# Improving Proactive Information Systems

Daniel Billsus, David M. Hilbert
FX Palo Alto Laboratory, Inc.
3400 Hillview Avenue, Building 4
Palo Alto, CA 94304 USA
+1 650 813 7516

{billsus, hilbert}@fxpal.com

Dan Maynes-Aminzade
Computer Science Department
Stanford University
Stanford, CA 94305 USA

monzy@stanford.edu

## ABSTRACT
Proactive contextual information systems help people locate information by automatically suggesting potentially relevant resources based on their current tasks or interests. Such systems are becoming increasingly popular, but designing user interfaces that effectively communicate recommended information is a challenge: the interface must be unobtrusive, yet communicate enough information at the right time to provide value to the user. In this paper we describe our experience with the FXPAL Bar, a proactive information system designed to provide contextual access to corporate and personal resources. In particular, we present three features designed to communicate proactive recommendations more effectively: *translucent recommendation windows* increase the user's awareness of particularly highly-ranked recommendations, *query term highlighting* communicates the relationship between a recommended document and the user's current context, and a novel *recommendation digest* function allows users to return to the most relevant previously recommended resources. We present empirical evidence supporting our design decisions and relate lessons learned for other designers of contextual recommendation systems.

## Categories and Subject Descriptors
H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *User profiles and alert services*

## General Terms
Design, Experimentation, Human Factors

## Keywords
Proactive Recommendations, Context, Agents

## 1. INTRODUCTION
Whether we are emailing, searching the World Wide Web, shopping for products, or perusing the news, computers play a central role in our daily lives. With so many online resources available to us, how can we stay abreast of all the potentially useful information without spending all our time searching and filtering?

The problems of *resource discovery* and *information overload* extend beyond our personal lives and into the office. When we are unaware of relevant information and human resources, the quality, efficiency, and satisfaction of our work suffers. When separate corporate units are unaware of one another's activities and expertise, they unnecessarily duplicate work. Meanwhile, we are faced with an increasingly dizzying array of information sources including the Internet, intranets, databases, file servers, personal computers, and mobile devices. As a result, systems that enhance resource discovery while limiting information overload continue to receive attention in both academic and commercial contexts.

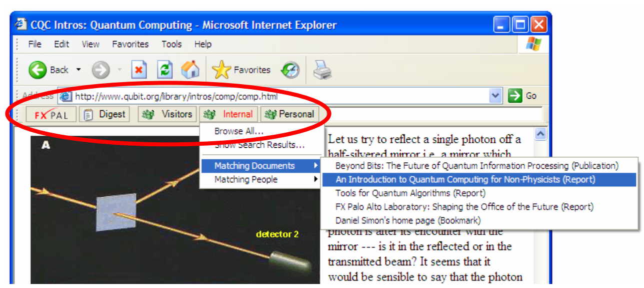



**Figure 1: Initial FXPAL Bar User Interface**

## 2. PROACTIVE INFORMATION SYSTEMS
Automatically generated recommendations are one way to assist people in discovering relevant information. Online retailers like Amazon.com display related products as we shop, and search engines like Google display targeted advertisements based on our search terms. These recommendations are both *proactive*, in that the user does not have to explicitly request a recommendation, and *contextual*, in that recommendations are related to the user's current context, e.g. the recent purchase of a rap album, or a web search for "German beer."

Proactive contextual recommendations are not limited to products and advertisements; they are also being leveraged to support corporate knowledge management. Research systems like Watson [2] and Remembrance Agent [3] provide proactive contextual


*IUI'05*, January 9–12, 2005, San Diego, California, USA.

access to corporate and personal resources. Products from knowledge and content management infrastructure venders like Autonomy and Verity proactively recommend documents, based on implicit techniques designed to anticipate users' information needs.

## 2.1 Problems with Proactive Information Systems

While proactive recommendations can be useful in a variety of scenarios, proactive information systems struggle with the following problems:

First, their interfaces are often either too subtle or too obtrusive. Some systems communicate recommendations through very minimalist interfaces, such as icon changes indicating the availability of related information. As a result, the user can frequently miss potentially useful information. Other systems are not subtle enough, using distracting recommendation methods like large, blinking windows or modal dialog boxes; these are so obtrusive and annoying that they are quickly ignored or disabled.

Second, not all contextual recommendations are relevant, leading some users to ignore them altogether. This problem is exacerbated by subtle recommendation interfaces that don't communicate enough information for users to quickly determine if a recommendation is of interest to them. This can lead users to ignore recommendations altogether.

Third, users are often very task-focused, and even if they notice an interesting recommendation, they may not be willing to interrupt their current task to examine the recommended information.

## 2.2 Improving Proactive Information Systems

In this paper we describe: (1) the design of the FXPAL Bar, (2) a user study we performed to identify how to improve it, (3) three design changes motivated by our study, and (4) a follow-up study to evaluate the effects of our design changes. We present design guidelines that are generally applicable to information recommendation systems, and include qualitative and quantitative evidence to support our claims.

# 3. THE FXPAL BAR SYSTEM

The FXPAL Bar is a corporate memory system that provides contextual access to relevant documents and contacts. It was originally designed to support serendipitous discovery and contextual access to relevant corporate and personal contacts, as described in [4]. Since then, the system has evolved into a more general corporate memory tool that also maintains and recommends corporate and personal documents. In the following sections we describe the original user interface, the system architecture, and the underlying recommendation algorithms.

## 3.1 Original User Interface

The FXPAL Bar is a toolbar for Microsoft Internet Explorer. Its initial design was quite simple, consisting of only four buttons (Figure 1). The three right-most buttons are typically grayed out, but change color when the system recommends resources related to the currently displayed web page. Clicking one of the three content buttons on the right displays drop-down menus containing lists of recommended resources. The left-most button produces a drop down menu that allows users to change their preferences about how recommendations are presented, and to browse the full contents of the document database.

Clearly, this interface was subtle and unobtrusive. However, its recommendations were almost universally ignored. After describing the system architecture, we go into the details of why this occurred and how we extended the Bar to address this problem.

## 3.2 System Architecture

As shown in Figure 2, the FXPAL Bar uses a client-server framework in which all Bar clients are powered by a single server, the Adaptive IR Server. This server exposes a set of web services that support various content management and recommendation functions, allowing it to be accessed from any software component that can interact with web services. This allows external applications or other embedded clients, such as toolbars for email applications or word processors, to take advantage of the system's recommendation capabilities.

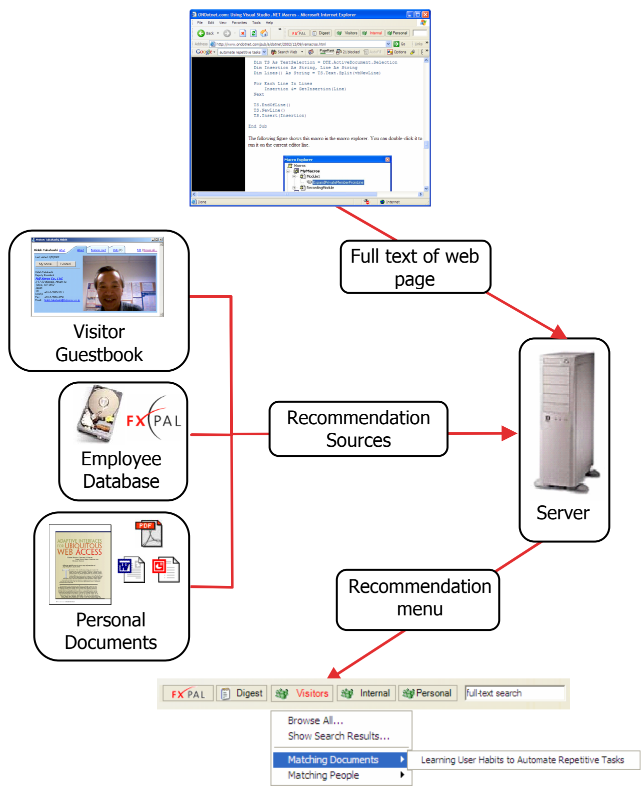



**Figure 2: FXPAL Bar Architecture**

The server runs a relational database that supports full-text search, which it uses to maintain information about users and available content. In its current deployment at FXPAL, the server has access to three distinct content repositories:

1. *Visitor* information from FXPAL's visitor guestbook, containing visitors' contact information, recorded videos, and uploaded documents, as described in [4].
2. *Internal* information, such as employee contacts, internal publications, reports, memos and patents.

3. *Personal* information, such as contact lists, bookmarks, and any personal documents uploaded to the server.

To keep the system's content up to date, we automated the content import process as much as possible. For example, new visitor information is automatically added to the database, and the system is periodically synchronized with FXPAL's internal document repository. In addition, when users bookmark new web pages, the system can automatically (or interactively) add the new content to the repository. Finally, users can easily upload collections of existing bookmarks.

The information flow through the system is straight-forward: when a user navigates to a web page, the FXPAL Bar extracts the full text of the page (currently ignoring any layout or format information), and sends it to the server. The server then identifies matching resources and sends information about identified matches back to the client.

All recommendations are transmitted in a serialized menu hierarchy, which the client renders as drop-down menus that are displayed when the user clicks on one of the three resource buttons. The advantage of this approach is that it is flexible and lightweight: changes in the menu hierarchies can be implemented on the server without redeploying the client. The client is simply a very thin communication layer without any recommendation logic; it knows how to extract text from web pages and send it to the server, how to render menus, and how to react to the user's button selections by displaying menus or navigating to web pages.

## 3.3 Recommendation Algorithms

The FXPAL Bar currently supports two separate recommendation approaches: (1) *contact* recommendations based on explicit matches of known contacts in the currently displayed page, and (2) *content* recommendations based on textual similarity between the currently displayed page and other documents in the system's database.

To determine contact matches, the server's recommender component analyzes the transmitted text and uses a matching algorithm to detect occurrences of contact information fragments that match entries in the system's contact database. The algorithm is sophisticated enough to deal with a wide variety of potential formatting differences of names and contact information (see [4] for details).

Recommendations based on content similarity are determined via a simple two step process. First, the *query generation* step converts the currently displayed web page to a weighted query. Second, the *recommendation* step uses the query to determine a set of candidate documents and then determines whether the retrieved candidate documents should be recommended to the user.

The *query generation* step proceeds as follows. When the server receives the extracted text of the currently displayed web page, it is converted into two separate normalized Vector Space Model *tf-idf* term vectors [5]. One term vector uses individual words (unigrams), while the other term vector uses word pairs (bigrams). The underlying *df* component of these term vectors is based on the server's document collection and an additional set of anonymously logged, previously visited web pages. This ensures that the document frequencies reflect both the document collection and users' actual browsing patterns.

The unigram and bigram vectors are sorted by their respective term weights, and the server uses the resulting term lists to construct the query in three steps. First, up to $n$ unigrams that exceed a fixed threshold $t$ (currently set to *5* and *0.2*, respectively) are taken as the initial query terms. Second, up to $m$ bigrams that exceed another threshold $u$ are added to the query (currently set to *5* and *0.1*, respectively). Finally, the third query generation step biases the query toward the server's document collection. The underlying intuition is that our knowledge about the available documents and recurring topics within these documents can be used to add query terms that represent important topics within the document collection, even if they do not stand out statistically on the currently displayed page. For example, several researchers at FXPAL have done work on video segmentation, and as a result, FXPAL's document collection contains many documents that contain the bigram "video segmentation." The goal of the final query generation step is to include bigrams like "video segmentation" if they appear on the currently displayed page, regardless of their associated *tf-idf* weights, thus making it more likely that highly relevant aspects of a displayed page contribute to the query. In our implementation, this step requires a pre-computed list of *informative query bigrams,* which is currently generated once a week according to the following algorithm. All documents in the document collection are converted to *tf-idf* bigram vectors, and restricted to the $n$ top-ranked bigrams. For each bigram in the resulting vocabulary, the server then counts the number of times the bigram occurs in a top-n bigram vector, and sorts bigrams according to this frequency count. The top $m$ bigrams from this list form the *informative query bigram* list. The server then uses all unigrams and bigrams identified in steps one through three and constructs a weighted query, where the weights are normalized *tf-idf* weights of all query terms.

The *recommendation step* proceeds as follows. First, the server retrieves the full text of the top $n$ query results and constructs corresponding *tf-idf* unigram vectors ($n$ is currently set to 10). The server then determines the exact similarity of the current web page and each retrieved document using the *cosine similarity measure* [5], and sorts the documents accordingly. Documents that exceed a similarity threshold $t$ are then added to their corresponding recommendation menus. The previously determined contact matches are added to the same menus, and the resulting menu hierarchy is sent to the Bar client. The client checks the received menu hierarchy, and changes the color of buttons whose corresponding menu contains at least one recommendation, indicating the availability of contact or content recommendations to the user.

# 4. INITIAL USER STUDY

After deploying the FXPAL Bar on machines throughout our lab, we found that its usage rate was disappointingly low. Users were generally ignoring recommendations, and we wanted to know why. We asked our users to fill out a questionnaire about their impressions of the Bar, in an attempt to identify design problems and discover opportunities for its improvement. The questionnaire consisted of a variety of statements about the Bar; users could indicate the extent to which they agreed or disagreed with each statement. We also included a section of free-form questions that allowed users to offer general comments and suggestions.

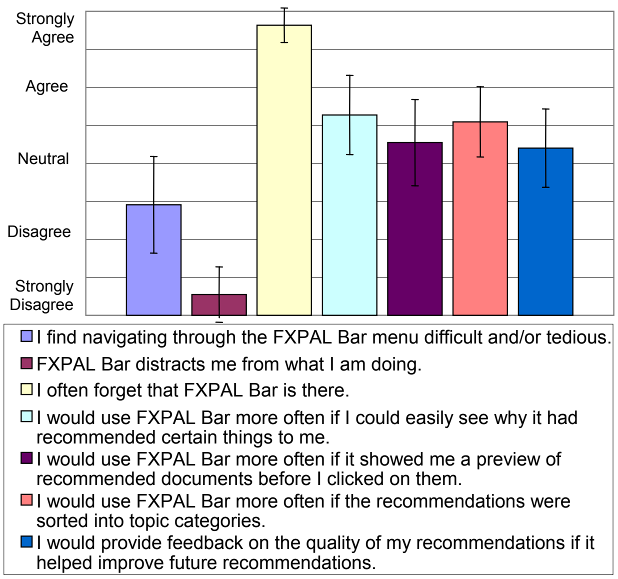



**Figure 3: Survey responses, averaged across all respondents, with 95% confidence intervals**

**Table 1: Problems and proposed design changes**

| Problem | Severity (1-5) | Proposed Fix |
|---|---|---|
| I often forget that the FXPAL Bar is there. | 4.82 | Occasionally display recommendation windows |
| I would use the FXPAL Bar more often if I could easily see why it had recommended certain things to me. | 3.64 | Highlighted query terms on page when recommendation window appears or user clicks on menu |
| I would use the FXPAL Bar more often if the recommendations were sorted into topic categories. | 3.55 | Server-side clustering of search results |
| I would use the FXPAL Bar more often if it showed me a preview of recommended documents before I clicked on them. | 3.27 | Content preview available in recommendation window |
| I find navigating through the FXPAL Bar menu difficult and/or tedious. | 2.45 | Decrease number of clicks for menu navigation. |

After collecting responses from 11 users (not including the authors or developers), we identified and prioritized a variety of problems with the FXPAL Bar. Selected responses from the study are shown in Figure 3, and a summary of the most significant problems and proposed design changes is presented in Table 1.

# 5. INTERFACE CHANGES

Based on the feedback from the user study, we introduced three new features to the FXPAL Bar, each intended to address particular problems with the interface.

## 5.1 Recommendation Windows

Our initial recommendation interface was so subtle that our users tended to forget that it was there. We decided to occasionally display recommendation notification windows in cases in which the system judged a recommendation to be of exceptionally high quality.

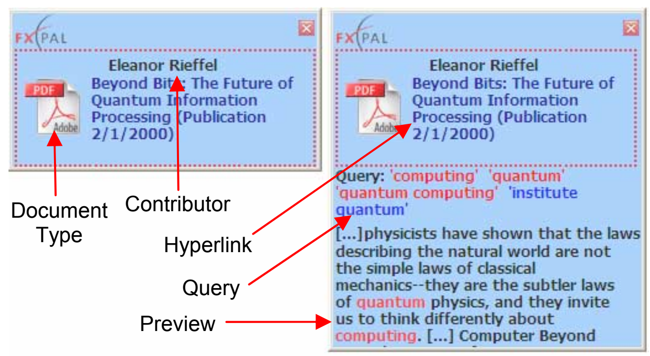



**Figure 4: Translucent Recommendation Windows**

When a recommendation is judged to be particularly relevant, the server sends a special command indicating that a recommendation window should be displayed. Recommendation windows are small, partially transparent windows that fade in on the corner of the user's screen, similar in style to the translucent email notification windows in Outlook 2003. The windows display a link to the recommended document, the document type, and the query terms, which are colored to indicate which terms did and did not appear in the recommended document. Users can also resize the windows to taste to show a document preview consisting of snippets extracted from the document, with the query terms highlighted. For example, the web page displayed in Figure 1 generated the recommendation window shown in Figure 4. Recommendation windows automatically fade away after a short delay unless the mouse cursor is moved over them.

There is, of course, a fine line between making our recommendations more noticeable and making them overly invasive or annoying. We addressed this by making our window design small and understated and making the relevance threshold for triggering a window quite high.

## 5.2 Query Term Highlighting

Several users of the original FXPAL Bar system noted that they did not understand why they were receiving a particular recommendation, and that this detracted from the credibility of the recommendations. To address this, we introduced query-term highlighting. When a user clicks on the recommendation menu, the system highlights terms on the current page that contributed to the automatically generated query. In addition, query terms are automatically highlighted when recommendation windows are displayed.

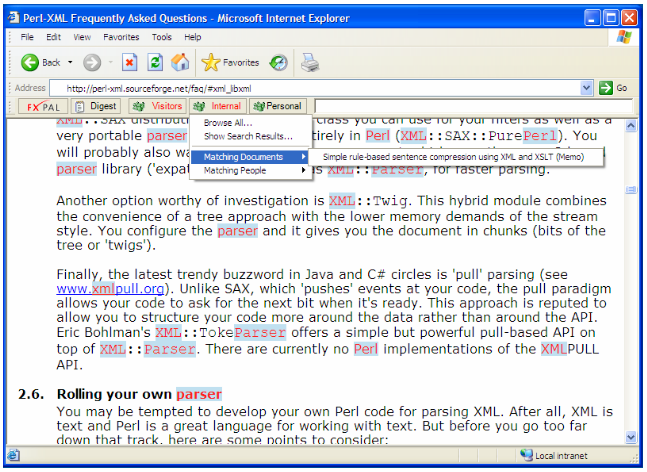


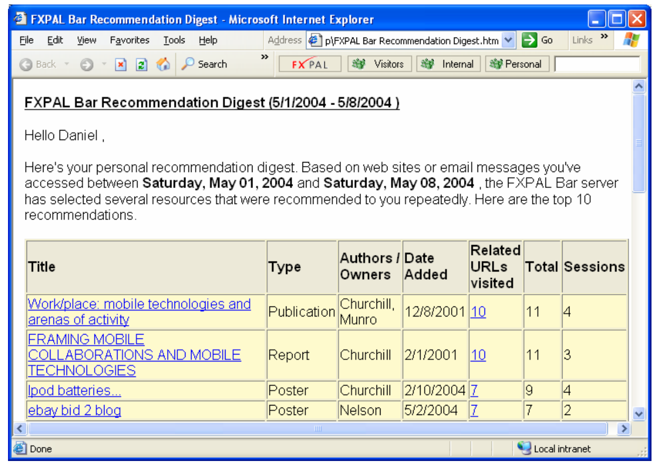

**Figure 5: Query Term Highlighting**

While we cannot design a system that always gives perfect recommendations, we can make the recommendation interface provide enough cues to allow users to quickly judge for themselves why the recommendation was made and how likely it is to be relevant. For example, if users notice a particular highlighted term on the page that grabs their interest, they can visit the recommended documents to learn more; conversely, if the system generated a less interesting set of query terms, users can ignore the recommendations without wasting any time.

## 5.3 Recommendation Digests

While the features described above are intended to make recommendations more understandable and bring highly relevant recommendations to the user's attention, they do not address the fact that users may be so engaged in other tasks that they do not want to access related content at the time it was recommended, no matter how interesting or relevant it seems. Furthermore, recommendations that are based exclusively on the current context are sometimes irrelevant or inaccurate, because they are based on limited contextual cues and not explicitly stated information needs. To address these two problems, we extended the FXPAL Bar with a *recommendation digest* function. This feature allows users to return to previously recommended content through a digest document that contains links to recommendations deemed particularly relevant. By clicking the digest button, users can request digest documents that contain a subset of the recommendations presented during the preceding hour, day, or week. Alternatively, users can request that digests be sent periodically via email. Most FXPAL Bar users currently receive an automatically generated digest document once a week.

To determine which recommendations to include in the digest documents, we use a *recommendation aggregation* approach designed to restrict digests to only the most relevant information. This approach is simple and general enough to be applied to a broad range of proactive recommendation scenarios. Whenever a recommendation R is recommended to user U, the pair R-U is logged in our database along with additional information such as the date and time of the recommendation, the context that triggered the recommendation, and a measure of the system's confidence in the recommendation (the similarity score). This produces a recommendation log that can be efficiently turned into a recommendation digest by aggregating individual recommendations. The intuition underlying this process is that items that were recommended multiple times to the same user over a specified period of time (potentially based on multiple different contexts) are more likely to be related to the user's information needs or interests than items that were recommended infrequently or only based on a small number of visited pages.



FXPAL Bar Recommendation Digest (5/1/2004 - 5/8/2004 )

Hello Daniel ,

Here's your personal recommendation digest. Based on web sites or email messages you've accessed between **Saturday, May 01, 2004** and **Saturday, May 08, 2004** , the FXPAL Bar server has selected several resources that were recommended to you repeatedly. Here are the top 10 recommendations.

| Title | Type | Authors / Owners | Date Added | Related URLs visited | Total | Sessions |
|---|---|---|---|---|---|---|
| Work/place: mobile technologies and arenas of activity | Publication | Churchill, Munro | 12/8/2001 | 10 | 11 | 4 |
| FRAMING MOBILE COLLABORATIONS AND MOBILE TECHNOLOGIES | Report | Churchill | 2/1/2001 | 10 | 11 | 3 |
| Ipod batteries... | Poster | Churchill | 2/10/2004 | 7 | 9 | 4 |
| ebay bid 2 blog | Poster | Nelson | 5/2/2004 | 7 | 7 | 2 |



**Figure 6: Recommendation Digest Example**

Each individual recommended document logged in our database is described by a *document ID,* a *date* and *time* stamp, and a *context identifier* (for our system, this is simply the URL of the page that triggered the recommendation). To facilitate subsequent aggregation we also store the recommendation *score, rank* and *session ID*. The stored *recommendation score* is simply the similarity of the recommended document and the visited page. The *rank* is the position of the recommended document in a list sorted by the *recommendation score.* The *session ID* helps us to distinguish between individual usage sessions during which an item was recommended to a user.

The main goal of the aggregation process is to identify recommendations that were repeatedly recommended to the same user during a specified period of time, preferably based on a large number of distinct contexts. The server generates the following statistics as part of the aggregation process:

- *Number of distinct contexts.* In the example shown in Figure 6 this is the number of related URLs visited, and all listed documents are sorted by this number. The number of distinct contexts is likely to be a strong relevance indicator, as every recommendation of the same document based on a different context can be interpreted as an additional piece of evidence supporting the document's relevance.
- *Total number of recommendations.* This is the total number of times a document was recommended, including recommendations based on repeat visits to the same page.
- *Number of distinct sessions.* This is the number of distinct usage sessions during which a resource was recommended, which is an additional indicator of recurring long-term interests.

It is important to note that the resulting aggregation approach is completely independent of the underlying recommendation algorithm that generates contextual recommendations. Although our system produces recommendations based on textual similarity, the aggregation method could be just as effective in processing the recommendations generated through collaborative filtering or any other predictive or associative method described in the literature. In addition, logged recommendations can be aggregated very efficiently, making our approach easily scalable to real-word requirements. For example, an obvious implementation choice is to persist recommendations in a relational database. In this case, a simple aggregation algorithm can be expressed efficiently in the form of an SQL query that sorts recommendations based on frequency counts of variables such as *context ID* or *session ID*, and limits results according to *date*, *score* and *rank* constraints.

The example digest shown in Figure 6 focuses on the user's most significant browsing patterns over a week and brings relevant corporate resources to the user's attention. In the example shown, the digest is rendered as a web page that the user has explicitly requested. The same digest can optionally be emailed to users at user-specified intervals.

## 6. EVALUATION

After the enhanced version of the FXPAL Bar had been in use for approximately a month, we asked our users to respond to a second questionnaire to find out if our changes had improved the system's effectiveness. The results of this second study are shown in Figures 7 - 11.

Since the new interface was less subtle than the original, we expected it to be noticed more. We found that significantly fewer people reported forgetting about the Bar (unpaired two-tailed t-test: $p=0.0093$). More people reported that the bar distracted them, but the difference here was not statistically significant. People also reported a higher frequency of use and greater relevance and usefulness of recommendations, but these differences were not statistically significant either.

More telling were the responses to questions in which we asked people directly what they thought of the new features. On average, the recommendation digest and the translucent recommendation windows were well-liked. For example, Figure 7 shows that our users felt they would be far more likely to access content through the windows and digest, than through the original toolbar menus. Our users were divided on their opinions of the query term highlighting. However, there was enough variability in people's opinions of all three of these features that we concluded that the best solution was to include all three as options, but to allow them to be enabled or disabled according to each user's preferences.

In addition to the questionnaire, we analyzed our server logs to determine if a quantitative analysis of usage data would corroborate our findings. To accomplish this, we instrumented the FXPAL Bar client to submit all user interactions such as menu and content selections to the server for logging purposes. Likewise, all URLs that were accessed through the recommendation windows or the digest used a redirector on our server, so that our users' browsing activity could be logged. Since the goal of the system is to bring potentially relevant documents to the user's attention, we wanted to see if there was a measurable difference in the number of recommended documents that were actually opened by our users. While this is not a perfect measure, we believe that it is at least an indicator of the system's capability to bring relevant content to the user's attention.

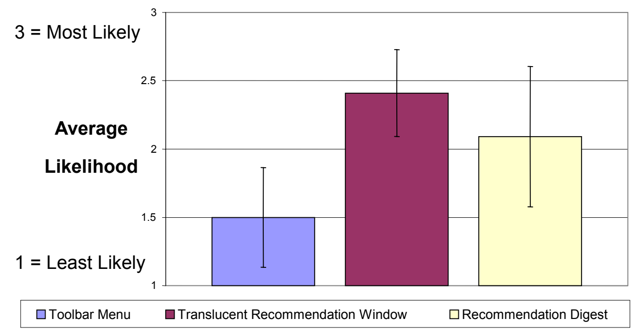



**Figure 7: Likelihood of Accessing Content from each Recommendation Source**

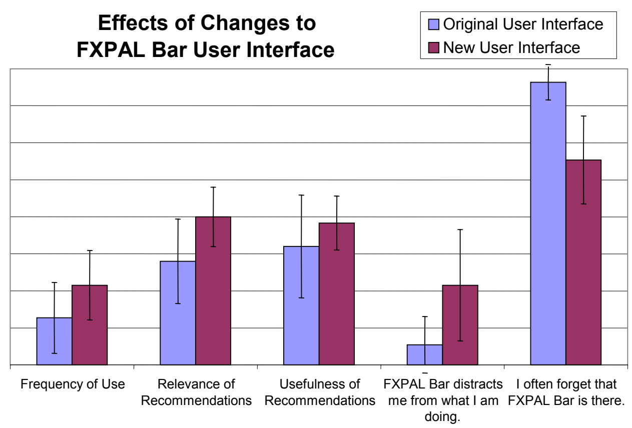



**Figure 8: Effects of UI Changes**

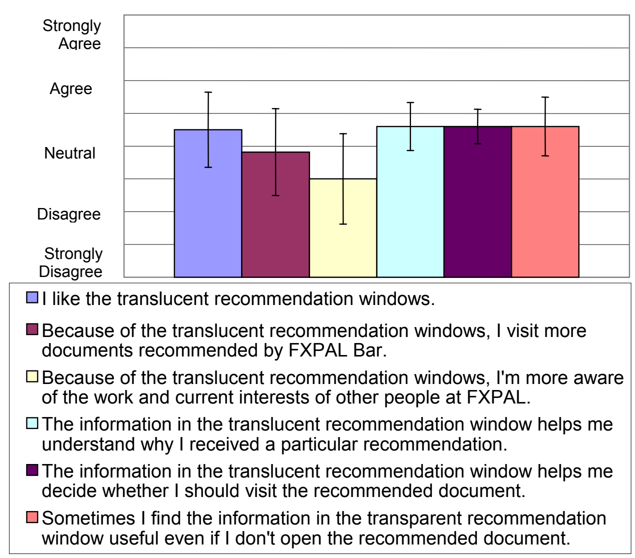



**Figure 9: Opinions of Recommendation Windows**

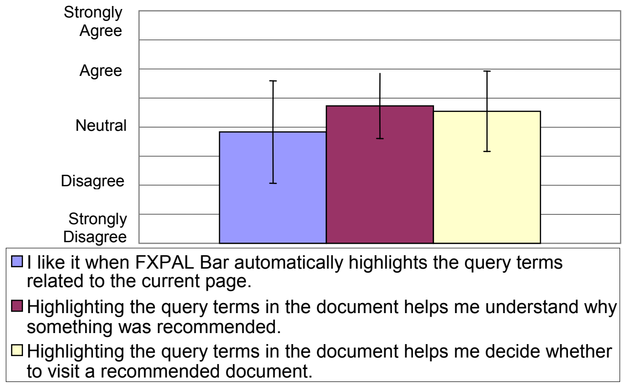



**Figure 10: Opinions of Query Term Highlighting**

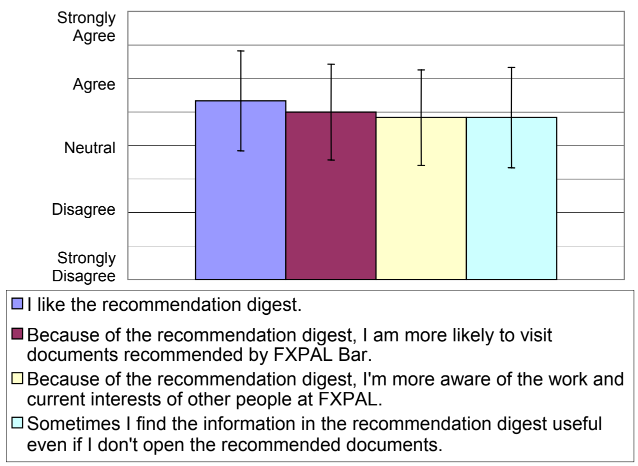



**Figure 11: Opinions of the Recommendation Digest**

Naturally, we ensured that the analyzed data did not include any usage data from the authors or developers who worked on the system. We also made sure not to collect any data immediately after deployment of the system or deployment of new features, so that the analysis would not be affected by users' desire to test new functions by selecting content without actually being interested in any recommendations. The resulting data set consisted of 10 distinct users. An initial analysis of a portion of this data set collected with the original Bar interface over a period of 6 months revealed that users rarely opened any documents through the recommendation menus: a total of 36 documents were opened by 5 users, meaning that only about 6 documents were accessed per month and that half of our users never opened any documents at all.

In contrast, data collected over a period of two months after we deployed the modified system (starting approximately one month after redeployment), revealed that 45 documents were opened by 8 users. 17 of these 45 documents were accessed through the recommendation windows, 26 through the digest and 2 through the menus. While the considerable increase in opened documents is based on a relatively small user sample, we interpret it as an additional indicator of the system's improved effectiveness. While the reported numbers remain low, they are realistic given that the current thresholds for recommendation windows are fairly high (typically, these windows appear less than 3 times a day per user) and that digests are sent out only once a week. In addition, we suspect that our current document collection may account for the overall low access rates – users at FXPAL, a small research institution with about 20 researchers, are already familiar with most existing internal reports or publications, limiting the novel content that can be discovered. It is also important to note that several users reported that the information contained in the recommendation windows and the digests can be quite useful even without opening the recommended documents, suggesting that the utility of the system cannot be measured through server logs alone.

In summary, we interpret both the qualitative and quantitative aspects of the evaluation as initial indicators of the system's improved effectiveness.

## 7. RELATED WORK

Several proactive information systems, sometimes called *reconnaissance agents* or *just-in-time retrieval agents*, are similar in principle to the FXPAL Bar. For example, Watson [2], WebTop [7], Remembrance Agent [3], Blinkx[1], and Dashboard[2] all have proactive querying and content recommendation capabilities. We believe that these systems share most of the UI design problems discussed in this paper: their interfaces are so subtle that suggestions are easily ignored; users may not understand why certain resources were recommended, do not know what to expect when visiting the recommendations, and generally cannot easily return to previously recommended content.

Clearly, interface elements similar to the ones discussed in this paper have been used previously in other systems. For example, our translucent recommendation windows are similar to the email notification windows used in Microsoft Outlook 2003: they communicate useful information without being too obtrusive, and fade away after a short delay.

Finally, the recommendation digest is conceptually similar to Amazon.com's "the page you made" feature. Amazon users can view a list of recommendations related to their past shopping interactions by clicking on a link labeled "the page you made". However, these lists are not produced using aggregation; clicking on Amazon's "Why was this recommended" links reveals that recommendations are based on the user's product access and purchase history, but not on frequencies of past recommendations or consideration of the various contexts from which recommendations were triggered. For example, if a user repeatedly returns to the product page of a certain CD to listen to sound clips, the user will likely receive repeated recommendations for a second related CD. Using our aggregation approach, the fact that the second CD was recommended repeatedly could be used to increase the relevance of that recommendation.

[1] http://www.blinkx.com

[2] http://www.nat.org/dashboard

## 8. FUTURE WORK

This paper focused primarily on the rationale, design, and algorithms of novel delivery methods for proactive recommendations. Clearly, the utility of proactive information systems is not only a function of the user interface or content delivery method, but is largely determined by the relevance and utility of the generated recommendations. In its current instantiation, the FXPAL Bar automatically finds resources similar to a displayed document. However, it is obvious that *similar* does not necessarily mean *useful*. More work is needed in the area of contextual retrieval algorithms, an area of active research that is widely regarded as a challenging but promising direction for IR research [1]. In particular, we plan to expand the system's notion of context to include models of the user's current task (e.g. "searching", "writing", "browsing", "shopping", etc), as well as user models that comprise information about the user's knowledge and interests. Some initial steps towards this goal are quite simple: for example, the server could easily keep track of documents that users have seen before or are likely to know about, so that the recommendation algorithm can take the *novelty* of content into account. Similarly, certain tasks during which recommendations are generally not welcome, such as checking a bank statement, or viewing a family picture album, may be easy to detect. Other contextual retrieval enhancements, such as modeling users' preferences and interests, may involve more sophisticated techniques, possibly based on implicit user model acquisition using machine learning techniques [6].

A second branch of future research will focus on contextual cross-lingual retrieval. A system that could proactively retrieve relevant documents in multiple languages would not only be useful for our own deployment at FXPAL, but also address a growing need for support of multi-lingual document collections [1].

## 9. CONCLUSIONS

Proactive contextual information systems help people discover useful resources while limiting information overload. As they grow in popularity, their designers face a variety of user interface challenges. Based on our experience with the FXPAL Bar, we described some problems that we believe to be representative for these systems, and presented three features we designed to overcome these problems. *Translucent recommendation windows* containing a brief document excerpt, paired with *query term highlighting*, make the recommendation process more understandable and bring particularly relevant recommendations to the user's attention. *Recommendation digests* identify particularly relevant recommendations based on access frequencies and the diversity of the recommendation context, and support any-time access to the resulting recommendation digests. Initial qualitative and quantitative results obtained from users in our lab confirmed that the features we described increased the usage and utility of our system.

## 10. REFERENCES


[1] Allan J. (ed) et al. Challenges in Information Retrieval and Language Modeling. *SIGIR Forum*, vol. 37:1, ACM, 2003.

[2] Budzik, J., Hammond K., and Birnbaum, L. Information Access in Context. *Knowledge-Based Systems* 14 (1-2), pp 37-53, Elsevier Science, 2001.

[3] Rhodes, B. *Just-In-Time Information Retrieval*. Ph.D. Dissertation, MIT Media Lab, 2000.

[4] Trevor, J. et al. Contextual Contact Retrieval. *Proceedings of the International Conference on Intelligent User Interfaces* (IUI 2004), Funchal, Madeira, Portugal, 2004.

[5] Van Rijsbergen, C. J. *Information Retrieval*. London: Butterworths, 1979.

[6] Webb, G., Pazzani, M. and Billsus, D. Machine Learning for User Modeling. *User Modeling and User-Adapted Interaction*, 11: 19-20, 2001.

[7] Wolber, D., Kepe, M., and Ranitovic, I. Exposing Document Context in the Personal Web. *Proceedings of the International Conference on Intelligent User Interfaces* (IUI 2002), San Francisco, CA, 2002.