a little to this burden.

2. Returning to knowledge recorded in a different context should be quick and painless. One can be hard at work on a project, rapidly switch context to (say) handle a phone call, and then switch back. The tool must support these context switches.

3. Reorganizing knowledge should be easy. As we learn more about a new domain, our perceptions about it change. The ideal tool should encourage the restructuring of our knowledge to reflect our current understanding. People are reluctant to abandon their mental organization patterns, even when these prove to be suboptimal[30]; the last thing we want a tool to do is further enforce this rigidity.

4. Users should be able to express knowledge both formally and informally. We sometimes wish to take the time to specify knowledge very meticulously, especially in a difficult domain where we had to work hard to discern its precise meaning. Other times, however, being forced to spell out details would only get in our way. A PKB should support any point along this continuum.

5. Public content should be easy to assimilate. One's private knowledge is often extracted from pieces of public information. A PKB should make it easy to incorporate snippets of information from the public realm (say, the Web) into the knowledge base, and to maintain the links to public sources so they can be revisited.

6. The tool should be compatible with human memory. If we wish to archive knowledge and return to it later, then the tool's data model and user interface paradigm should interface well with our internal memory processes. We identified three key points of compatibility.

a. Memory is associative. Evidence suggests that much of our knowledge is held in memory in a structure similar to a concept map[3] The associations between ideas support retrieval in a flexible and fluid way. A PKB should therefore allow users to represent connections among elements and use them flexibly in retrieval.

Furthermore, associations between ideas often cross between contexts. If idea A is associated in human memory with B in one context, and B is associated with C in a different context, C can often be retrieved from A. This pattern of traversal occurs quite naturally due to the phenomenon of "spreading activation"[10]: recalling one item partially triggers the other items we associate with it and renders them more readily available. We aimed to support the same retrieval facilities in Popcorn, by making associations specified in one context usable in other contexts, and by making all the contexts for a given item readily accessible when that item is considered.

b. Working memory is limited. Only a tiny subset of the knowledge in memory is available for mental work at a given time (cf. the concepts of short term memory or working memory in [3, 20, 23].) This suggests that a tool should allow users to represent relatively small portions of their knowledge at a time, and should

allow them to retrieve and view small portions of the contents of the PKB at a time.

c. Finally, visio-spatial organization is a natural part of human thinking[4, 21] and can be an aid to managing information[22]. We wanted the Popcorn user interface to exploit this.

### 3.2 The Popcorn data model

With these goals in mind, we devised a data model to support them. In Popcorn, the basic building block of knowledge is the *kernel*: a named entity corresponding to a real-world concept in the user's mind. A kernel can appear simply as a small shape, and it can also be expanded to show its inner contents, the *kernel view*. Each kernel view is a two-dimensional canvas on which can be placed other kernels and also free-text snippets called *notes* (Figure 1 in the discussion of the Popcorn user interface shows how the data model is rendered on the screen.) It is important to understand that not only can a given kernel view contain many kernels, but that a single kernel can appear on many different kernel views. Indeed, all kernels have global scope in a Popcorn knowledge base, and can appear in any spatial position on any number of other views. This is what gives Popcorn the transclusion property, allowing a single item to appear in many different contexts simultaneously.

Kernels and notes can also have *relationships* with each other, which can be named. Relationships between kernels are "global" in the sense that they are not bound to any particular kernel view. For example, if a kernel "Antony" has a relationship named "married" to another kernel named "Cleopatra," then that relationship holds (and will automatically appear) on *any* kernel view that contains both Antony and Cleopatra.

Finally, notes can have *source* information associated with them. This is useful for snippets of text that were excerpted from information sources, so that the original source of the excerpt can be maintained. (Popcorn automatically captures the source URL of snippets that were dragged into it from a Web browser, as explained below.)

Notice that there are two ways that kernels can be connected to each other. One is explicitly through *relationships*: the user can create a relationship from "Brutus" to "Caesar" named "conspired against." The other is implicitly through *containment*: the user can specify that "Bach" appears *in* the view for the kernel "Baroque composers", which will also establish connections with any other kernels in that view. In figure 1, for example, the kernel "Abraham Lincoln" has a *relationship* with "Mary Todd Lincoln" (which is called "married"), but is *contained within* the kernel called "Lincoln assassination."

Although Popcorn includes the notion of containment (one kernel can appear "in" another), the kernels in a knowledge base do *not* form a tree (or hierarchy). This is because placing a kernel A in a kernel view B does not preclude A from also appearing in kernel view C. "Bach" may well appear in both "Baroque composers" and "Influences on Mozart," as well as many others. Indeed it is very possible (and common) for cycles to exist in the knowledge base: A may appear in B and B in A, for instance. This design decision was made because empirically, people often consider their ideas in alternate contexts. When I am thinking about "students on my research team," one of the entities that comes to mind is "James Carter," along with other students. But when I am thinking about "James Carter," I naturally remember

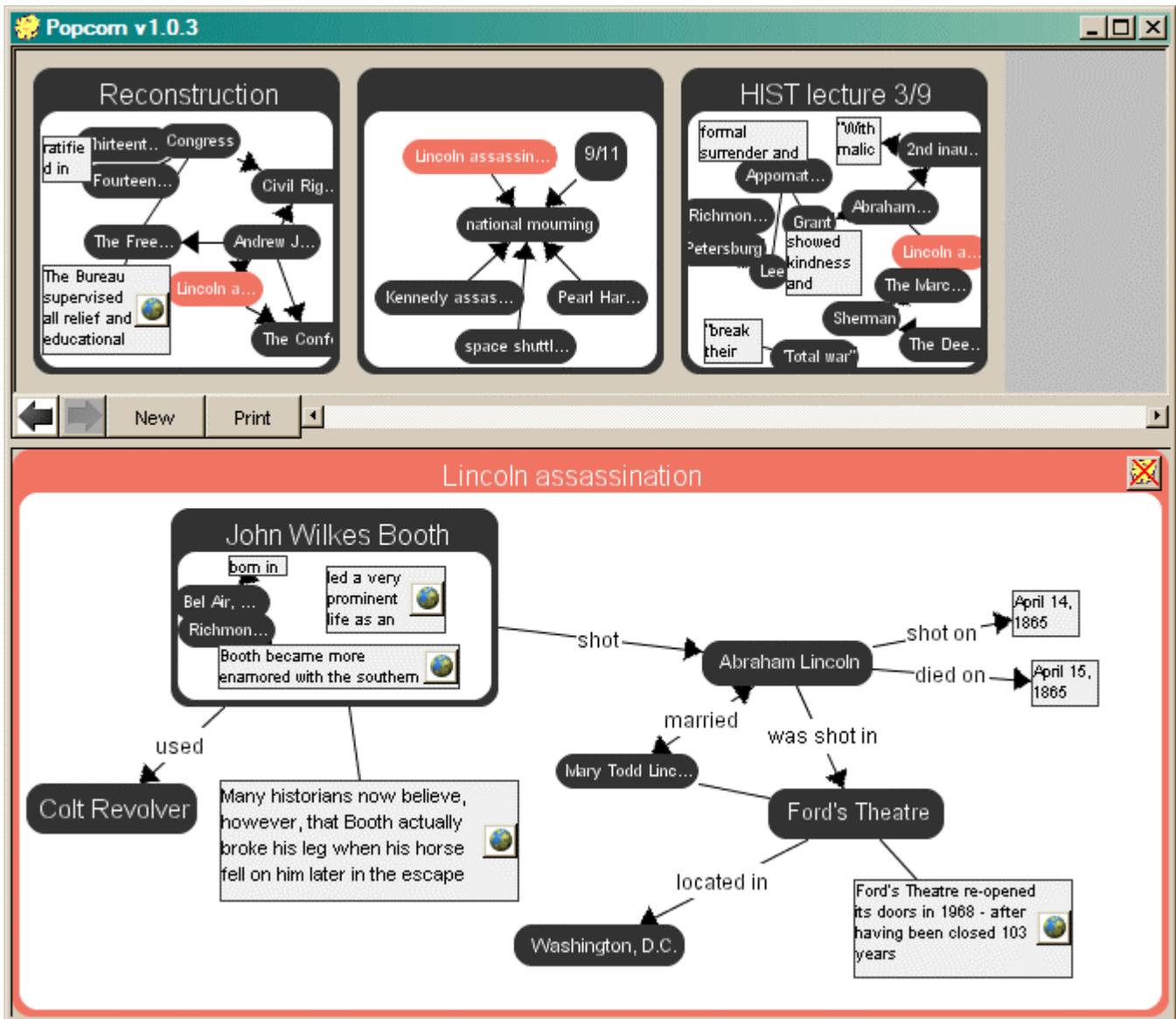


Figure 1. The Popcorn user interface. The view of the active kernel (here called “Lincoln assassination”) occupies the viewport, while all of its parents (other kernel views in which it appears) are displayed immediately above it. The active kernel is displayed in a different color wherever it appears, in order to draw attention to it. Kernels within a kernel view may be expanded, as is the case with the “John Wilkes Booth” kernel here. Popcorn’s layout engine attempts to show as much meaningful information for each component as possible, subject to the space available, while still preserving the spatial relationships between components. Note, for instance, that in the “Reconstruction” kernel (upper left), the kernel named “The Freedman’s Bureau” has been shortened to “The Free...”, and that in many places only the first few words of each note are visible. The user can hover over notes or kernels with the mouse to summon a popup “tool tip” with the complete contents.

that he is one of the “students on my research team,” as well as other things (a grader for my course, a fellow fan of mystery novels, or whatever.) My mind does not hold either idea as “primary,” and so it seems artificial to force one of these two ideas (either “James Carter” or “students on my research team”) to be above the other in some sort of hierarchy. Rather, they are both simply ideas, each of which contain other ideas, and they both exist in my mind.

The kernel view is our effort to work effectively with the limitation of working memory. Rather than attempting to capture all of a large body of knowledge at once, the user represents only a small piece in a kernel view. Similarly, at retrieval time, a kernel view presents only a small part of the total contents of the PKB. Allowing kernels to be embedded in multiple kernel views supports some of the flexibility of human associative memory. It may sometimes be convenient to model knowledge as a hierarchy: “Joe DiMaggio” might be contained within “New York Yankees,”

which in turn is contained under “American League” and then “Major League Baseball.” But we can come to know things about “Joe DiMaggio” that have nothing to do with baseball, such as “Marilyn Monroe was married to Joe DiMaggio.” Popcorn’s data model lets us embed “Joe DiMaggio” in other kernel views (“Celebrity marriages”?) without interfering with our representation of him as a ball player.

It is worth noting that when looking only at a single kernel view, Popcorn’s data model is essentially identical to a concept map[26], or semantic network.[32] Both of these representations also consist of items (often called “concepts” or “nodes”) and the named associations between them. What differentiates Popcorn are the complex linkages that can be created *between* the many concept maps a user creates. Each kernel can appear on any number of other kernel views, and contains its own embedded kernel view besides. The concept maps that a user creates are thus not isolated fragments of knowledge, but part of a richly interconnected knowledge network that can be traversed from context to context via the kernels that are shared between them.

### 3.3 The Popcorn interface

#### 3.3.1 Recording knowledge

Popcorn’s user interface follows this data model exactly. The kernel currently being viewed (known as “the active kernel”) appears in the application’s *viewport*, a rectangular area near the bottom of the screen. The kernels and notes in this active kernel are shown within the viewport, wherever the user has placed them. The top of the window contains the *parents panel*: a scrollable list of thumbnails depicting each kernel view in which the active kernel appears. The active kernel itself is shown in a distinctive color in the thumbnails so it can be identified easily (See Figure 1.)

Kernels and notes can be moved about the viewport using the familiar drag and drop paradigm. Creating a relationship is accomplished by positioning the mouse pointer near the *edge* of an object (the cursor changes to a crosshair) and then dragging to the to-be-related object. New kernels and notes can be added to the view by double-clicking or right-double-clicking, respectively.

When the user double-clicks to add a kernel to the view, they may type a name for it. At this point an autocomplete search pane appears immediately below the kernel name, offering to match existing kernel names – or the contents of notes within them – while the user types. This allows users to place an existing kernel on the view without typing its entire name, and also encourages the re-use of kernels in multiple contexts.

A kernel can be expanded while inside the viewport so that its inner contents are revealed in miniature. This allows users to “peek inside” the kernels in the viewport without changing contexts.

All basic operations in the Popcorn UI are accomplished without menus, toolbars, or palettes. This is possible since Popcorn permits only a small number of operations, and it speeds up the interface experience considerably, a key design goal.

#### 3.3.2 Recovering knowledge

Browsing through the knowledge base can be done in two ways: navigation, and search. To navigate, the user makes some visible kernel the active kernel by double-clicking on it. A “zoom”

animation sequence smoothly brings the selected kernel into the viewport, giving a sense of continuity as the context changes.

Searching is initiated simply by typing when nothing is selected. Typing immediately pops up a search pane, which, like the autocomplete pane for kernel name completion, matches kernel names and contents as the user types. This allows for very rapid switching of contexts: the user only need type the name of something they are thinking of in order to instantly bring that information to the foreground.

These are the *only* ways to bring up previously entered data. There is no all-encompassing browsing function, for instance, that would allow the user to see everything they have entered at a glance. This omission is intentional, part of our designed compatibility with working memory limitations, in which we want all views to be of limited size. An implication of this limitation is that a user can never reach a previously recorded kernel unless they can remember (a) its name, (b) any of its contents (notes), or (c) the name or contents of any kernel it is connected to, either directly or indirectly. We will discuss the implications of this in section III.

#### 3.3.3 Restructuring knowledge

Users can freely drag around kernels and notes in any kernel view, and can even drag them inside or outside of expanded kernels and parents. The intent is that reshuffling objects into different configurations should be easy with direct manipulation.

#### 3.3.4 Incorporating information from public sources

Users can highlight text in any drag-and-drop-enabled application, and then drag the selection into Popcorn, which will automatically create a note for it. To provide further support for incorporating information from the Web, we provide additional integration with the Firefox Web browser[1]. When the user drags an excerpt from a Web page into Popcorn, the created note will have a special icon attached to it (the icon looks like a small earth, and is visible in several of the notes in Figure 1.) The source URL is automatically stored in Popcorn. If the user later clicks on the icon, a new Firefox window will automatically pop up and connect to the Web page, scroll to the excerpt, and highlight it. This makes it easy to incorporate publicly-available information into one’s personal knowledge store, without dealing with the hassle of maintaining links or copying URLs.

## 4. USER RESULTS

We deployed our prototype version of Popcorn to twenty volunteer users for a one-month trial period.<sup>2</sup> This group included students from a variety of engineering disciplines, computing professionals in various roles, a schoolteacher, a business consultant, a newsletter editor, a product development manager, and a Presbyterian minister.

Our goal was to determine how naturally the tool would integrate with a user’s daily life, how much it would be used and for what tasks, and so forth. Hence we simply installed the software on each user’s system, gave them a brief tutorial, and then instructed them to use the system in any way they wished during the testing

---

<sup>2</sup> Several users requested to continue to use Popcorn after the one-month trial expired, and the statistics presented here in some cases represent a longer period of usage.

period. No specific usage requirements (total amount of data to be entered, number of times per week the application was to be used, etc.) were stated or implied.

As might be expected, the results were widely variable. Some testers used the system heavily, nearly every day; others were more sporadic, only entering data on a few days during the month. Some adopted Popcorn as a full-fledged personal knowledge base, recording knowledge about numerous diverse aspects of their lives; others concentrated only on a few domains or even a single domain. The types of information recorded were many and various, including such domains as culinary recipes, event planning and scheduling, personal health concerns, source code snippets, procedures for system administrators, guitar chords, contact information, lesson preparation, and notes on the Russian Revolution, just to name a few.

To give a concrete example, one computer science student used Popcorn to store a variety of programming-related knowledge. This included the relationship between components in the C++ standard template library, which ones he had used in certain modules of a class project, and snippets of Web pages that contained relevant tips and techniques. He also archived the syntax for certain Linux commands that he had trouble remembering, and useful ways he had seen them combined in scripts. This user stored more than just technical knowledge, however. Planning a trip to a dinner theatre involved the times, dates, and prices of shows, directions to the playhouse, and a record of a credit card transaction. Meeting minutes from his part-time job tracked the status of decisions on key projects, and the influences on them. A schedule of deadlines and relevant procedures helped him keep track of his graduation requirements. Notes from all of his class lectures captured and cross-referenced key points and highlighted pertinent sections of reading. And a multiply-categorized “to do” list, crossing several kernels, helped him manage his list of monthly, daily, and even hourly tasks. Note that Popcorn was used to manage knowledge in diverse domains, with varying degrees of structure, complexity, and malleability over time.

#### 4.1 Quantitative analysis

Our testers were invited to run a simple, voluntary program which would extract certain statistical information from their databases. We wanted to get an idea of the general structure of Popcorn knowledge bases, and the degree to which they varied among users. Fourteen of the testers elected to run this program, and a very brief summary of the results is presented in Table 1.

We can discern several important truths about the tool’s usage simply from these statistics. For one, the quantity of knowledge stored – measured as the sum of the number of kernels, notes, and relationships – varies widely among individuals, and is quite bimodal: note that the mean is much higher than the median, which was caused by a cluster of six testers who used the system very heavily. Paradoxically, the PKB size did not always correlate with users’ qualitative assessment of the tool: several users with relatively small knowledge bases said the tool was very effective for them.

Also highly variable were the overall ratio of notes to kernels, and the percentage of relationships that were named (as opposed to being left as simple blank lines.) This underscores how users can differ significantly along the formality/informality continuum. The

extent to which users represent knowledge as concept maps, rather than as raw phrases or sentences, is a matter of personal choice: one tester had eight times as many kernels as notes; another had nearly three notes for every kernel. And some users clearly prefer to precisely specify the nature of relationships between kernels, while others are content to simply draw informal connections. Formality and informality are evidently each called for in different situations, and Popcorn allows both.

**Table 1. Summary of knowledge base characteristics across the test group, giving the minimum, mean, median, and maximum of each statistical measure.**

	<u>min</u>	<u>mean (med)</u>	<u>max</u>
Total size of PKB	86	768.36 (208)	3661
Note-to-kernel ratio	0.12	1.34 (1.19)	2.84
Relationships w/ names	13%	49% (45%)	100%
Empty kernels	36%	53% (51%)	76%
Kernels w/ parents	66%	82% (83%)	96%
Kernels w/ children	14%	28% (28%)	44%
Kernels w/ parents <i>and</i> children	2%	19% (18%)	38%
Kernels w/ multiple parents	2%	15% (15%)	33%

More stable numbers include the percentage of empty kernels (ie., those with no notes or child kernels), the percentage that had parents, and the percentage that had children. Most testers had very close to half their kernels containing inner detail, and half representing simple unelaborated concepts. The typical knowledge base has about 80% of its kernels appearing on some other view, and about 30% containing other kernels within their own view. Curiously, the fraction of kernels that had both parents *and* children fluctuated more widely among users. We suspect that a high number here is indicative of hierarchical thinking.

The most illuminating statistic in this table is the last one: the percentage of kernels that appear on *more than one* other kernel view. In other words, it measures how often a user places an item in multiple contexts. We were surprised to discover how perfectly this number correlates with a user’s subjective appraisal of the tool. Without exception, a high number here (15% or higher) corresponded to an overall positive evaluation and enthusiasm for the application. Low percentages always accompanied dissatisfaction.

This reveals a powerful truth: transclusion may be the most critical ingredient in Popcorn’s success. When users are unable to take advantage of it, they become frustrated and encumbered; but those who master it reap tremendous benefits. It is difficult at this stage to say exactly why it is so important – more directed investigations aimed specifically at uses of transclusion are needed. But our hunch is that this is simply the aspect of Popcorn that diverges most sharply from today’s status quo. The vast majority of information management tools today permit users to “file” items into only one place. But Popcorn encourages the same item to appear in many different contexts, and in each of these to be associated with a completely different set of related items. Mastering this modeling technique, it seems, can unleash new dimensions of multidimensionality that were perhaps always present in the user’s mind, but were never before expressible with conventional tools.

## 4.2 Qualitative impressions

We now present a summary of individual structured interviews with the testers after the one-month trial period. We asked each one a broad range of questions designed to elicit how the tool functioned for them practically, and what obstacles they faced.

Overall, the qualitative impressions were mixed and noticeably bimodal: users tended to either embrace the system enthusiastically, often using it as a replacement for all other forms of knowledge management, or else have great difficulty in adapting to the paradigm. A few occupied a middle ground where they could use the system effectively enough to see its advantages, but ultimately rejected it as it stands because of some inhibiting factor. Here we present observations on usage patterns, strengths, and weaknesses of the tool.

### 4.2.1 Common patterns of use

We saw two very different modes of use emerge during the trial. One is information capture. Popcorn users stored various bits of important data in the tool, loosely grouping them and naming them, and taking great comfort in knowing that they were “safe.” A few examples would be phone numbers, driving directions, passwords, and quotes from Web pages, but the variety here was quite large. This kind of information is typically relevant for a short to medium duration, has very little structure, and its primary purpose is to be later recalled.

The second common use is knowledge formulation. Users brainstormed difficult domains with Popcorn, with the intent to form a better understanding of the topic. They most often did not begin with a thorough grasp of the knowledge, but rather discovered it as they went, using Popcorn as a sort of multidimensional scratch pad. Interestingly, many users reported that once they generated a concept map, even if it was very detailed and crossed over numerous kernel views, they had little interest in being able to return to it again. Hence Popcorn’s primary function as a storage repository is not even used in this case. The knowledge gained theoretically has very long-term value (especially as compared with a phone number or password), and yet apparently its chief purpose is to be formed in the mind, not recalled via the tool.

The interesting fact here is that nearly all users (18 out of 20) employed *both* of these modes, but with little to no overlap. Over time, their knowledge bases became a growing collection of two spheres: loosely connected bits of information, entered hurriedly and recalled frequently; and richly structured knowledge maps, created painstakingly and reviewed rarely. Users reported that both have value, but it is surprising that one tool seemed to serve both purposes.

A third, less common use for Popcorn (reported by 7 users) is to manage ongoing, changing knowledge. Users kept a set of kernels up-to-date so that it always represented the “current state of affairs” in some area. Examples here were personal “to do” lists, project management resource allocations, and the state of ongoing dialogues with customers. Users report that this can be very helpful in staying on top of tasks that involve multiple interdependencies, but that it is impractical for more than a small number of areas simply due to the time required to keep them up to date.

### 4.2.2 Domains

As a knowledge management tool, Popcorn is best-suited to ill-understood, unexplored domains with many arbitrary relationships. These are clearly something that most existing software applications have trouble with, and is what can make Popcorn a very valuable tool. We discovered, however, that the knowledge most users generate is a combination of this kind of domain plus hierarchical (e.g., outlines, org charts), tabular (accounts, contact info), and purely textual (memos, essays) information. Users liked Popcorn for the first kind of knowledge, but for these other areas, many of them (9 of 20) became frustrated with it. The unrestrained flexibility, which is Popcorn’s hallmark, makes it ill-suited to domains that have a regular, predictable structure: for these areas, users prefer tools that more rigidly guide them. And entering textual information is frustrating since it lacks the text editing features of word processors. Hence our original vision for Popcorn as a “store everything” knowledge base needs to be adjusted. If it were better integrated into a user’s overall toolset, each application could be used to process the type of information for which it is intended.

### 4.2.3 Strength: Visual representation

Popcorn’s method of displaying knowledge visually through concept maps was mostly well-received, though not without a few criticisms. Users reported that the spatial nature of kernel views helped them more quickly zero in on important information, more easily get their bearings when returning to a previously created view, and more easily retain the knowledge they created in their own memories (nine testers mentioned one or more of these benefits, even without being asked.) One user said that after constructing a kernel view, a certain concept was thereafter “always in the bottom-right corner” of their mind, even when not using the tool. This feedback was common, and suggests that the act of using Popcorn helps create robust cognitive structures.

Spatial placement on a canvas involves more degrees of freedom than writing on a lined notebook page, and users varied in their ability to adjust to that. For some, it was extremely positive: one stated that after using Popcorn, it was painful to return to linear text because it felt arbitrarily confining. Another commented that working spatially let them “think like a human, rather than the computer forcing me to think like a machine.” A few users (4 of 20), however, found this extra dimension unnecessary and even inhibiting. The knowledge itself was sometimes harder to enter, because in addition to specifying the concepts the user must also specify their spatial positions, which can be distracting. Another frequent comment (from 7 users) was that certain patterns of information – most commonly, lists – did not lend themselves well to a spatial representation.

### 4.2.4 Strength: streamlined user interface

Popcorn’s user interface is unusual in that it features no menus, toolbars, dialog boxes, buttons, or many other elements familiar to most Windows-based applications. Instead, the interface is optimized for maximum speed of data entry and retrieval (via direct manipulation with a single mouse click or keystroke), at the cost of discoverability. We did not know how well users would adapt to this, and were relieved to learn that in most aspects it was welcomed. Nearly every user (17) reported that although the interface took some getting used to initially, they were quickly able to adjust to the paradigm, and soon preferred it for speed’s sake. We believe this is because Popcorn supports such a small

number of operations, all of which are used commonly, and so are easily remembered.

Only two difficulties with the interface were commonly noted. First, our technique for creating relationships requires the user to position the mouse cursor over the edge of the object, and this was hard for some users to carry out precisely. When working quickly, it is easy to miss the zone around the object's edge and accidentally move a kernel instead of creating a relationship (or vice versa.) A meta-drag operation may be more appropriate here, though this requires a coordinated keyboard/mouse effort and this may be cumbersome. Second, some users found initiating a search unnatural, since there is no visual indication that simply typing would trigger a search. Suggestions for improvement included dedicating a small piece of the screen for an always-visible search box, or requiring the user to prefix each search with a special character (perhaps the "/" key, which triggers Firefox's internal page search.)

#### 4.2.5 *Strength: Search and recovery*

Popcorn users were surprisingly successful at retrieving old knowledge from their knowledge base. We expected this to be a problem since recall ought to be more difficult than recognition, and Popcorn requires the recollection of verbatim text (or the verbatim text of a connected kernel) in order to retrieve an item. However, nearly every user (17 of 20) stated flatly that remembering the names of (or retrieval paths to) desired kernels was never an issue for them.

There are several possible reasons for this. For one, the trial period was brief, and it is possible that after a year or more of using Popcorn, users will find recovery substantially more difficult. It is also possible that users' retrieval patterns are usually short-term in nature: one normally needs to return only to kernels that have been created recently, thereby lessening the burden on biological memory. It may well be, however, that Popcorn's recipe of "free associate to a related kernel, then navigate from there" is in fact an effective retrieval technique for the long-term. Longer-term testing should reveal whether this is the case.

#### 4.2.6 *Strength: Multiple contexts*

We were unsure how often users would take advantage of the ability to place kernels on multiple views, since this idea is not supported by most existing applications. We were pleasantly surprised, however, that many testers (14) found this quite natural and useful. Some commented that even after limited usage, kernels began to naturally overlap multiple domains and different views of the same domain, usually without preplanning. This is borne out by the statistical data, and suggests that transclusion is widely applicable to human thinking.

To give but a single example, one common way in which transclusion was applied was with kernels that represented a user's friends and co-workers. A user might store contact information and other miscellaneous facts inside a "person" kernel's view, and then place that kernel in many other views. Each such view would represent a context in which that person played a role: a project they were assigned to, the guest list for an upcoming company party, a list of workers that possessed certain technical skills, possible roles in a pending team reorganization, or the opinions expressed by various colleagues on a particular topic. Since the user thought of that acquaintance in a variety of different settings, they naturally desired to place the

corresponding kernel simultaneously in many different contexts. And by navigating to the person kernel itself, it would be easy to see at a glance all the different contexts in which it appeared.

#### 4.2.7 *Weakness: Reorganization*

Despite our attempts to facilitate the reshuffling of knowledge, Popcorn did not live up to our user's expectations in this regard. Many users (about 15) related that they found it difficult to restructure kernels to conform to new understandings. This is partially due to the specifics of the tool's interface. We currently do not support a "multi-select" operation, which could be used to select and then move a number of related kernels in concert. Also, the screen features just one kernel view at a time (the active kernel in the viewport), with all others relegated to a smaller, thumbnail status. This makes it difficult to break one kernel view into two, since one has to create a new, expanded kernel on the viewport, move some of the contents into it, name it and delete it from the current view, and then reorganize the remnants, an unnatural and tedious process. We also discovered that even sparsely populated kernels tend to fill up over time as more facts are learned about a topic. An eventual reorganization of the kernel is actually the most fortunate outcome here; more problematic is that the user will become reluctant to add newly acquired knowledge for fear of the view becoming overcrowded.

Clearly, better methods for restructuring are needed. We expect that modest enhancements to the user's repertoire of operations (such as adding multi-select and multiple top-level windows) will help a good deal.

#### 4.2.8 *Weakness: Hardware availability*

A basic problem cited by 12 of our users is simply that Popcorn is tied to the platform on which it runs. Hence it is easily accessible only at one's desk, when one is normally working on a specifically computer-supported task. Users wanted Popcorn to be available to them in the kitchen, during meetings, on the road, even in the shower. Even laptop users reported that their machines were not always available or convenient when needed.

We anticipated this problem, but were surprised to discover that knowledge *recovery* was not the feature most sorely missed, but in fact knowledge *entry*. Evidently, users do not need to access their previously entered data in remote settings so much as they need to record the new knowledge that they generate there. The reason for this, as interviews made clear, is that knowledge is elusive: it disappears quickly. One user, for example, recounted that he would often discuss a key idea in a meeting and have some critical insights, but that by the time he got back to his desk to record the knowledge into Popcorn, many of the important nuances had already faded from his memory. For maximum effectiveness, it seems, Popcorn must be made available at the point of knowledge creation. And considering how fleeting newly discovered knowledge appears to be, it could be an extremely valuable tool in this setting.

#### 4.2.9 *Weakness: Not integrated with toolset*

As alluded to earlier, one inhibiting factor is that Popcorn does not interface with most of the other software a typical user employs. Ten users cited this, and the level of integration desired varied among them. Some suggested that if the data they currently managed in other applications could be easily imported into Popcorn, they would be content to abandon those applications entirely and use Popcorn as the sole repository. Others wanted to

use Popcorn and other tools together: ideas included exporting a network of kernels to create a Microsoft Word outline or a set of PowerPoint slides; including shortcuts to files in a kernel view; and directing e-mail messages from a particular sender into a kernel representing that person.

The basic message here is that an individual uses a variety of applications to manage their knowledge effectively. And although a personal knowledge base should perhaps be the centerpiece of this toolset, it should work seamlessly with the other applications to carry out tasks. At the very minimum, Popcorn needs to be able to import and reference data stored in other tools, and it might also be advantageous to more tightly integrate with certain tools so that the knowledge it contains can be used to maximum effect.

#### 4.2.10 Strength and weakness: Lack of structure

If freedom and fluidity are Popcorn's greatest assets, it appears that they can also be its greatest liabilities. Several users (8) commented that they sometimes felt paralyzed by too many options: since Popcorn imposes no particular structure, the onus is completely on the user to generate their own, and this can be an intimidating proposition. It is often not clear at the outset which of several modeling choices will turn out to be the best. So the user must simply choose one, and proceed half-convinced, always wondering if they made a suboptimal choice. Later on, new knowledge may arise that shares features with something they previously recorded, but the user will inadvertently encode it differently than they did before. Such inconsistencies can be maddening when they are later exposed, and can actually lead to fear and uncertainty.

Part of the difficulty is the fact that much creativity is involved in encoding knowledge at all. One user echoed the thoughts of many when he said, "I'm very impressed with the way I see Popcorn working for others. But when I sit down to input something myself, I often just stare at that blank screen and don't know how to proceed." The flexibility itself, it appears, sometimes stifles thought.

There are several ways this could be addressed. One is simply through training. A brief "introduction to knowledge mapping through Popcorn" course could explain the principles of concept mapping, plus common techniques and tricks to materialize one's knowledge within the constraints of the tool. Another idea would be to introduce sample skeletal structures for particular domains. When modeling a new area, users could perhaps browse a central location (or even the Web, with a service discovery approach) to find "Popcorn starter structures" for download. These would contain basic patterns that have been proven successful by other Popcorn users in a particular field, which the user could fill in, imitate, modify, and extend. This would help bootstrap a beginning user by giving them some guidance as to which way to proceed initially.

#### 4.2.11 Reflection: The difficulty of knowledge formulation

The most significant barrier to using Popcorn, however, and also the most enlightening one, was simply this: knowledge formulation is a more difficult endeavor than either we or our users imagined. Our hope was that since the mind can be said to store knowledge as a concept map, allowing users to transcribe that structure directly into electronic form would be easy and painless. What we discovered, however, was that representing

knowledge can be a vexing task, not just when the interface is suboptimal, or when the data model is insufficient, but when *the knowledge we wish to record is not fully understood*.

It was abundantly clear from user experiences that Popcorn exposes ill-understood knowledge. Its strength is that it allows knowledge to be expressed as a cognitive structure; but the catch is that it essentially *demand*s the user enter it in that form. Time and again, users would be frustrated in trying to properly model their knowledge in the tool, and they often reported that their lack of understanding is what really stopped them. They couldn't figure out how to express the knowledge in terms of concepts and relationships because they didn't truly *understand* the concepts and relationships. The experience was frustrating because Popcorn confronted them with their lack of understanding and forced them to deal with it.

Users had two basic responses to these situations. They either gave up using the tool out of frustration, or else they took much longer than expected to record their knowledge. Those who took the first course were conscious that they did not understand the topic as well as they thought, and this frank realization became Popcorn's primary contribution for them. The others engaged in more time-consuming investigations and longer brainstorming sessions, after which they invariably emerged with a more thorough grasp of the domain. Popcorn's value for them was perhaps not that it stored the knowledge itself, but that it dispelled the illusion that they really understood it, and encouraged them to explore it further. A very common remark from users was that although it was time-consuming to construct a set of kernel views, in the end the process was well worth it.

## 5. CONCLUSION

User trials show that transforming information into knowledge is a time-consuming and demanding task. There are no shortcuts, and Popcorn does not offer any. Nevertheless, the results also show that Popcorn can be a powerful tool in the hands of a user who has lots of complex knowledge to deal with and is comfortable with abstract, free-form thought. Despite some weaknesses in realization, especially in the area of knowledge reorganization, the results suggest that our analysis of the personal knowledge management task is reasonably sound. In particular, transclusion appears to be a critical feature. If Popcorn were made ubiquitously available, and cooperative with other common information tools, it might well become a "killer app" for knowledge-intensive professionals.

## 6. ACKNOWLEDGMENTS

In addition to our marvelous test group, we wish to thank Edwin Eng and Nick Terkay for their valuable contributions to this project.

## 7. REFERENCES

- [1] The Mozilla Foundation. The Firefox Browser. Available at: [www.mozilla.org](http://www.mozilla.org), 2005.
- [2] Adar, E., Karger, D. and Stein, L.A., Haystack: per-user information environments. in *Proceedings of the Eighth International Conference on Information Knowledge Management*, (Kansas City, Missouri, 1999), pp.413-422.
- [3] Anderson, J.R. *Cognitive Psychology and Its Implications*, 3rd Ed. W.H. Freeman, New York, 1990.

- [4] Antonietti, A. Why does mental visualization facilitate problem-solving? in Logie, R.H. and Denis, M. eds. *Mental Images in Human Cognition*, Elsevier Science Publishing Company, Inc., New York, 1991, 211-227.
- [5] AquaMinds Software Corporation. NoteTaker 1.9. Available at: [www.aquaminds.com](http://www.aquaminds.com), 2005.
- [6] Bernstein, M., Collages, Composites, Construction. in *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, (Nottingham, UK, 2003).
- [7] Bush, V. As we may think. *The Atlantic Monthly*, 1945, pp.101-108.
- [8] Buzan, T. and Buzan, B. *The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential*. Plume Books, 1996.
- [9] Canas, A.J., Hill, G., Carff, R., Suri, N., Lott, J., Gomez, G., Eskridge, T.C., Arroyo, M. and Carvajal, R., CmapTools: a knowledge modeling and sharing environment. in *Proceedings of the First International Conference on Concept Mapping*, (Pamplona, Spain, 2005), pp.125-133.
- [10] Collins, A.M. and Loftus, E.F. A spreading-activation theory of semantic processing. *Psychological Review*, 82 pp.407-428,1975.
- [11] Conklin, J. and Begeman, M.L., gIBIS: a hypertext tool for exploratory policy discussion. in *Proceedings of the 1988 ACM Conference on Computer-supported Cooperative Work*, (Portland, Oregon, 1988), pp.140-152.
- [12] Davies, S., Velez-Morales, J. and King, R., Building the Memex sixty years later: trends and directions in personal knowledge bases. University of Colorado Technical Report CU-CS-997-05. Available at: <http://www.cs.colorado.edu/departments/publications/reports/docs/CU-CS-997-05.pdf>, 2005.
- [13] Dourish, P., Edwards, W.K., LaMarca, A. and Salisbury, M. Presto: an experimental architecture for fluid interactive document spaces. *ACM Transactions on Computer-Human Interaction*, 6, 2 pp.133-161,1999.
- [14] Dumais, S.T., Cutrell, E., Cadiz, 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 the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, (Toronto, Canada, 2003), pp.72-79.
- [15] Gentner, D. and Stevens, A.L. (eds.). *Mental Models*. Lawrence Erlbaum Associates, Inc., New Jersey, 1983.
- [16] Halasz, F.G., Moran, T.P. and Trigg, R.H. NoteCards in a Nutshell. *ACM SIGCHI Bulletin*, 17 pp.45-52,1987.
- [17] Hog Bay Software. Hog Bay Notebook v3.5. Available at: [www.hogbaysoftware.com](http://www.hogbaysoftware.com), 2005.
- [18] Jones, W.P., The Memory Extender personal filing system. in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (Boston, Massachusetts, 1986), pp.298-305.
- [19] Kaplan, S.J., Kapor, M.D., Belove, E.J., Landsman, R.A. and Drake, T.R. Agenda: a personal information manager. *Communications of the ACM*, 33, 7 pp.105-116,1990.
- [20] Kintsch, W. *Learning, Memory, and Conceptual Processes*. John Wiley & Sons Inc., 1970.
- [21] Logie, R.H. and Marchetti, C. Visio-spatial working memory: visual, spatial, or central executive? in Logie, R.H. and Denis, M. eds. *Mental Images in Human Cognition*, Elsevier Science Publishing Company, Inc., New York, 1991, 105-115.
- [22] Marshall, C. and Shipman, F. Spatial hypertext: designing for change. *Communications of the ACM*, 38, 8 pp.88-97, 1995.
- [23] Miller, G.A. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63 pp.81-97,1956.
- [24] Nelson, T.H. The heart of connection: hypermedia unified by transclusion. *Communications of the ACM*, 38, 8 pp.31-33,1995.
- [25] Novak, J.D. *Learning, Creating, and Using Knowledge*. Lawrence Erlbaum Associates, Inc., Mahwah, New Jersey, 1998.
- [26] Novak, J.D., The theory underlying concept maps and how to construct them. Available at: Available at <http://cmap.coginst.uwf.edu/info>, 2003.
- [27] Omni Development, Inc. OmniOutliner 3. Available at: <http://www.omnigroup.com/applications/omnioutliner>, 2005.
- [28] Selvin, A.M. Supporting collaborative analysis and design with hypertext functionality. *Journal of Digital Information*, 1, 4,1999.
- [29] Sowa, J.F. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley Publishing Company, 1984.
- [30] Tulving, E. Subjective organization and effects of repetition in multitrial free-recall learning. *Journal of Verbal Learning and Verbal Behavior*, 5 pp.193-197,1966.
- [31] Wolber, D., Kepe, M. and Ranitovic, I., Exposing document context in the personal web. in *Proceedings of the 7th International Conference on Intelligent User Interfaces*, (San Francisco, California, 2002), pp.151-158.
- [32] Woods, W.A. What's in a Link: Foundations for Semantic Networks in Brachman, R.J. and Levesque, J. eds. *Readings in Knowledge Representation*, Morgan Kaufmann, 1985.