Improving the expressiveness of navigation task to evaluate users’ interest for pages

Bruno Merlin
Universidade Federal do Pará
TV. Dom Romualdo Coelho, 829
Belém, PA, Brazil – 66055-190
brunomerlin@ufpa.br

ABSTRACT
Social bookmarking is a popular way to share and publish bookmarks. The growth of the social bookmarking community is creating a parallel resource for web searches. In this paper we propose an approach we recently presented, called implicit social bookmarking, that creates a bridge between search in social bookmarks and classical web searches initiated in a search engine. Then we propose an additional user interface to improve the concept. Our initial approach allowed a user to implicitly both contribute to a social bookmarking system and benefit from other user searches, therefore improving web searches. The contribution to social bookmarking was based on the user’s navigation analysis and an automatic evaluation of the user interest for pages. The additional interface offers a tool that helps the user to revisit pages at short, medium and long time. The use of the tools is supposed to highlight the interest for pages and turn the navigation more expressive in order to improve the implicit social bookmarking.

Author Keywords
Social bookmarking, collaborative filtering, tagging.

ACM Classification Keywords
H.5.3: Information interfaces and presentation (e.g., HCI): Collaborative computing.

INTRODUCTION
Since its creation, one of the main stakes of Internet is its capacity to guide the user toward pertinent links and contents. Several generations of research and indexation algorithms followed one another to support this task. These successive algorithms enabled the growth of published contents, but, also fought against marketing techniques that abuse of indexation mechanisms to artificially increase the visibility of contents. Among these algorithms, PageRank was the first implicitly collaborative. This algorithm analyses the link contained in a page and increases the rank of the targeted sites in relation to the semantic content of the site. Thus, by the publication of its content, a site contributes to the evaluation of other sites. The sum of every contribution fixes the PageRank for a page and, reciprocally, a page publication is a contribution to the evaluation of other pages.

In addition, explicit ways to share links and opinions about contents have increased their popularity in the last years. Among them, recommenders [12] determine a user profile in function of user’s visited links, purchased objects, downloads, etc., and suggest other content related to the user’s profile. Another popular concept is social bookmarking [5], which consists in explicitly tagging and sharing contents and references. This is usually done on web-sites like del.icio.us, BlogMarks, Flickr, and YouTube.

In this paper, at first, we describe the approach based on recommenders and social bookmarking that are combined to improve the indexation of web pages and improve the results provided by classical search engines. Our approach is called implicit social bookmarking [24]. We present it alongside our first implementation (DJINN) and a first evaluation of the concept. But, despite interesting results in an experimental context, we observed that in normal situation of web search, the navigation is not so expressive. For instance, several tools such as historic or bookmarks are rarely used. Consequently the page evaluation remains mainly based on the time and times the page is visited and raises an excessive number of implicit social bookmarks.

Thus, in a second part we present DJINNI, an interface aiming at turning the user’s navigation more expressive. DJINNI combines historic, bookmarks tools and provides new features in order to ease the revisit of pages at short, medium and long term. The implicit objective of DJINNI is better integrating these tools in the user ambient in order to encourage the user to use it.

The rest of this paper is organized as follows. In the next section, we will discuss the concept of social bookmarking as well as associated research on the topic, and tackle the problem of spamming in social bookmarking discussing the conditions that can make social bookmarking a pertinent data source to improve web searches. Then, we introduce the concept of implicit social bookmarking, which is followed by a section about DJINN, our implementation of
IHC2010 – Anais Estendidos

this concept and a brief conclusion of these first works. At
last we present DJINNI and discuss the impact of the
interface on the user’s navigation toward a first evaluation.

SOCIAL BOOKMARKING

Social bookmarking [5, 8] enables users to store, manage
and share bookmarks, and are classified and organized by
tags. Web sites, such as del.icio.us or BlogMarks, collect
the bookmarks and offer tools to perform bookmarks
searches in the collected tags.

Several previous researches [8, 9] dedicated their efforts to:
decrease the cost of different tasks such as tagging and
bookmark publication; propose models to share tags; and
models to build and share standardized vocabularies
(folksonomies [11]) for the tags. Other works studied the
social impact of the tag clouds [10] (representation of the
folksonomies).

Social bookmarking publication is nowadays well
integrated into the user environment through links enabling
one to tag and/or share the page directly integrated in the
targeted page. Nevertheless, research in a social bookmark
database (directly in a social bookmarking web site or
through tag clouds) remains a step independent of a search
by a standard way (classic search engine).

Reliability of shared links

The main question that drives this research is the following:
can we improve web search using social bookmarking? Heymann and colleagues [2] explained that the relevant
quality of links and tags provided by a web site such as
del.icio.us enables one to provide pertinent links; and
sometimes, to provide pertinent links not found during a
classical web search. However, they also observe that
del.icio.us covers only a small part of the Web. According
to these authors, social bookmarking has a large potential to
grow. Furthermore, social bookmarking is maintained by its
communitarian usage. By that we mean, that the usage
relies on a relationship of trust between one publisher and
several consumers of the published information (1 toward
N). In a more open context of use, we must consider that
social behavior [1] or economical interests (e.g. spamming
[3]) may impact the reliability of suggested links. To
preserve this reliability, [27] suggests way to establish a tag
author “trust rank”.

It is easier to artificially highlight and promote contents in a
relationship publisher toward consumers. In fact, because
of it, many researchers identify the fight against spam in
social tagging as a main stake for the social bookmarking
issue [3, 4]. To benefit from a systematic use of social
bookmarking to improve our web searches, it is necessary
to reduce the weight of spammers by equilibrating the
number of publisher and consumers and turning the action
of more systematic tagging.

Our hypothesis is that concepts like social navigation [5]
and collaborative filtering can augment social bookmarking
approaches to equilibrate this relationship “1 toward X” and
enable recommendations “several consumers toward
several consumers.” Of course, we are strongly convinced
that the problematic “who work, who’s the beneficiary?”
[14] is the central question. Tagging is time consuming and,
beside the context where the tagger is the principal
beneficiary [6], it requires a social or economical
motivation [1] to devote time to do so. Consequently, two
concrete options to increase the number of taggers and
reduce the weight of spammers are: (i) to reduce drastically
the time necessary to tag; or (ii) to increase the benefit for
the tagger.

At first sight, even if we reduce the tagging time, it seems
not very realistic to obtain an active collaborative filtering
[7, 13] by asking every consumer of the service to
systematically evaluate and tag the result of his searches. In
this article, we want to promote the idea that we may obtain
more taggers by passive collaborative filtering and propose
the concept of implicit social bookmarking to tackle this
issue.

OUR APPROACH: IMPLICIT SOCIAL BOOKMARKING

We propose the concept of implicit social bookmarking
aimed at improving web searches. This approach is based
on social bookmarking and passive collaborative filtering
[13] used by recommenders. When the task of social
bookmarking needs specific actions from the user, the
implicit social bookmarking:

- Automatically deduces information required for social
  bookmarking. This information must be extracted from
  the user’s interaction with the browser during other
  navigation tasks (web search, visit/re-visit of pages,
  personal bookmarking etc.); and

- Automatically provides information from social
  bookmarking without requiring an explicit request
  from the user.

Each one of these aspects is discussed below.

Deducing information from user’s behavior and interaction

Two types of information must be inferred from the
interaction between the user and the browser: an evaluation
of the user’s interest for the pages he visited; and a
semantic context (a set of tags) needed to tag and bookmark
these pages.

The user’s interest for a page can be inferred by several
users’ comportments. For example: a long consultation of
the page, the re-visit of the page at a different time [], the
visit of the page links, the addition of the page in the
bookmarks or else the forward of the page to other users,
are all gradual ways to express an implicit interest for the
content present in the page. Of course, all these approaches
are somewhat limited, for instance, time visiting a page
might be artificially increased because the user left the
browser open for a coffee. In any case, we believe that by
combining multiple sources of information (time, frequency of re-visits, bookmarking, etc) we can reduce possible errors.

Meanwhile, different data sources may contribute to feed a semantic context. At first, the input information by the user during its searches may be the most pertinent source: vocabulary input in search engines, words used to retrieve specific topics in a page, etc. However, we must filter this information for orthographic errors and auto-correction from the user. At second, the description words provided by visited pages (title and *keywords*) can help to characterize a search. At last, by a lexical analysis, the proper content of the pages may contribute to feed the semantic context.

The tuple “pertinence/tag/page” for every user must be collected to create a bookmark database. Every user may contribute to the page evaluation with the same weight to attribute a global mark to page toward a vocabulary. The database is then created by an implicit collaborative way.

Providing information from social bookmarking
Currently, searches among social bookmarks and searches using search engines are performed independently. A user request does not provide results simultaneously from social bookmarking and from a search engine. The second aspect of *implicit social bookmarking* is to combine these two results, for example, by using the social bookmarked data to reorganize and highlight the results provided by search engines. This aspect is better described in our implementation below.

DJINN: AN IMPLEMENTATION OF IMPLICIT SOCIAL BOOKMARKING
DJINN is an implementation of the *implicit social bookmarking* concept. First, it automatically analyzes users’ navigation and interaction to retrieve information about users’ interest in a page. Second, it automatically extracts semantic information from the user’s navigation to tag the interesting pages. These two pieces of information are used to create tagged bookmarks that are shared between users. Lastly, it augments the search results made in classical search engines (*Yahoo* and *Google*) with information from shared bookmarks (cf. Figure 2).

DJINN Structure
DJINN is composed by a client part and a proxy shared between the users (Figure 1). The client part is itself divided in two modules communicating through the file system: a Firefox add-on and an analysis module (the “context manager”).

Providing information from social bookmarking
Currently, searches among social bookmarks and searches using search engines are performed independently. A user request does not provide results simultaneously from social bookmarking and from a search engine. The second aspect of *implicit social bookmarking* is to combine these two results, for example, by using the social bookmarked data to reorganize and highlight the results provided by search engines. This aspect is better described in our implementation below.

DJINN: AN IMPLEMENTATION OF IMPLICIT SOCIAL BOOKMARKING
DJINN is an implementation of the *implicit social bookmarking* concept. First, it automatically analyzes users’ navigation and interaction to retrieve information about users’ interest in a page. Second, it automatically extracts semantic information from the user’s navigation to tag the interesting pages. These two pieces of information are used to create tagged bookmarks that are shared between users. Lastly, it augments the search results made in classical search engines (*Yahoo* and *Google*) with information from shared bookmarks (cf. Figure 2).

DJINN Structure
DJINN is composed by a client part and a proxy shared between the users (Figure 1). The client part is itself divided in two modules communicating through the file system: a Firefox add-on and an analysis module (the “context manager”).

The Firefox add-on is developed in JavaScript/XUL. It collects data such as: interaction with the tabs (tabs opened, closed and focused); visited pages and interaction used to visit pages (text input of the page address, click on a page link, bookmark, menu to open a page in a different tab etc.); text input in search fields (fields named “search”, “q”, “query”, “recherche”, “pesquisa” etc. contained in the page forms); text input in search bars and Firefox field used to search a word into a page; use of bookmarks and historical data; and, at last, the title and *keywords* of visited pages.

The Context Manager is developed in Java. It collects and analyses data provided by the Firefox add-on to build *search contexts*. Three algorithms enable: to determine the beginning and the end of a search and the tabs and windows involved simultaneously in the same search; to mark between 0 and 10 the pertinence of the visited pages (according to the user interaction); and to select between 0 and 7 words to qualify the semantic context of the search. The context manager transfers the triplets page/tags/mark to the proxy.

The proxy, equally developed in Java, collects the triplets provided by different clients in a database and maintains a global mark for the association page/tags. A unique mark is given by a user for a pair page/tag. Nevertheless, this mark can be updated during further searches. In addition, the proxy propagates the HTTP requests and responses toward common web sites without impacting them. But, it enables a specific treatment to handle the HTTP requests performed toward search engines (nowadays *google* and *yahoo*). The proxy not only retrieves the first result page. It retrieves a maximum of result pages during 1.5 second. The time has been chosen arbitrarily and enables to receive several results pages (an average of 8 pages) without delaying a lot.
the response to the user. In parallel, the proxy use the vocabulary used in the request to retrieve bookmarks in its database. The results provided by the search engine are then reordered, eventually completed and highlighted (Figure 2) in function of the mark of the bookmark contained in the database.

**EVALUATION**

We conducted a first evaluation of the concept in a controlled environment. The objective of the evaluation was to determine if the use of implicit social bookmarks could reduce the search time for different users doing successively a search on the same theme. This usage context targets different users working in the same team and frequently tackling the same search subject (for example, researchers from the same lab or students doing a collaborative homework).

Five users from the same research team participated in the experiment. In a first session, they conducted 5 short searches (about 15 minutes each for the first user) on 5 different themes: 3 scientific themes and 2 historic topics. They had to retrieve specific information about each theme and provide a copy of three links reporting this information. They performed the same exercise a week later.

The users easily trusted the link implicitly recommended and highlighted by the proxy in the search results. The number of different visited pages and the time required to retrieve the information decreased rapidly in function of the number of searches done on each theme. The time to perform the task became very low during the second session (Figure 3), since users retrieved the links leading to the pages selected during the first session in the first page of results.

**SYNTESIS**

The concept of implicit social bookmarking enables to integrate the social bookmarking task into the other users’ activity without requiring specific users’ actions. By his interaction with the browser, a user implicitly informs the service of his interest for a page and provides semantic information to qualify the context of this interest.

In return, the user benefits of the collective contribution through additional information integrated to his search results. The neutrality of this additional information is preserved because: every system’s user contributes to social bookmarking; every user gets a limited impact on page mark (a mark for each user consulting the page).

**LACK OF EXPRESSIVENESS OF THE NAVIGATION**

In a second pre-experimentation step, we shared the system between the research team members during six weeks. Then we studied the triplets tag/page/mark recorded in the bookmark database. We selected randomly a set of triplets and analyzed this set with the system users.

We observed that the system enabled to raise pertinent links. But, the number of links by tag was high and the classification between these links (the mark) was not always very pertinent. The results were not so optimistic than the result observed during the first experiment in a controlled context.

Two reasons justified these differences. At first, the task proposed during the first experiment implicitly suggested the words used by the users to start their searches. Consequently, the users started their searches by the same requests, visited a lot of similar links and contributed by the same way to mark the pages. It contributed to reinforced the
discrimination between the links and improve the classification. But, most of all, the task encourages the users to visit and revisit the pages and to use bookmarks and history tools. The use of these tools increased a lot the pages marks.

We noticed that, in a non controlled context, rarely the users accessed to a page by the same way, but above all, bookmarks and history are rarely used [15, 16]. Consequently, during the six weeks of pre-experimentation, the navigation were less expressive than during the controlled context phase and do not enable to completely rich our expectations.

Consequently, these conclusions raised a determinant question: why the bookmarks and history tools are rarely used and how could we improve their use in order to turn the user’s navigation more expressive.

**IMPROVING BOOKMARKS AND HISTORIC TOOLS**

Several projects chased the goal to improve bookmarks and historic tools.

The first axis of research focused on the representation of historic and bookmarks. *WebView* [17] and *Browsing Icons* [18] improved the reuse of web pages history by representing each search as a graph of links pointing on visited web pages. *Web Forager* [19] suggested representing the bookmark as a “library” and storing the page in “books”. *PadPrints* [20] and *WebView* [16] organized history as a navigation tree and associated each page with a page images to facilitate the remembering process of previous researches. These three representation families brought a new way to organize and structure history meliorating the user appropriation of the history. *PadPrints* and *WebView* introduced too an interesting support for link identification by associating a page image to each node.

But, these different representations remains confronted to the same problem. They have to be displayed in a large space to remain efficient in a complex revisitation sequences. In addition, our preliminary studies illustrates that the identification through different page images is not so easy, especially if graphical charts of pages are similar (for example, distinction between pages extracted from a same web site).

The second axis of research focused on data grouping. [21] connects links by an automatic identification of data connectivity. *PageLinker* [22] groups connected links by a simple user interaction and suggests contextual bookmarks during the user navigation. If this solution does not target short term revisitation, it’s a powerful tool to reduce the interaction cost of researches in bookmarks. *SearchBar* [23] propose a powerful tool to improve the searches across the history of different application.

Re:Search engine [28] observed that the user frequently redo the request into a search engine to retrieve a page previously visited [29]. Whereas between two searches, the results displayed by a search engine may change. To ease the re-visitation of these pages by exploring the user’s memory, Re:Search maintains the position of the visited pages into the search engine results.

**DJINNI IHM**

Consequently, according to the other project results, an efficient tool for short/medium/and long term page revisitation would be a good way to increase the use of bookmark and history and, thus, to turn the navigation more expressive. But the question of the tool integration remains decisive. Based on the state of art analysis, we proposed an IHM: DJINNI, oriented by the following guidelines:

- The links must be directly accessible in the user environment and the design must resolve the contradiction to dedicate a large space to the link representation without reducing the web page surface;
- The cost of interaction to access to the link, to store and manage the bookmarks must be very low;
- The design must help the user to have a better appropriation of its revisitation context;
- The tool must anticipate the user’s needs and suggest appropriated bookmarks.

**Concept for short term page revisitation**

Such as *WebView*, for each tab or window, *DJINNI* dynamically builds a navigation graph/tree during the web pages exploration.

But, instead of dedicating a specific area to the graph display, *DJINNI* displays it in a window without background, positioned above the others (cf. figure 4).

A header enables to move the graph in the screen space. Each visited page is represented by a bullet. A circle around the bullet highlights the current active page. A new visited page appears as a son of the current active page in the tree.

The page link is displayed as a tooltip when the mouse flies over the bullet (and every link is displayed by a mouse middle button pressure on the graph header).

**Figure 4: DJINNI graph displayed in a transparent canvas**

In order to guide the page revisitation and to ease the user’s appropriation, *DJINNI* enables the user to personalize the graph by adding visual mnemonic supports (cf. figure 5). A simple pressure on the F2 key (with the focus on the web browser or on the *DJINNI* Window) substitutes the page bullet by a page image of the current visited web page.
Each bullet and page image is interactive (cf. Figure 6) and enables to revisit a link by a simple click.

Three paradigms are supposed to make DJINNI an interesting alternative to short term revisitation tools:

- Giving direct access to all link visited during the search;
- Progressively structuring the data in function of the user activity to ease the memorization process;
- Increasing the user appropriation of contents by a simple customization.

Naming and saving the tree

Naming a bookmark gets an important interaction cost. So, in general, the bookmark tools suggest a default name based on the page address or page title.

DJINNI does not exactly manage bookmarks whereas it manages the *implicit social bookmarks* made up during the users’ searches and organized as a hierarchical and personalized tree. DJINNI reuses the tagged compute by the *implicit social bookmarking* algorithm to name and save the whole graph, the search context. The tags are grabbed during the user search and based on the user’s input, the visited page titles and keywords [24].

However, DJINNI allow too the user to modify the tags. Its shows the current name for the context and enable to add or remove words from the name and suggest a list of alternative words (cf. Figure 7).

The trees are automatically stored at the end of a search. The end of a search is automatically determined in function of several criteria: tabs closed, new search in a motor engine with a complete different vocabulary etc. (cf. [24]).

**Assistant for restoring contexts**

To encourage the user to use the search contexts previously saved during further researches, we had to decrease the interaction cost for the action to retrieve a context. The use of these search contexts is not spontaneous for a user and it is difficult to imagine that a user will look for a context previously saved before starting a new search. Consequently, DJINNI has to suggest search contexts and incites the user to restore it during the search.

Once again we used the implicit tag based on data typed in search engines, page titles and page addresses to suggest pertinent saved contexts. The titles of stored contexts that match with the current implicit tags are highlighted (cf. figure 8). The user can substitute the current context by a suggested one (click right on the item) or graft a contexts to the current one (click left on the item).
two search contexts. Branches of graph can be cut or collapsed to make the information displayed more relevant.

![Figure 9: cleaning up the tree to keep only the relevant data](image)

The selection of graph nodes was an important question to solve. Selecting each node one by one was not acceptable. In another hand, we could not interact directly on the background of the DJINNI window to make lasso selection or marking menu for the simple reason that there was no background. The solution has been to give a background to the window the time to make the selection.

A pressure on the Ctrl+E key engenders a capture of the whole screen. Temporarily, the window background is restored and this capture was put as window background. This capture fakes that the window remains without background but enable the marking menu interactions (cf. figure 10 and 11). The process is performed if the focus is on the web browser window or if the focus is on the DJINNI window.

![Figure 10: By pressing Ctrl+E, the windows background display a copy of the screen and turn interactive](image)

**EVALUATION**

We conducted a first experimentation to measure the impact of DJINNI on user’s navigation. We compared the navigation tools used with and without DJINNI in order to determine if the tool is useful and able to improve the expressiveness of the navigation to turn implicit social bookmarking more efficient.

**Procedure**

The experimentation involved 10 users (all master/PHD student or professors from two research teams) during two weeks (five day per week). The users are all Firefox users. The experimentation occurred in the normal users’ professional ambient. The Firefox version used during the experimentation was 3.6.3 (the version includes an intelligent search system when entering the beginning of an address into the address bar).

During, the first week, DJINNI was not displayed. The Firefox plug-in automatically collected the users’ habit and particularly how users revisited the pages (button previous/next, input of the address, bookmarks, fast-dial etc.).

During, the second week, DJINNI was displayed. We collected the same information and collected the user interaction with DJINNI.

**Result**

*Substitution of previous/next buttons*

During the first week, the previous and next buttons were the tools mainly used to revisit pages at short term. The combo-box proposed to retrieve a page visited several page before / after the current page was rarely used. Instead, the users performed several click on the next / previous button and indirectly visited briefly the intermediary pages.

During the second week, the navigation graph progressively substitutes the use of the previous/next buttons. The two last days, the use simple of the buttons decrease of 70% and the use of the combo-boxes was abandoned and completely substituted by the interaction with the graph. Even if for
some users (2 on 10), the use of next/previous remains the main reflex and the size of bullets were considered globally a little too low, the graph was as a good reference to represent the context and memorize the visited pages. Moreover the tree representation enables to access to more link that the simple previous/next list.

By suppressing the several consecutive clicks on next and previous buttons, the tool reduces the number of revisited pages and improves the page evaluation algorithm.

Use of tree personalization
The efficient interaction to personalize the tree nodes with a page image encourages the users to use it. Initially attracted by the originality of the representation, the users used it intensively during the first day. Then the use decreased and stabilized (an average of 12 personalized nodes per users and per day the three last days with a large variation inter-individual). The nodes personalized represented a high percentage (about 85%) of the nodes revisited during further searches.

The use of this personalization increases efficiently the identification of pertinent links and would contribute deeply to improve the implicit social bookmarking.

Bookmark tools and reuse of previous stored context
During the first week, the main tools used to revisit the pages at medium and long term were:

- At first: tools such as ‘Fast Dial’ (page shortcuts displayed as home page for the navigator) and links displayed in the personal toolbar of the navigator. These links contain the pages that the user daily visits such as web-mails, newspapers, users’ site, bank, search engine, social network sites, etc.
- At second: the users retrieved pages by entering the beginning of address names in the address bar, or reinitiating the search in the motor engine.
- The bookmarks tools were rarely used.

The DJINNI was not efficient to substitute the preferred site tools such as ‘fast dial’ and the personal toolbars. It was even judged intrusive for sites such as web-mails and newspapers because a user rarely revisits articles and mails and then do not need to remember is former searches into these kind of sites.

However, DJINNI was very efficient when the user revisited pages by entering an address or reidid its search in Google. In these circumstances, DJINNI was able to suggest an existing context during the address or keywords input. 8 of the 10 users frequently restored the suggested contexts when a context was suggested. It enabled to reduce the number of key pressures and click to revisit a page of about 40%. DJINNI has been particularly appreciated when the user needed to revisit several pages in the same context.

The restoration of previous contexts reduced the number of revisited pages and thus contributed to revisit only the pertinent ones. This property enabled to increase the pertinence of the page evaluation.

CONCLUSION
At first, we resumed the concept of implicit social bookmarking that we lately proposed. The concept of implicit social bookmarking enables to integrate the activity of social bookmarking in the other user’s navigation task. By interacting with his web browser, the user implicitly informs the social bookmarking system about: his interest for web pages, and the semantic context of this interest.

But we observed that the navigation task was not always expressive enough to evaluate and compare efficiently the users’ interest for different pages. To improve the expressiveness of the navigation, we designed a HMI: DJINNI. DJINNI merges the short, medium and long term revisitation tools offered by browsers and incites the user to use it. By its interaction with the IHM, the user increases the expressiveness of it navigation turning more efficient the implicit evaluation of the interest for a visited content.

Our first experimentation shows that the tools were easily adopted and could contribute to improve the efficiency of implicit social bookmarking with DJINNI. But, moreover, the DJINNI appeared as an interesting and efficient stand-alone application substituting several revisitation tools.

The experimentation raised observations that will contribute to improve the HMI. These observations will lead us to improve DJINNI as stand alone application and as implicit social bookmarking contributor. Then we plan to evaluate DJINNI in a wider context.

ACKNOWLEDGMENT
We thanks Cleidson de Souza for his contribution to the elaboration of the implicit social bookmarking concept; and the students and researchers of the Institut de Recherche en Informatique de Toulouse to there participation to the experimentations.

REFERENCES
11. http://vanderwal.net/folksonomy.html
23. Dan Morris, Meredith Ringel Morris, Gina Venolia, SearchBar: a search-centric web history for task resumption and information re-finding, Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, April 05-10, CHI 2008, Florence, Italy.