

Tag-Based Metonymic Search in an Activity-Centric Aggregation Service

Michael J. Muller, Werner Geyer, Beth Brownholtz, Casey Dugan,
David R. Millen, and Eric Wilcox

IBM Research

*{michael_muller, werner.geyer, beth_brownholtz, cadugan, david_r_millen,
eric_wilcox}@us.ibm.com*

Abstract. Knowledge workers often need to find, organize, and work with heterogeneous resources from diverse services, information stores, and repositories. This paper analyzes two problems that knowledge workers frequently encounter: difficulty in finding all relevant resources across diverse services, and difficulty in formulating and executing searches for resources related to their current activity-of-interest. The Malibu project explores solutions to these problems through a dynamic peripheral display that aggregates knowledge resources from multiple services to support activity-centric work. Of particular interest is the ability to select a knowledge resource and use it as a metonym (a proxy) for its social-tagging metadata in a tag-based search for related resources among heterogeneous services. We evaluated our solutions to these two problems through convergent analyses of quantitative (data log) and qualitative (interview and discussion data) data. Our partial successes show the strength of these new ideas, and indicate areas for future research.

Introduction

While working on a single, integrated activity or concept, knowledge workers often need to find, organize, and save heterogeneous resources. Using the concepts of activity theory, Bardram and colleagues have studied the assembly of diverse resources in medicine (Bardram, 2005). Others have noted the heterogeneous resources needed for knowledge work in activity management in offices (Bellotti et al., 2003; Ducheneaut and Bellotti, 2001; Halverson et al., 2004; Kaptelinin, 2003; Moran et al., 2005; Muller et al., 2004; Whittaker, 2005). A number of projects

have explored systems for supporting knowledge workers in assembling and using such heterogeneous resources (Bardram, 2005; Bellotti et al., 2003; Geyer et al., 2006; Muller et al., 2004).

During the same period, researchers have described knowledge workers' issues with interruptions and interruption management. Knowledge workers frequently change tasks, either by choice or through interruptions (González and Mark, 2005; Iqbal and Horvitz, 2007; McFarlane and Latorella, 2002). Interruptions are commonly blamed for impairments in attention, task completion, and quality of work (Cutrell et al., 2001; Hudson et al., 2002; Monk et al., 2004). Several systems have intended to support knowledge workers by managing interruptions and/or by preserving or restoring context (e.g., Sen et al., 2006; Sullivan, 2006; Whittaker, 2005).

Organizing and Using Activity Resources

Several research programs have explored support for context switching and resource rediscovery by organizing and integrating resources, tools, and people around the computational concept of a work activity (e.g. Bardram, 2005; Bellotti et al., 2003; Geyer et al., 2006; Gwidzka, 2002; Kaptelinin, 2003; Muller et al., 2004). Many of these approaches have in common that they provide some structure within which all records of an activity may be collectively located and (re)discovered.

In our research, an “activity” is a structured set of diverse objects that are shared among specified collaborators to accomplish a shared task (Geyer et al., 2006; Moran et al., 2005; Muller et al., 2004). An activity can begin with something as simple as a chat or an email, shared between two people, and can grow into structured collections of over a hundred files, messages, chats, links, etc., that are shared among dozens of users, and that have lasting corporate value long beyond the time of the last update to any of the objects in the activity (Muller et al., 2004). Alternatively, an activity can be instantiated from a stored customizable template (Moran et al., 2005).

Representation Gap

Activity-centric computing provides powerful tools to support collaborative work. However, one problem with activity support systems is that the user is burdened with manually managing the resources and the structure of many activities simultaneously. Worse, despite the fact that the various activity-based approaches allow the collection of heterogeneous resources inside a single structure, users often must continue to store some information in multiple services – e.g., documents and tasks in an activity-centric store *plus* feeds in a feedreader *plus* bookmarks in a social-bookmarking system *plus* communications events or logs in email or instant messaging clients. As a consequence, much information that is part of the

cognitive model or otherwise related to the activity might not get captured or displayed in the formal representation (e.g., Bardram, 2005; Moran et al., 2005; Muller et al., 2004). We call this problem the **representation gap**.

Currently, as noted above, users have to manage multiple information stores. Many of these stores do not even have the ability to link to relevant resources in other information stores. This problem is further complicated by the fact that, in complex activity environments, new resources are always being added to the community's or enterprise's various storage services by colleagues. Users need to be able to find these newly added resources whose relationship with their current activity structures has not yet been determined. With today's tools, users must monitor multiple stores and services, including feeds from individual sources of information, and then must take manual actions to aggregate these diverse resources into their activities.

A few corporate services have experimented with socially-informed feeds, such as the ability to subscribe to individual users' bookmarks in Dogear (Millen et al., 2006) and the ability to receive alerts as colleagues added new information to team-based activities (Geyer et al., 2006; Muller et al., 2004; Sen et al., 2006). However, some of these experiments have led to a deluge of alerts with limited relevance to the user's actual needs (e.g., Muller et al., 2004), and this problem has led to additional research into ways of limiting those information feeds and subscriptions according to individual actions or social recommendations (Sen et al., 2006). The research in this paper intends to aggregate many of these sources of new information, and to use a single, user-tunable method to control and limit the information from these sources.

Metonymic Search

In order to satisfy their information needs beyond what's represented in an activity, knowledge workers typically use standard web or desktop search tools. However, this requires the user to interrupt their work and actively seek out information. Since today's search technologies do not take into account the user's current activity, users need to manually encode their information needs through appropriate search terms. Users typically try to identify key phrases that can be used in a search engine to find related resources. This strategy is labor-intensive, error-prone, and fails to take advantage of available metadata about the user's current activity or any object in general. Users would have an easier time searching if they could base their search on automatically derived attributes of the item of interest. In rhetorical terms, this could be called **metonymic search** – i.e., the use of the item itself as a metonym (a referent) for its attributes, and the execution of the search by specifying the item as a placeholder for its attributes.

Several schemes have been developed that follow the Superbook model (Egan et al., 1989) of (a) selecting an item of interest and then (b) invoking an operation that is typically called "more like this." However, these approaches generally

work only within their own domain (e.g., documents for Superbook, or webpages for Internet search engines), and are often procedurally opaque to end-users (e.g., the concept of vector-space cosine similarity in latent semantic indexing is difficult for most users to grasp). There appears to be no solution for our problem domain, namely helping users to cope with the representation gap by easily invoking a metonymic search for related items across multiple stores of heterogeneous resources. The “Malibu System” Section (below) provides examples of tag-based inter-service search, and the scenario of use (below) provides examples of how tag-based searches can help to resolve the representation gap, and how a metonymic approach to those searches can make them easy to initiate and manage. We used a stack architecture for all of our tag-based searches, which manifested in the user interface as the ability to return to previous states of our client. We believe that this design not only addresses the representation gap but also provides better support for recovering from interruptions since each stack represents a search context that can easily be restored. In order to support interruption management We we provided Malibu as a peripheral display that could be invoked and dismissed with a single gesture, leaving the user’s display with preceding, underlying work intact.

Summary and a Look Forward

We developed the experimental Malibu system to assist knowledge workers in their activity-centric work, with the goal of improving or overcoming the limitations of the current activity management approaches. Malibu aggregates and provides fast access to information from different data sources; it finds relevant information across data sources using social tags and person information; it allows the user to rapidly switch and restore contexts; and it helps users manage discovered resources by flagging them as tasks.

The organization of this paper is as follows. We describe the Malibu system, and we describe three issues that our design addresses to deliver value to our end-users. We describe a field trial, in which we provided Malibu for download by employees in IBM. We then present usage data (logs and interviews); these data allow us to test our hypotheses of Malibu’s value. Finally, we close with a self-critique and questions to be addressed in future work.

Malibu System

Malibu runs as a desktop side bar (“Malibu Board”) that slides out when users hover with their mouse at the left or right side of the screen.¹ Malibu provides peripheral access to and awareness of multiple data sources contained in a series of

¹ Malibu is a research prototype. A full version of the system would include appropriate accessibility features.

configurable views, each one displaying multiple instances of a single type of data, such as tasks, bookmarks, and activities (see Figure 1). We envisioned that the user would invoke Malibu with a mouse gesture, would be able to work with old or new objects there, and would be able to dismiss Malibu with a second gesture, returning to the pre-Malibu context of her or his work. As we will describe, some users preferred to work in this way, while others maintained a Malibu window open all the time (often on a secondary display), and yet other users complained that the gestural-display user interface was annoying.

Similar to the Google side bar (Google, n.d.), Malibu can be extended with new views and data sources. The system currently supports the following views and data sources: My Tasks (A), My Activities (B) from the Lotus Activities system (Geyer et al., 2006), Dogear Bookmarks (C) from the Dogear social bookmarking system (Millen et al., 2006), and My Feeds (D), a feed reader that supports RSS and ATOM.

We decided to implement Malibu as an extension to IBM's latest corporate instant messaging client Sametime 7.5 because the client gives us access to the people information for Malibu's search engine. All the aforementioned data sources (tasks, feeds, activities, bookmarks) have people information associated with them, i.e. an instant messaging client becomes an ideal launching pad for pivoting on people, e.g. search for all bookmarks of a certain user. There are also technical reasons why we used Sametime 7.5 as a platform. Sametime 7.5, based on Eclipse and Java, is designed to be an extensible framework. In particular, Sametime 7.5 includes an infrastructure for easy deployment of extensions (plug-ins). Since Sametime 7.5 has a large install base in IBM, it allowed us to easily target a large user group. People are usually more reluctant to install new applications rather than plug-ins to existing applications. The download web site of the Sametime 7.5 client also offered other plug-ins, so employees were accustomed to searching for new functionality there. Finally, the Sametime platform allowed us to leverage corporate directories and authentication resources.²

Malibu Views

Each view has a different set of features. Consider first the Activities view. As mentioned above, an activity is a shared structured collection of heterogeneous objects, assembled by members of a team, to accomplish a group objective; the objects in an activity may include documents, messages, files, and so on (Geyer et al., 2006; Muller et al., 2004).

² We explored possibilities to develop a Malibu-based people view, with enhanced capabilities. However, we thought it might be confusing to offer a second, seemingly redundant view of people ("buddies"), even if our second view had greater functionality than the conventional buddylist.

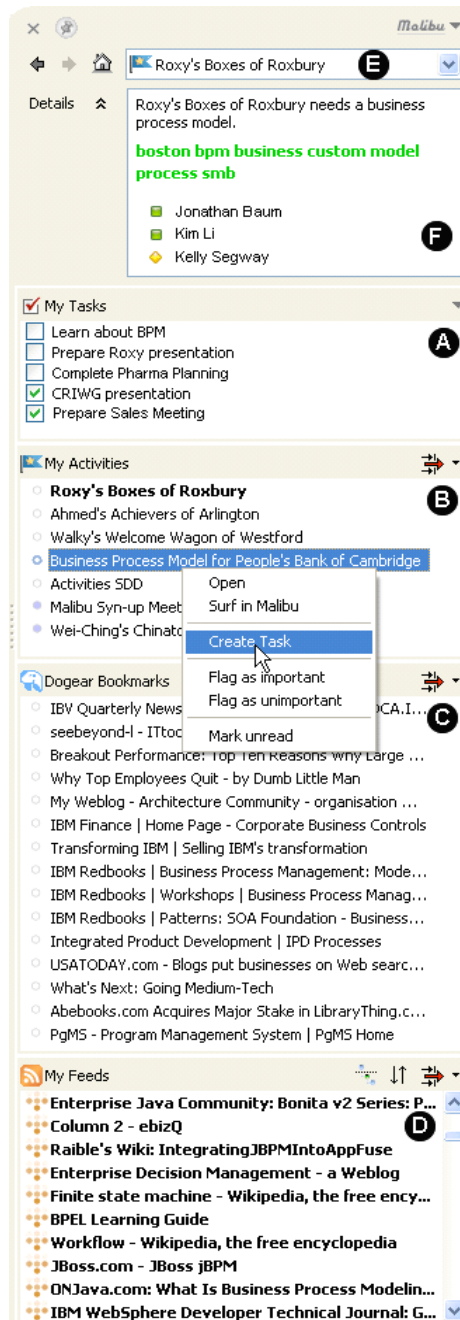


Figure 1. Malibu Board

Users of an activity (also known as “members”) often need to know the current status of the activity, or of one of the component objects within that activity. Malibu displays activities with updated information in bold, and sorts activities in terms of their relevance to the user. Malibu also supports user operations that are specific to activity management, such marking activities as important or unimportant. The Activities view also has filters to show only important and modified activities. This way, users can manually manage and prioritize their activities.

Similar filters are available in the Dogear view, which shows recent bookmarks from an enterprise social-bookmarking system (Millen et al., 2006).

In contrast to Activities and Dogear, the feeds view allows people to add or delete feeds, and to mark any feed as “read.”

We also implemented a task view that provides the ability for other views to designate any item in Malibu as a task (including a user-controlled completion indicator) with a pointer to the original item. The main purpose was to provide some capability for the user to flag (or bookmark) resources discovered during the tag-based search as follow-up tasks for later review.

Malibu Navigator

At the top of the Malibu Board, is the “Navigator” (E) which can be used to bring any Malibu item, including people from the instant messaging client, into focus, i.e. items can be selected as pivot objects on which Malibu performs a tag-based or people-centric search (“Surf in Malibu”).³ When a user pivots (“surfs”) on an item, views are reconfigured to display contextual information related to the current pivot object; which becomes the current focus of the board. For example, if the user pivots on a bookmark, the system uses the tags associated with that bookmark to execute tag-based searches for other Dogear bookmarks, activities, feeds, and tasks. If the user pivots on a person, the system uses that person’s name to search for bookmarks created by that person, activities in which that person is a member, and feeds that mention that person. The search can be refined by selecting individual tags or manually adding key words to the query. The details box (F) shows information about the current focus item, including the social tags from that focus item (which were used to perform the tag-based search in all of the views). Similar to a browser, the navigation buttons and the history drop-down menu can be used to restore the search results of previous pivot objects. Details such as tags, description and author information can be also viewed in a slide-out window by clicking on items in any view.

Extensible Tag-Based Resource Aggregation

Malibu thus provides views of several different types of data, including activities, feeds, shared bookmarks, and tasks. In practical business settings, users are likely to have a large number of these resources. As we noted earlier, the sheer number and diversity of these resources is likely to present problems to users. We described the representation gap as the difficulty in discovering all of the resources relevant to a current activity or task. Malibu assists with that problem through the

³ In our environment, tags are used to describe not only resources (Millen et al., 2006) but also activity records (Geyer et al., 2006), selected feeds, and people (Farrell and Lau, 2006).

use of automated tag-based search. Most of the resources in our environment have been tagged by users.⁴

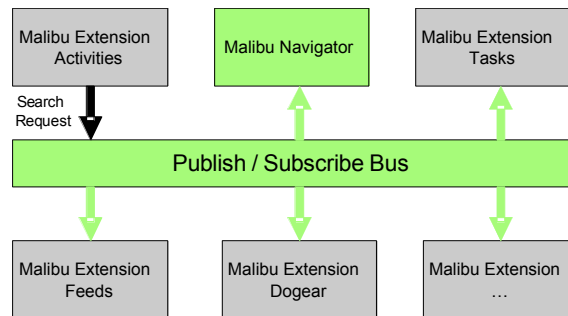


Figure 2. Malibu Publish / Subscribe infrastructure.

We designed Malibu to be extensible, i.e. developers can add new data sources and views to Malibu if the predefined set is not sufficient. Each Malibu extension implements the visual representation (view) and a search interface for tag-based retrieval that is registered with the Malibu Navigator component when the new Malibu extension is installed. In order to allow for a decoupling between the Malibu extensions (i.e. they do not need to know about one another), Malibu components communicate through an internal publish / subscribe system.

When a user executes a “Surf in Malibu” operation on an object in one of the Malibu views, the corresponding Malibu extension submits a search request to the publish / subscribe bus as illustrated in Figure 2 for the Activities Extension. The search request contains the metadata of the focus item: name, description, tags, and people information. Each Malibu extension can register with the Malibu Navigator to receive search requests from the internal publish / subscribe system. Malibu extensions receiving a search request, analyze the tags associated with the selected item, and conduct a tag-based search. Each Malibu extension can independently decide how to implement the search, e.g. whether or not to search the local cache versus issuing a request to a server, or both. Local search in our four out-of-the-box extensions is done through basic pattern matching of tags. The Dogear and the Activities plug-in also issue search requests directly to the servers of their data using REST APIs. Each Malibu extension updates its view with the search results ordered, for example, by recency of the item. The Malibu Navigator also receives the search request from the publish / subscribe bus and displays the search item as a focus item.

In the same way, Malibu is also aware of the identities of users (encoded in the search request), and extensions can conduct searches based on identity as well as on tags. For example, we added a buddy list context menu and a button into the

⁴ Tags are user-generated descriptive words or phrases that are, in general, visible to other users of a social bookmarking system such as del.icio.us (<http://del.icio.us>). For reviews, see Cameron et al. (2005); and Golder and Huberman (2006).

Sametime 7.5 chat window (see Figure 3) that allows users to trigger a “Surf in Malibu” action on their buddies. This also submits a search request to the publish / subscribe bus and Malibu extensions can perform a similar search on the buddy’s identity, showing items related to the searched person. Future versions of Malibu may make use of tags that have been associated with users, e.g., through an enhanced directory service that allows one user to write tags to characterize another user (Farrell and Lau, 2006).



Figure 3. “Surf in Malibu” action integrated in the chat window.

The tag-based and people-centric search are designed to help address the representation gap by finding relevant resources across heterogeneous resource domains and by providing sufficient data to evaluate the relevance of each returned object through tag-comparisons with the “surfied” object.

Thus, in a simple sense, Malibu acts as a resource aggregator by collecting views of heterogeneous resources into a single user experience. In a more substantive sense, Malibu acts as a sophisticated resource aggregator by finding and displaying related and relevant resources through automated tag-based search leveraging common, social metadata of objects.

Scenario of Usage

The following usage scenario illustrates some core features and demonstrates how Malibu can be used to support activity-centric work.

*Kim, the lead member of a customer care team, opens the Malibu Board to start her day. The Activities view (B) shows her activities filtered by importance. Kim sees that there is a new collaborative activity named **Roxy's Boxes of Roxbury**. It is marked unread (bold) and shows up at the top of the activities list. Kim clicks on the activity and a details view slides out. Kim sees that the date of the*

activity was yesterday afternoon; and the creator was Kelly from sales. The description says “Roxy's Boxes of Roxbury needs a business process model”. Kim knows that Kelly will expect to see Kim’s responses and relevant resources added to that activity. Because Roxy’s Boxes is an important customer, Kim knows that Kelly will be waiting anxiously for Kim’s response.

Kim right-clicks on the **Roxy’s Boxes** activity and chooses “Surf in Malibu” to find resources related to the **Roxy's Boxes of Roxbury** activity (i.e., an object pivot). Malibu analyzes the tags that Kelly has already associated with the **Roxy’s Boxes** activity (as shown in Figure 1, these tags are “boston, bpm [business process model], custom, smb (small-and-medium business market segment), and the components of the phrase ‘business process model’.” In this display, Malibu has alphabetized the tags, emphasizing that each one will have the same weight in searches. Kelly could have overcome this system default, if needed, by hyphenating the phrase “business-process-model.” Kim could overcome this system default by modifying the tag list before invoking the search.). Malibu conducts a tag-based search to find resources in Dogear bookmarks, Activities, and Feeds, that have been associated by other users with some or all of the same tags. Malibu views reconfigure their user interface to show resources related to the **Roxy’s Boxes** activity (E). The Activities view shows some related activities. Kim hopes to use one of these related activities as a template for meeting Roxy’s needs for a business process model. One activity that is now displayed, **Business Process Model for People’s Bank of Cambridge**, looks very similar to the new Roxy assignment, and might contain useful modeling information. Kim right clicks on the **People’s Bank of Cambridge** activity and selects “Create Task”. A new task linked to that activity shows up in the task view and Kim renames it as **Evaluate usability of BPM for Roxy’s**. Kim can use this task both as a memory aid and as a link to the resource that is the basis of the task.

An incoming chat from Mary interrupts Kim. Mary is asking for the planning documents for the “Pharma” project. Kim interrupts the current task, and the configuration of the Malibu board to support that task, confident that Malibu will be able to return to that configuration when the interruption is over. Kim right-clicks and surfs on Mary’s name in the buddy list (i.e., a person pivot). Malibu views use information about Kim and Mary to reconfigure, now showing all activities that Kim shares with Mary. The activity **Pharma Customer Planning** appears in the list. Kim had previously finished the planning document that Mary was asking for, Kim drags the document from her desktop onto the **Pharma Customer Planning** activity. As a result of the drag operation, Malibu automatically uploads the document as a new object in the **Pharma Customer Planning** activity. Kim replies to Mary in the chat, informing her that the document is complete and now available in the activity.

To recover from the interruption, Kim presses the back button. Malibu reconfigures into the pre-interruption state, showing Kim the context of her work before

Mary's interruption and the consequent reconfigurations that Kim had to make in order to work with Mary. Because Malibu preserves context, Kim can recover from Mary's interruption.

*Kim knows little about business process modeling in Roxy's business domain. She looks at the Dogear Bookmarks (C) and Feeds (D) views for related information. Kim notices a feed entry with a news article that could be interesting to all the members of her Roxy customer care team. She drags and drops the entry onto the **Roxy's Boxes of Roxbury** activity to share it with the team; Malibu creates a new object in that activity containing a link to the feed item. Kim also sees an interesting shared bookmark referring to a BPM tutorial she would like to look at later. She drags and drops the bookmark into the task view, creating a new task linked to bookmark. In this way, Kim prevents a self-interruption: She avoids having to read the BPM tutorial now, because she can create a link to it in a task that she can execute later. In the task view (A), Kim is reminded of the **Evaluate usability of BPM for Roxy's** task that she had created previously. She double-clicks on the task which opens up the activity the task is referring to.*

After reviewing the activity in the browser, she decides to take a new look at all her current activities. She presses the "home" button to restore overall work awareness again, clearing the previous focus on Roxy's. The Activities view shows all her activities newly filtered by importance, in keeping with updates and new resources that have been added since the last time Kim engaged in this overview.

Intended Benefits to End-Users

In summary we hoped that Malibu would improve knowledge workers' experiences in the following ways:

Issue 1. Representation Gap: Malibu should help users to view resources relevant to their current activity, even if those resources occur in diverse services or stores.

Issue 2. Metonymic Search: Malibu should support simple (one-click) operations to execute complex searches based on the attributes of an index item, across multiple services.

Issue 3. Recovery from Interruptions: Malibu should allow users to recover from interruptions by through a simple "back" operation to access previous searches and views. Also, Malibu's operation as a slide-out/slide-in sidebar should provide a useful peripheral display, providing and preserving work context when needed, but easily dismissed when not needed.

We explored these three issues in an eight-month field trial.

Malibu in Practice

We provided the Malibu client for download at an internal company website for early adopters of new technologies. Usage was entirely voluntary. As part of the installation procedure, we informed users that we would be collecting their data for study purposes. Users were at liberty to halt the installation if they did not want their data to be collected. The data reported in this paper come from a subset of the users who downloaded and installed the experimental Malibu client. The data were collected in three ways:

- Usage logs were automatically recorded from each user's client.
- Telephone interviews were conducted with thirteen users. A fourteenth interview was conducted via instant messaging, at the informant's request.
- A discussion database (internally called a "forum") was maintained for asynchronous dialogue among users and the members of the Malibu team. Ultimately, this database contained both informal comments, requests for assistance, and answers, and also a somewhat more organized survey process conducted by the company department that operates internal trials.

We will organize the presentation of results around themes, rather than around the sources of our data. We precede those results with a discussion of some of the issues in conducting this kind of large-scale, elective trial.

Description of the Sample

Offering Malibu for company-internal download by a diverse employee population was a good way to collect data and feedback, but it was also more difficult from a research perspective. Malibu was one of a large number of experimental plug-ins to the Sametime 7.5 instant messaging (IM) product, which was itself undergoing rapid beta-version development iterations during our trial. Company culture encouraged employees to download any and all of the plug-ins, whether or not the employee was using the services that were accessed through each plug-in. In order to filter out noise from users who had never used any of the services leveraged in Malibu, or one-time curious early adopters or beta testers, we focused our report of early experiences with Malibu on a subset of 57 users who participated in one or more of the services, and who made strong use of Malibu functionality.⁵ We consider these 57 users to be representative of usage in the kind of "ideal case" that is often involved in the first usage of an experimental system – i.e., a small, carefully selected sample of users who have motivation to use the new system.

Our sample of log records from 57 employees included women and men from 16 countries. Most were not part of the research organization. Users' jobs and

⁵ We excluded logs from members of the Malibu team (authors of this paper plus two summer interns), because we didn't want to treat our testing protocols and/or demonstrations as if they were real usage.

roles ranged from sales and distribution, to service delivery, to operations management, to training, to market intelligence analysts, to a researcher, and a variety of information systems responsibilities (e.g., architect, developer, designer). Users' clients ranged from small and medium businesses up to US federal government contracts. Roughly 20% were in IBM's consulting organization. Participants in the interviews were men and women from five countries, with a range of 0.5 to 20 years in the company, and a range of Malibu usage history of 1.5 to 8 months. Interviewees were chosen from the same pool as the employees whose logs we studied, so their backgrounds spanned a similar range. Comments in the forum were from a wide range of employees; the comments ranged in content from one-line "me too" messages to lengthy discussions of features and problems.

Issue 1: Addressing the Representation Gap

We designed Malibu to help users deal with resources scattered across multiple systems. One way to work on this issue was to examine how many times there was search content in more than one Malibu view for each of the 57 users. The lowest percentage of searches with content returned was 64.91 for the FeedsView. Thus, most of the views presented content most of the time. More than 70% of our sample received resources in at least two of the three views, and just over half of the sample received resources in all three views. On a statistical basis, there is evidence that Malibu was successful in reducing the representation gap for more than two thirds of our sample.

A second source of data about the representation gap came from interviews. While discussing "best" and "worst" features of Malibu, users spontaneously commented that

I use Malibu to level-set what I need to get done. I live and breathe in an interrupt-driven environment. I use Malibu to return me to sanity. About half the time I need to dive into the details of an activity or a task (business operations manager)

you could capture all the bits and pieces you are involved in... it's a good idea to have it captured somewhere, for future use or if an auditor walks in (procurement program manager)

Comparing Malibu with a similar web-based service, one user noted,

The level of integration among all the component [services], that plugged in this, is what gives it the edge. (communications manager)

Comments in the forum were convergent:

It is great because [of] the decision to integrate Malibu with [instant messaging] and Activities... (designer)

The concept "find everything related to the activity I'm working on", is very powerful (business operations manager)

Users were also critical of aspects of these features. Several users argued that we had not completed the integration work: Malibu allows users to view and open *existing* activities, but

You can't actually create a new Activity from... Malibu... you have to create in the web interface [to the Activities server], then refresh the Malibu UI to see it, and work with it. (business operations manager)

We were particularly intrigued with the critique that

The collaboration capabilities of Activities are largely ignored in Malibu... Ideally, I'd like to... PUSH those... results out to the members [of my team]... (business operations manager)

With regard to our first issue, Malibu has made some progress toward repairing the representation gap. The statistical results from the log analysis demonstrated that Malibu provides access to diverse resources from heterogeneous sources. User remarks in interviews and in the online forum indicate that people found those diverse resources useful. In some ways, the users understood our solution better than we did, and recommended ways to extend our solution to the representation gap into new services and features.

Issue 2: Exploring Tag- Based Metonymic Search

Our second Issue was that it would be valuable for people to be able to select an object and initiate a search across heterogeneous services for diverse objects similar to the selected object. We called this operation “metonymic search,” because the selected object was being used as a referent (a metonym) for its attributes. Searching (“surfing”) a unitary object (metonym) was equivalent to specifying all the attributes of the object (tags), in searches in all of the available services. More than 89% of the 57 users surfed on one or more persons, and over half of the sample surfed on Activities. About a quarter of the sample surfed on Dogear bookmarks, Feeds (some of which had tag or person metadata), or Tasks (which could optionally have tag metadata). The number of objects returned from each service was roughly proportional to the number of searches.

Crucially for part of our claim of metonymic search across diverse services, surfing on each type of object produced results from at least two types of services. Thus, the metadata from the surfed object was useful in finding resources not only in the same type of service (e.g., surf an activity to find other activities), but also in different types of services (e.g., surf an activity to find bookmarks and feeds). In this way, metonymic search also helped to reduce the Representation gap (the scatter of related resources across multiple services).

As in the previous section, the statistical results demonstrate that the data are available to support tag-based metonymic search. The next question is whether users experienced it that way.

Users' views on what was, to them, the “surf in Malibu” operation, were mixed. Some users found the concept straightforward and obvious:

Once you get the hang of the 'surf in Malibu' concept, it begs to be used, and more importantly, applied to more and different data sources, such as emails, [IM] chat transcripts, Google desktop, etc.etc.etc. (business operations manager)

Put an item in focus [“surf”], surfing [is] supposed to be able to provide things that are related, based on the thing you are pivoting on (developer)

Some users seemed less confident of the concept, or its utility;

Trying to figure out exactly how it was supposed to work was a problem. The desire to be able to have that right-click context-sensitivity was useful. (senior consultant)

And um the ability to see a filter – surfing in Malibu – that hasn’t been – either I don’t have enough in there... It doesn’t seem to bring up anything of help yet... I don’t know where it’s looking, maybe in the title... (sales executive)

About a quarter of the users were confident enough of their understanding of the tag-based search mechanism to critique it or their own practices:

Yeah that’s the pivot thing. I think it works really well, but you need the tagging stuff to support it. (communications manager)

It only surfs by the tag associated with the activity. I’m trying to be more diligent about tagging things. (business operations manager)

Most disturbingly, some users considered surfing to return bad results:

I could IM [instant-message] somebody, and then I could open another window in which it would show me other activities or maybe something else associated with that person... [Surfing on objects] sometimes the relevance of what came back with – was questionable. (chief engineer, federal systems)

With regard to our second Issue, Malibu has made some progress toward a functional tag-based metonymic search, but there is clearly room for improvement. The statistical results from the log analysis demonstrated that Malibu is capable of retrieving resources via tag-based metonymic search. User remarks in interviews and in the online forum indicate that some people found the concept quite clear, and most users had at least an intuitive understanding that surfing would produce related resources. However, several users remain confused or “struggling” with the concept, and some users were not convinced that the resources returned by the search were relevant to the surfed object. Other research (Muller, 2007) has showed that there is relatively low overlap of vocabularies from one service to another, so we will be investigating multiple normalization strategies (e.g., lower-case normalization, stemming, prefix-removal, lexical look-ups) with the hopes of increasing the number and relevance of returned objects. We also hope to add machine-learning methods to automate and improve the effectiveness of the search in the near future.

Issue 3: Recovering from Interruptions

Our third Issue was to support recovery from interruptions. We provided two features to address this problem, with two very different outcomes. First, we implemented the tag-based search feature as stack of searches. Users could navigate through previous states of the stack (and the contents of views that were produced by those states) with a simple “back” operation. In session logs, we coded this feature as a type of “focus” event – counted similarly to a focus (or pivot) on an ob-

ject. The “back” operation was the most frequently observed focus event (768 events among 57 users, or 34%), with a focus on a person as the next most frequent focus event (654 events, or 29%). Users had few comments about this feature, other than to note that it worked, apparently as expected.

Our second feature for interruption recovery was to provide a peripheral display. We hoped that this feature would allow users to preserve their work context *outside* of Malibu, as needed, because users could easily dismiss Malibu when it was not needed. We hoped that the gestural invocation (move the cursor to the edge of the screen) would help users differentiate between Malibu and conventional applications (typically invoked with a click or a command).

The log data show a very large number of event sequences in which the Malibu window slid out and then back in with no Malibu operations occurring between those two events. These data suggest that users may have invoked Malibu by accident, and then either dismissed it or waited for it to time-out and slide back into the screen border.

User comments confirmed that this was sometimes a problem:

The slide-out – at first I thought it was quite handy, but often out comes Malibu, and you start getting irritated... (procurement program manager)

You know the window, you can hide the window, but if you move your mouse cursor over a certain part of the screen, then the window appears ago [sic]. My intention was not to re-open the Malibu window, and so I stop now to use it. (information architect, whose first language was not English)

However, some users considered the slide-out/slide-in behavior to be useful:

[it] runs in the background. I have it to be hiding, and then I bring up to kind of look at my bookmarks (business process developer, whose first language was not English).

It uses a bit of your brain that has peripheral vision, it's not the kind of thing that other [applications] on your desktop use very well. I can look at Malibu without having to open my feedreader. I need to stay aware of new things, because I'm in Communications... For me, picking up my feed when I'm popping into other stuff... When it broke, I was gutted... I love the slideiness... (communications manager)

One user engaged in lengthy remote access to other machines. The “slide-in” feature of the client – a thin four-pixel bar at the side of the screen – persisted during the remote access, allowing the user to continue to use Malibu-accessed resources despite the fact that her/his current desktop view was of another machine. This user-based insight taught us that a peripheral display may provide two types of value: (a) allowing users to maintain/regain context in their own machines, and (b) allowing users to access their “home” (local) services while “remote desk-topped into another machine,” one informant added.

Earlier versions of Malibu included an option to run the client as a separate window. Some users remembered that feature, and asked us to restore it. Several of these users had dual monitors, and had developed practices that allowed them to treat one monitor as secondary, where they placed windows that functioned in a manner analogous to a peripheral display.

It appears that user reaction to our peripheral display feature was quite varied. For some users, it contributed to Malibu's success. For other users, it became a reason to stop using Malibu. We will probably return to providing options for operation as either a peripheral display or a stand-alone application.

Related Work

In addition to the large array of activity-related research we presented in the introduction of this paper, researchers have studied the usefulness of side bars for peripheral awareness in Side Show (Cadiz et al., 2002). Side Show evolved into a side bar for Microsoft Vista. Other similar products are DesktopSidebar (DesktopSidebar, n.d.), and Google Sidebar (Google, n.d.). These products focus on providing a side bar user interface framework for controlling and managing views. They do not provide tag-based or other forms of aggregated search across different data sources displayed.

Watson (Watson, n.d.) is a side-bar-like product that features automated search based on the active application on the desktop. Watson is an implicit query system that is similar to our notion of metonymic search in that it extracts key words from an object (e.g., a Word document) in focus. However, the keywords are neither social tags nor metadata, and the system sends queries via traditional search engines to find related information. Dashboard (Dashboard, n.d.) provides similar functionality. As the user reads email, browses the web, writes a document, or chats with friends, the system proactively finds objects that are relevant to the user's current desktop activity, and displays them in a separate window. Henzinger (Henzinger et al., 2005) tries to automatically find news articles on the web relevant to the ongoing stream of TV broadcast news. Their approach is to extract queries from the ongoing stream of closed captions, issue the queries in real time to a news search engine on the web, and present the top results to the user. All these implicit query systems do not leverage social tags to perform search, but rather automate the process of generating a query from content and submitting it to a search engine.

Social tags have become increasingly popular to organize and find information. Numerous systems on the Internet, -- e.g. Flickr (Cameron et al., 2006) or Delicious (Golder and Huberman, 2006), use tags as a way of managing information. However, tags are used only inside those repositories to manage a single content type. We leverage tags to search across repositories, and our search approach is metonymic, i.e. we implicitly use metadata of an object (tags) of interest to find related resources across data sources.

Conclusions and Next Steps

The Malibu Board was intended to help knowledge workers engaged in complex and intermixed collaborative activities. We outlined two particularly vexing problems: the scatter of relevant resources among multiple services, and the difficulty of searching for information relevant to the current item-of-interest.

We addressed the problem of scatter in our discussion of the representation gap. Malibu provides access to multiple stores of data and documents, and allows searching by using the attributes of other users, as well as of objects. Malibu thus provides a form of social navigation through complex information spaces, in the context of real-time social awareness via its IM environment. Statistical evidence from usage logs showed that Malibu often finds and displays resources from these multiple services, and users' comments showed that in general the combined information, in a social-awareness context, makes sense and is of value to users. The next steps are to explore the feasibility, utility, and desirability of the many extensions that users proposed to us. We will also enhance our use of person metadata from recent advances in person-tagging (Farrell & Lau, 2006).

We addressed the problem of search in complex, heterogeneous domains through a multi-service, tag-based re-implementation of the familiar "more like this" search user interface – we called this approach "metonymic search," because the item-of-interest (person or object) is used as a referent (metonym) for the attributes that are actually searched. This kind of pivoting from object to person and back again has become a strong feature in social software and social navigation: We hoped that our metonymic search could provide a more powerful basis for this emerging new socially-informed search paradigm. As we did with the representation gap analyses, we used two convergent types of evidence. We used statistical analysis of log data to show that our approach was technically feasible and functional, and we consulted with users to explore the utility and meaningfulness of our solution. User reaction ran from uncertainty to enthusiasm, with one user stating that "the 'surf in Malibu' concept... begs to be used, and more importantly, applied to more and different data sources"; this statement was repeated in the official internal evaluation of the experimental client.

Most of our users evidenced at least an intuitive understanding of the concept of tag-based search. In a subsequent experiment, we have exposed the tags associated with a surfed object for user inspection and manipulation. Early results show that users take the opportunity, when appropriate, to refine their search by selecting a subset of the tags of the surfed object. We have not been able to compare the quality of search that results from user-modification of the tag-list. At this stage, all we know is that the feature called "surf in Malibu" (tag-based metonymic search) does indeed make sense, and that users understand it well enough to want to adjust its parameters while using it.

Malibu also addressed the problem of preserving the context when users switch activities or during interruptions through a “back” operation to return to previous Malibu states, and through a peripheral sidebar user interface (the “Malibu board”). Users’ reactions to this set of features were sharply divided. For the short-term, we are considering restoring the option to run Malibu as either a sidebar or a separate application. Over the longer term, we hope to conduct a more ethnographic study to understand how people use Malibu in situ, and to understand perhaps which settings or activities make sidebars more attractive to some users and stand-alone applications more attractive to others.

Acknowledgements

We thank all our anonymous users and the Lotus Activities product team for their support, and the ECSCW 2007 reviewers for their insightful recommendations.

References

- Bardram, J.E. (2005): ‘Activity-based computing — Lessons learned and open issues’, Position paper at ECSCW 2005 workshop, *Activity — From a theoretical to a computational construct*. Paris, September, 2005.
- Bellotti, V., Ducheneaut, N., Howard, M., and Smith, I. Taking email to task: The design and evaluation of a task management centered email tool. *Proc CHI 2003*, ACM, Ft Lauderdale, FL, USA, pp. 345-352.
- Cadiz, J.J., Venolia, G.D., Jancke, G., Gupta, A. (2002): ‘Designing and deploying an information awareness interface’, *Proc. ACM CSCW 2002*, ACM, New Orleans, LA, USA, pp. 312-323.
- Cameron, C., Naaman, M., boyd, d., & Davis, M. (2006): ‘HT06, tagging paper, taxonomy, flickr, academic article, toread’, *Proc. Hypertext 2006*, ACM, pp. 31-40.
- Cutrell, E., Czerwinski, M., & Horvitz, E. (2001): ‘Notification, disruption, and memory: Effects of messaging interruption on memory and performance’, *Proc INTERACT 2001*, ICS press, Toyko, pp. 263-269.
- Dashboard (n.d.): <http://www.nat.org/dashboard/>, retrieved 9/28/06.
- DesktopSidebar (n.d.): <http://www.desktopsidebar.com/index.html>, retrieved: 9/28/06.
- Ducheneaut, N., and Bellotti, V. (2001): ‘E-mail as habitat: An exploration of embedded personal information management’, *ACM interactions*, vol. 9, no. 5, 2001, pp. 30-38.
- Egan, D.E., Remde, J.R., Gomez, L.M. Landauer, T.K., Eberhardt, J., and Lochbaum. C.C. (1989): ‘Formative design evaluation of superbook’, *ACM TOIS*, vol. 7 no. 1, January 1989, pp. 30-57.
- Farrell, S., & Lau, T. (2006): ‘Fringe contacts: People-tagging for the enterprise,’ Workshop paper at WWW2006, Edinburgh, 2006.
- Geyer, W., Muller, M.J., Moore, M., Wilcox, E., Cheng, L., Brownholtz, B., Hill, C.R., Millen, D.R. (2006): ‘ActivityExplorer: Activity-Centric Collaboration from Research to Product’, *IBM Systems Journal*, vol. 45 no. 4, 2006, pp. 713-738.
- Geyer, W., Vogel, J., Cheng, L., Muller, M. (2003), ‘Supporting Activity-Centric Collaboration through Peer-to-Peer Shared Objects,’ *Proc. ACM Group 2003*, ACM, Sanibel Is., FL, USA, pp. 115-124.

- Golder, S., & Huberman, B.A. (2006): 'Usage patterns of collaborative tagging systems', *J. Info Sci*, vol. 32, no. 2, pp. 198-208.
- González, V., and Mark, G. Managing currents of work: Multi-tasking among multiple collaborations. Springer, Paris, *Proc ECSCW 2005*, Springer, Paris, pp. 143-162.
- Google Desktop (n.d.): <http://desktop.google.com/>, retrieved 9/28/06.
- Gwidzka, J. Reinventing the inbox — Supporting the management of pending tasks in email. *CHI 2002 Doctoral Consortium*, Minneapolis, 2002.
- Halverson, C.A., Erickson, T., and Ackerman, M.S., (2004): 'Behind the help desk: Evolution of a knowledge management system in a large organization', *Proc CSCW 2004*, ACM, Chicago, pp. 304-313.
- Henzinger, M., Chang, B. Milch, B., and Brin, S. (2005): 'Query-free news search', *World Wide Web*, 8(2):101–126, 2005.
- Hudson, J.M., Christensen, J., Kellogg, W.A., & Erickson, T. (2002): "'I'd be overwhelmed, but it's just one more thing to do": Availability and interruption in research management', *Proc CHI 2002*, ACM, Minneapolis, pp. 97-104.
- Iqbal, S.T., & Horvitz, E. (2007): 'Disruption and recovery of computing tasks: Field study, analysis, and directions', *Proc. CHI 2007*, San Jose, CA, USA, ACM, pp. 677-686.
- Kaptelinin, V. (2003): 'UMEA: Translating interaction histories into project contexts', *Proc. ACM CHI 2003*, ACM, Ft Lauderdale, FL, USA, pp. 353-360.
- McFarlane, D.C. (2002): 'The scope and importance of human interruption in human-computer interaction', *Human Computer Interaction*, vol. 17, no. 1, pp. 1-61.
- Millen, D.R., Feinberg, J., Kerr, B. (2006): 'Dogear: Social bookmarking in the enterprise', *Proc CHI 2006*, Montreal, ACM, pp. 111-120.
- Monk, C.A., Boehm-Davis, D.A., & Trafton, J.G. (2004): 'Very brief interruptions result in resumption cost', Poster at 26th Annual Cognitive Science Society meeting, Chicago, 2004.
- Moran, T.P., Cozzi, A., and Farrell, S.P., 'Unified Activity Management: Supporting People in eBusiness,' *Communications of the ACM* 48, No. 12, Special section on Semantic eBusiness Vision, 67–70 (December 2005).
- Muller, M.J.: 'Patterns of tag usage across four diverse enterprise tagging services', paper at HCIC 2007, Winter Park, CO, USA, February 2007.
- Muller, M.J., Geyer, W., Brownholtz, B., Wilcox, E., and Millen, D.R. (2004): 'One-hundred days in an activity-centric collaboration environment based on shared objects', *Proc. CHI 2004*, ACM, Vienna, pp. 375-382.
- Sen, S., Geyer, W., Muller, M.J., Moore, M., Brownholtz, B., Wilcox, E., & Millen, D.R. (2006): 'FeedMe: A collaborative alert filtering system', *Proc. CSCW 2006*, ACM, Banff, Alberta, CA, pp. 89-98.
- Sullivan, S. (2006): 'A generalized interruption manager', short paper submitted to CHI 2006, Montreal.
- Watson (n.d.): <http://www.intellect.com/>, retrieved 09/28/06.
- Whittaker, S. (2005): 'Supporting collaborative task management in email', *Human Computer Interaction* 20, 1&2, 49-88.