Involving Remote Users in Continuous Design of Web Content

William C. Hill and Loren G. Terveen
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932
+1 201 360-8342
{willhill, terveen}@research.att.com

ABSTRACT

PHOAKS is a system that automatically recognizes URLs recommended in Usenet messages and continuously updates a large web site that summarizes the recommendation data. We view the automatically generated pages as "rough drafts" that users help to refine. We report here on the mechanisms that allow users to do this, our rationale for these mechanisms, and the issues raised by involving thousands of remote anonymous users in the continuous design of web content.

Keywords

human-computer interaction, human interface, computer-supported cooperative work, organizational computing, social filtering, collaborative filtering, resource discovery, World Wide Web, Usenet, participatory design, remote evaluation, end user modification.

INTRODUCTION

The main goal of participatory design [13] is to involve users in the process of system design. It focuses on a specific common design process and style of participation. First, the approach assumes that design (and thus user participation) occurs mostly before use. Second, the approach centers on small groups of professional designers and users working together in face-to-face meetings. Neither of these assumptions fits the design situation we face with our PHOAKS (People Helping One Another Know Stuff) system — http://www.phoaks.com/phoaks. With PHOAKS, we are exploring issues of continuous and remote participatory redesign.

PHOAKS is a general architecture for recognizing recommendations in conversational data, maintaining databases of recommendations and associated contextual data, and continuously updating a web-based interface to the data. The PHOAKS application recognizes recommendations of web resources (URLs) from netnews messages. Since the recommendation data we present on our web pages is recognized algorithmically, it is possible for it to be *inaccurate* (if our algorithms categorize something as a recommendation that is not), *incomplete* (if our algorithms miss something that should be counted as a recommendation), and *out-of-date* (when a recommendation no longer is relevant or the location of a web resource has changed).

Because of these potential problems, we view the automatically generated PHOAKS web pages as "rough drafts". We want to address the problems of accuracy, completeness, and currency by involving our users. Our users, after all, know and care about the covered topics. This is a type of participatory design of content. However, this is a participatory design with some peculiar characteristics. First, content is continuously refined as it is used rather than prior to its use. Second, user feedback is received from hundreds or thousands of remote anonymous users rather than being produced and interpreted in a small working group.

Initially, PHOAKS only accepted email feedback from users, and we, as the system designers, manually dealt with all user suggestions. However, by analyzing user feedback (and patterns of access to our web site), we were able to identify and design several specialized mechanisms that support user involvement, including expressing an opinion on existing content, updating existing content, and suggesting new content to be added.

In the remainder of the paper, we describe the PHOAKS system, examine in detail the type of feedback we have received from users, report on the mechanisms we have added in response, and discuss problems and issues associated with this type of participatory design in use.

BACKGROUND AND RELATED WORK

PHOAKS is a computer-mediated collaborative (or social) filtering system [11]. Its job is to recognize, manipulate, store and present content that occurs first in on-line conversational streams. Specifically, it offers lists of web resources obtained by examining netnews messages and classifying certain references to URLs as recommendations. Three design principles distinguish it from other collaborative filtering systems: *role specialization, reuse and recontextualization.*

DIS '97 Amsterdam, The Netherlands

Many collaborative filtering systems, particularly ratings-based systems [3, 9, 14, 15], are built on the assumption of *role uniformity*. They expect all users to do the same types of work in return for the same types of benefits: for example, in the case of ratings-based systems, everyone rates objects of interest. Yet there is evidence that people naturally prefer to play distinct producer / consumer roles in the information ecology [3]; in particular, only a minority of people expend the effort of judging information and volunteering their opinions to others. Independently, we have observed such role specialization in netnews. First, a very small number of posters volunteer long lists of recommended web resources at a low but stable rate. Second, a small minority of posters (5-20%) account for the majority of message traffic on most newsgroups. PHOAKS assumes that the roles of recommendation provider and recommendation recipient are specialized and different.

PHOAKS *reuses* recommendations from existing online conversations. This reuse requires no extra work from providers, which distinguishes PHOAKS from collaborative filtering systems that make recommending an explicit task [5, 12] and from many group memory systems [1, 2, 16]. Nor does PHOAKS require judgments of information quality from users, which is another difference from ratings-based systems.

Reusing information for new purposes forces either decontextualization or *recontextualization* of the information. PHOAKS takes the second option. It pays a great deal of attention to context and bases the interface on it [7, 8, 19, 20]. PHOAKS selects which URLs to present by computing over contextual information. For instance, PHOAKS removes web resource messages that occur as part of contact information in signatures. Furthermore, PHOAKS attaches links to source message text and ranks resources by the number of times they occur in a recommendation context in the newsgroup stream.

Since PHOAKS content is automatically generated, it is likely to be of less quality than human-generated content. (We have studied the precision/recall performance of the system and report on the results below.) We want to involve users in improving the quality of the content. Since PHOAKS is a web-based system, this means involving unknown users at remote locations. And in contrast to most end-user modification facilities, changes made by one user will affect all users, thus making social issues important.

There have been several previous research efforts that have attempted to involve end users in the remote design and evaluation of systems as they are in use. Girgensohn, Redmiles, and Shipman [4] report on a technique called "expectation agents" that allows designers to represent their expectation about how end users will use their system. As the system is used, an active agent monitors user behavior for mismatches with these expectations. When a mismatch is detected, the agent may notify developers and give users a chance to communicate with the developers. The expectation agent approach differs from our approach in assuming that developers can state their expectations for how the system may be used. Although we are not sure how often this assumption is true, this assumption relieves users of the need to decide explicitly that feedback is appropriate, and it could lead users to deliver feedback just when it is most appropriate.

Hartson, Castillo, Kelso, and Neale [6] report on an approach they call "remote evaluation". Data is constantly logged while a system is being used. Then, when users experience "critical incidents", roughly breakdowns, they can create a report which will be sent to system developers or professional evaluators, along with the log of activity immediately preceding the critical incident. One assumption of this approach is that users can identify critical incidents; Hartson et al present some evidence that they can. However, we have chosen not to impose any such a priori structure on user feedback; rather, users can send free-form feedback whenever they choose.

Both the expectation agent and remote evaluation work differ from ours in three important respects. First, they focus on system design and usability; while we too exploit user feedback for these general issues, we focus on involving users in designing system content. Further, we are interested in *automated* mechanisms for user involvement in content design, moving our work in the direction of end-user modification. Finally, we have tried out our ideas in a deployed system with several thousand daily users and have received feedback from hundreds of users. As far as we know, neither of the other previously mentioned approaches [4, 6] has advanced beyond the idea and prototype stage.

THE PHOAKS SYSTEM

Empirical Basis

The PHOAKS application attacks the problem of extracting recommendations of web resources (URLs) from Usenet messages and creating interfaces to the recommendation data. It builds on work [10, 17] that provides empirical evidence that Usenet messages are a useful source of resource recommendations. To summarize these results:

- Usenet messages often mention pointers (URLs) to web resources.
- A significant portion of resource mentions are done for the purpose of recommending a resource.
- Recommendation instances can be recognized accurately by machine.
- Some resources recommendations are confirmed by more than one person.
- The number of distinct recommenders of a resource is a plausible measure of resource quality.

What PHOAKS Does

PHOAKS searches netnews for mentions of web resources, finding about 39K each day. It applies rules that categorize each mention. The most important categories are recommendations and contact pages, since these are the resources presented in the interface. PHOAKS maintains a database of recommended resources and associated contextual information, and generates web pages as an interface to the recommendation data.

How PHOAKS Does It

PHOAKS consists of a general architecture (see [18] for details) for filtering information from electronic messages and a set of techniques for generating and managing dynamic web-based interfaces. The architecture consists of three main processes:

- *search* — search messages for a specified pattern (such as "http://") and extract contextual information surrounding each instance of the pattern,
- *categorization* — apply rules that classify each instance of the pattern (e.g., URLs used as recommendations vs. personal home pages), and
- *disposition* — process the categorized information in some way (e.g., store it in a database or fetch the content of a URL).

The most important process for the purposes of this paper is categorization, since it primarily determines the quality of content on PHOAKS pages, and the page content is the major object of user design.

PHOAKS categorizes a mention of a web page as a recommendation if it passes a number of tests. First, the message must not be cross-posted to too many newsgroups. Messages posted to a large number of groups are so general that they are not likely to be thematically close to any of the groups, and PHOAKS attempts to traffic in thematically relevant recommendations. Second, if the URL is part of a poster's signature or signature file, it is not counted as a recommendation. Third, if the URL occurs in a quoted section of a previous message, it is ruled out. Fourth, if the textual context surrounding the URL contains word markers that indicate that it is being recommended and does not contain makers that indicate that it is being advertised or announced, then it is categorized as a recommendation. We have developed rather complicated categorization rules that implement this basic strategy to distinguish the different purposes for which web resources are mentioned.

There are two aspects to rule accuracy: *precision* (the percentage of data that the rules classify into a certain category that actually belong to the category) and *recall* (the percentage of data that belong to a category that the rules actually classify into that category). In our most recent validation study, the rules for recognizing recommendations had a precision of 88% and a recall of 87% (on a sample of 526 url mentions, with an inter-rater reliability of 88%).

After a web resource is categorized as a recommendation, it is fetched (i.e., downloaded) before being included in a PHOAKS web page. This is necessary to verify that the URL is valid (e.g., that it was not a typo) and to obtain the title to use on the PHOAKS web page.

The PHOAKS Web Site: Scope and Status

The PHOAKS web site (http://www.phoaks.com/phoaks/) has been up since February of 1996. It contains about 37,000 pages of recommendation data for about 1500 newsgroups. About 400,000 people used PHOAKS during its first year, and, as of February 1997, about 3000 people use PHOAKS each weekday.

What Users See

The interface presents a tightly interlinked set of pages. The major page types are:

- *resource summaries* — information about resources, number of recommenders for each resource, and the frequency and recency of recommendation.
- *recommender summaries* — frequent recommenders and the resources they recommended.
- *resource pages* — people who recommended a particular resource.
- *recommender pages* — resources recommended by a particular person.
- *message pages* — contextual information from a message in which a resource was recommended.
- *index pages* — summaries of "internal nodes" in the news hierarchy (like rec.music), used for navigation.
- *feedback pages* — users can add links to the site, give opinions on links already present, and express comments about PHOAKS.

Figure 2 shows the resource summary page for the newsgroup rec.music.dylan.

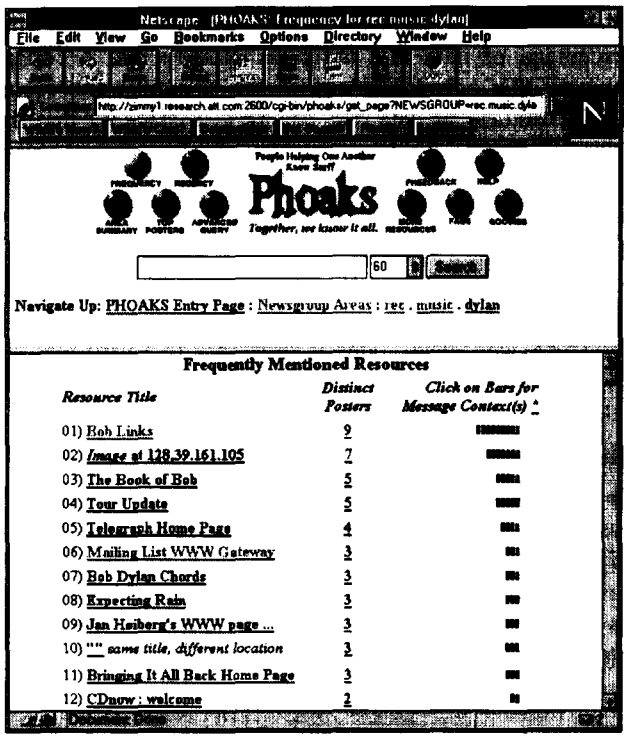

Figure 2: PHOAKS Resource Summary Page

Design Rationale

The PHOAKS interface design embodies three aims. First, we want to make the conversational context of the extracted information (e.g., recommended web resources) a resource for users. Second, we want to foster community among both the original conversational participants and users of our interface. Third, we want to achieve both goals while respecting the privacy concerns of the original conversational participants.

The PHOAKS interface ranks resources by the number of distinct recommenders. In addition, users can find out who the recommenders are and other resources they recommended to the newsgroup being viewed. This is useful in several ways. First, regular readers of a newsgroup are likely to know who else in the group has opinions that they trust. Thus, they can investigate just those resources that have been recommended by these people. Second, after investigating and liking a resource, one can find other resources recommended by the people who recommended that resource. PHOAKS limits the scope of this feature to work within a single newsgroup. So, if a poster recommends web resources in two different groups, say talk.politics.drugs and comp.lang.java, PHOAKS provides no way to correlate the activities. That is, we do not allow name searches, so PHOAKS cannot be used to construct a comprehensive view of a person's posting behavior. PHOAKS also respects the **x-no-archive: yes** protocol, for posters who do not want any of their messages archived.

We include hyperlinks to recommender's personal home pages (if we have them) to allow readers to find out more about recommenders. However, we chose not to include "mailto:" links because we thought that by default people who participate in a newsgroup ought not to be easily contacted by people who are not part of the group, especially weeks or months after they posted the message that caught someone's interest.

Users also can access opinions about a resource as expressed in the surrounding message context. This is especially important because our classification rules use only limited semantic information. Another important benefit is that a message may mention many resources, some of which did not make it into our summary (e.g., because we could not verify their existence or because we categorized them as home pages), yet users still may visit them when they encounter them in the message context.

Finally, users can see the timeliness of a resource within a community. A "histogram" of shaded boxes is displayed for each resource, with one box for each distinct recommender — the more recent the recommendation, the darker the box. Thus, one can get an impression of whether a resource has been mentioned a lot recently, whether it has been mentioned steadily over time, or whether it appears to have fallen out of favor.

MOTIVATION FOR USER INVOLVEMENT

When PHOAKS was first made available, the only form of feedback we allowed was email messages. We read, evaluated, and responded to each message manually. Over time it became clear that much of the feedback dealt with the content of the pages. This should not be surprising: what PHOAKS provides are many organized sets of pointers to content (along with associated contextual information), and it is the content that people really care about. Analysis of PHOAKS access records supports this interpretation. People typically don't access many PHOAKS pages at a time; rather, they seem to go to the PHOAKS page for the topic they are interested in, then follow links to the content. So, people don't engage in extended interactions with PHOAKS per se.

And not only is the content that PHOAKS points to of most concern to users, the content also is likely to have some problems since it is automatically generated. At a minimum, since our recommendation rules are not 100% accurate, we expect a small percentage of PHOAKS URLs not really to be recommendations.

Further, detailed content analysis has shown that the concept of recommendation is difficult to define algorithmically; specifically, there are hard, ambiguous, or controversial cases. There are rare cases when a web resource is warned about, rather than recommended. For example, someone posted a message to soc.culture.african-american condemning the contents of a URL operated by a hate group. He was drawing his community's attention to

the hate group, making it aware of it's activities, but was he recommending it? More prosaically, it is sometimes hard to detect when a URL is being advertised by someone with a personal or organizational connection with the URL, rather than being recommended by a disinterested third party. And finally, sometimes a web resource is of only ephemeral interest to a community. For example, when one of the stars of the X-Files was scheduled to appear on the Tonight Show, someone posted a message announcing this that included the URL for NBC. This URL was recommended and was relevant, but only for a short time.

In addition, since content is generated by reusing information that people have posted, it is not surprising that some of the original producers want to express their opinions, both on the content itself and on how it is reused.

Web sites also go dead or move frequently, so a list of 10 good web sites for a topic that is valid today might well be out of date in 3 months.

Finally, PHOAKS contains tens of thousands of recommendations, the vast majority of which have never been verified by a human being. This task clearly is beyond the capability of one or two system maintainers; the only way it will be done is by involving PHOAKS users.

For all these reasons, we began to focus on designing a set of mechanisms that allow users to tailor the content of our pages. Of course, this also necessitates system and interface redesign: the interface must be re-designed to allow tailoring and present its results, and algorithms must be redesigned to take into account information provided by users. We also have made some design changes based on a different type of user feedback, namely analysis of site usage data. When several thousand people a day use a site, patterns can be observed that indicate problems, opportunities for more efficient design, etc.

INVOLVING USERS IN CONTENT DESIGN

What Users Have Told Us

Table 1 show shows the user feedback for each month since PHOAKS has been up. The first column gives the number of email messages. The second and third columns measure use of two specific types of content tailoring that we added in May 1996; we discuss these below. At a purely quantitative level, we note that feedback peaked in August of 1996 and dropped off a bit and leveled out since then. Another perspective is that during the 7 months from July 1996 to January 1997, we received about 2.9 messages per day. In the busiest month we received 3.5 message per day; in the slowest month, 2.5.

Month	*Email messages*	*Added url instances*	*Url ratings offered*
February 1996	35		
March	25		
April	24		
May	54	40	123
June	53	48	74
July	96	90	186
August	110	124	203
September	90	28	75
October	91	67	146
November	80	48	56
December	80	42	41
January 1997	77	45	86
TOTAL	815	532	990

Table 1: Overview of User Feedback

In May 1996, we categorized the email messages we had received to that point. The left half of Table 2 shows the results. 27% of messages were requests for us to add links to other web sites (usually ones maintained by the person sending the email), and 15% were requests for us to update a link that had changed. Almost 15% either told us that a link didn't belong on a particular page or questioned why a link was not there (or more generally questioned or complained about our filters).

These categories of messages all concerned (adding new or modifying existing) content; together they accounted for 57% of user feedback. We set out to design mechanisms that enabled users to modify and add content in an automated or semi-automated manner.

Enabling User Tailoring of PHOAKS Content

In May 1996, we added forms that allow people to volunteer links to associate with the PHOAKS pages for any newsgroup and to offer opinions on the links that already are there. In December 1996 we added a form for submitting link update notifications.

We view what we've done as "end user modifiability on demand". A designer who wants to let users modify a system is faced with the problem of deciding what should be modifiable. This is hard to decide in advance. Rather than doing so, we have been guided by the expressed feedback of our users.

A system that permits user modification must have the appropriate "hooks" for gathering and computing with user modifications. In our case, we had to change our architecture in three ways:

Category, as ranked in Feb 96 — May 96 email	Feb96 — May96		Dec96 — Jan97		Category, as ranked in Dec 96 — Jan 96 email
	#	%	#	%	
Add link request	34	27.42	63	36.63	**Topic related question**
Link update request	19	15.32	21	12.21	***Add link request***
Topic related question	13	10.48	19	11.05	***Link update request***
Compliment on PHOAKS	12	9.68	16	9.30	**Question about or to a person**
Question/complaint about filters	11	8.87	14	8.14	**Compliment on PHOAKS**
Link does not belong	7	5.65	13	7.56	**Commercial/business/advertising**
System (e.g., browser) problem	6	4.84	9	5.23	**Bug, including bad url**
Other	4	3.23	6	3.49	**Other**
Acceptability of PHOAKS	4	3.23	5	2.91	**Acceptability of PHOAKS**
Question about or to a person	4	3.23	2	1.16	**System (e.g., browser) problem**
Commercial/business/advertising	4	3.23	1	0.58	***Link does not belong***
Bug, including bad url	2	1.61	1	0.58	**Functionality request**
Functionality request	2	1.61	1	0.58	**Comment about page design**
Comment about page design	2	1.61	0	0	***Question/complaint about filters***
TOTALS	124	100	171	100	**TOTALS**

Table 2: Categorizing User Email Messages (italicized categories are content related)

- for user volunteered links: we changed our databases to store user-volunteered information and changed the interface to include pages of user-volunteered links.
- for user ratings of links: we have to modify the algorithm for ranking web resources (which currently is based only on number of distinct netnews recommenders and recency of recommendation) to include user ratings. Weighting these ratings appropriately is a difficult problem (which we discuss more below), and we have not yet implemented an algorithm that does so.
- for link updates: we added a database of old-link/new-link pairs and changed the algorithm that generates PHOAKS pages to consult this database.

Have the Content Tailoring Mechanisms Been Used?

The second and third columns in Table 1 show how much the content tailoring mechanisms have been used. First, let us consider URLs added by users. A total of 453 unique URLs were added to 317 groups. Some URLs were added to more than one group; counting an URL one time for each newsgroup for which it was added gives a total of 532 added URLs.

Second, let us look at the opinions people offered on existing URLs. The interface lets people express two opinions for an URL, either that it is good or that it does not belong. We chose the description "does not belong" rather than "bad" to emphasize that it was not so much the quality of a site we were after as its relevance to the theme of the containing newsgroup; also, quality judgments are more subjective. That is, if we are considering the X-Files page, our primary goal is to get rid of sites that are not about the X-Files (present perhaps because of a spamming message or because our algorithm miscategorized a signature URL) than low-quality X-Files sites.

990 opinions were offered for 798 distinct URLs in 138 groups. 660 of the opinions were that an URL was good; 384 were that the URL did not belong.

It is too soon to give any real quantitative data about user-submitted link update notifications. We note, however, that unlike the other two mechanisms, the link update mechanism is not completely automated. Instead, users fill out a form that results in email being sent to us. We then evaluate the newsgroup, old url, new url, and user email address to determine whether it is an appropriate update. We did it this way because the possibility for malicious misuse (e.g., submitting a bogus update for an URL maintained by someone you do not like) seemed large. However, all of the approximately 40 link update requests we have received so far have been legitimate.

One conclusion is that the content tailoring mechanisms are not used much: no tailoring has been done for most of the 1500 or so groups on the PHOAKS site. However, this is not really surprising: people come to the PHOAKS web site to find information, not to do extra work like rating information. (Our principle of role specialization leads us to believe that most people do not want to do such extra work). Perhaps the more important question is: can

tailoring from remote, anonymous users be trusted? Do they suggest on-topic links to be added? Do they really tell us which links are not appropriate for a topic? We next describe some analysis we have done into these questions.

Can User Content Tailoring Be Trusted?

In order to address this question, we picked a random sample of roughly 10% of the user-volunteered links and the ratings offered by users to analyze.

A total of 453 unique links were volunteered for 317 groups. We evaluated a random sample of 59 links distributed over 21 groups to see whether (in our best judgment) they were on-topic. We judged 44 to be on-topic, we were unsure of 2, and the remaining 13 were dead links (in some cases, the link may once have been valid; in other cases, the user who typed in the link had clearly made an error). Thus, it appears that the information users add is likely to be on-topic. This is not a surprising result, since the primary reason for adding information seems to be to advertise one's own site. And it also is not all that useful, since PHOAKS does not face the problem of too little information (for the newsgroups/topics that it covers, that is; clearly, we do have too little information about topics for which there is no newsgroup or which PHOAKS does not monitor).

The problem PHOAKS does face is information quality. Rather than having 40 links for a topic, some good and some poor or irrelevant, it would be much better to have only 10 high-quality, on-topic links. We hoped that opinions from users would help us to eliminate off-topic and low-quality links. Thus, in evaluating user ratings, we considered only "does not belong" opinions. Users have submitted "does not belong" ratings for 286 URLs distributed over 66 newsgroups. We randomly selected 29 URLs from 10 groups to analyze. We evaluated the links to see whether we agreed that they did not belong.

The results were not encouraging. We judged that 9 links did not belong (i.e., we agreed with the user ratings), we were unsure about 10 links, and we believed that 10 links definitely did belong (i.e., we disagreed with the user ratings). (In the third case, is possible that the users who submitted ratings thought the links were on-topic, but of low quality; since we are not topic experts, we cannot be sure.) Even if we assumed that all the links we were unsure about did not belong (which is quite possible), this still would mean that just over a third of user ratings are questionable. It would appear that systems that accept feedback from remote, anonymous users must expect and deal with a significant amount of untrustworthy input.

We did try one other analysis. In analogy to our reliance on number of distinct netnews recommenders as an estimate of URL quality, we decided to examine URLs that had received "does not belong" ratings from multiple users. We found only 12 such URLs, 1 that three users said did not belong, and 11 URLs with two such users. These results were more encouraging. Only one of these URLs appeared to be on-topic, 6 definitely were off-topic, and we were unsure about the other 5 (although we leaned toward believing they were off-topic). However, even if we were to take multiple negative ratings as a perfect measure, such feedback appears to be extremely rare, too rare in fact to be of any practical importance.

Does The Availability of Content Tailoring Mechanisms Reduce User Email?

One goal of adding content tailoring mechanisms was to automate certain tasks and thus relieve the burden on the system designers. Was this goal achieved? The right half of Table 2 shows the results of categorizing the most recent user email. The percentage of messages accounted for by the content-related categories "add link request", "link update request", "question/complain about filters", and "link does not belong" declined from 57% to 24%, so there is some evidence that users who want to give these types of feedback are using the automated mechanisms rather than sending email. The categorization of the most recent email contained a few surprises, too.

Surprises in User Feedback

The number of "topic related questions", messages about the topic dealt with by a newsgroup, particularly surprised us. Among the many examples are messages sent from the PHOAKS pages for:

- rec.antiques, asking for information about pricing and selling antique adding machines;
- rec.video, asking for help in setting up a camcorder as a video source for a PC videophone;
- rec.music.indian.classical, asking for help in finding music notation for playing carnatic ragas on an electronic keyboard.

Another surprising type of message was what we describe as a "question about or to a person". The people who sent such messages either wanted to contact an individual mentioned on a PHOAKS page (often because they shared the same last name and were searching for relatives), or thought they were in fact sending email to an individual mentioned on a PHOAKS page.

Both categories pointed out several things to us. First, they suggest that our interface is not clear. PHOAKS reuses and repackages information produced by many different individuals. It was one of our design goals to make this fact apparent, i.e., to communicate the context in which recommendations were produced, including some information about the producer. However, it seems that we may have achieved this goal at the expense of obscuring the role of computational mechanisms in extracting information from its original sources.

Second, they remind us how much people want to talk to other people. We might call what's going on here "information-based introduction". It's the opposite of

social filtering: rather than using people to filter information, these PHOAKS users used (or tried to use) information as a pointer to appropriate people. Note that there is a tension between this desire and the PHOAKS design goal of preserving the privacy of the original producers of information. If we had included a mailto: link to each information producer, we would have greatly helped information seekers; however, we were pretty sure that at least some information producers would not be happy about this sort of contact.

We have made a simple change to our feedback interface in response to these categories of messages. Now, clicking on a "Send us email" link yields a page of text that reminds users of the type of automated feedback they can give us and that we are not experts in the topics mentioned on PHOAKS pages. At the bottom of this page is an actual mailto: link. We made this change very recently, so we are not yet sure how effective it will be.

Alternatives to Explicit User Feedback

Given the limits on what we can expect from explicit user feedback, it is important to consider some alternatives.

First, sometimes user feedback indicates a problem that can be solved partly by user participation and partly by automation. We have received a few messages from users pointing out dead links on some of our pages. When we receive these messages, or if users submit "does not belong" ratings for dead links, we can remove the links immediately. However, we also have modified our architecture to re-fetch all links periodically and remove any links that are not accessible at the time.

Second, patterns of access can be a valuable source of information. This insight is in the tradition of "computational wear" [7, 8, 20]. A simple example is to observe which types of pages are accessed most: in our case, it turns out to be resource summary and index pages. In the new PHOAKS architecture, pages are dynamically generated by a cgi script, then cached. However, since we know that the most heavily accessed pages are resource summary and index pages, we run queries to generate and pre-cache these pages fresh every time we complete a run through netnews and update our databases. Therefore, the most accessed pages always already exist when they are requested, speeding their delivery to users.

A more interesting use of access data that deals directly with page content is based on "jump-off" information, a record of each click on a link on a PHOAKS page. By analyzing this data, we can determine that, on average, the 1st ranked link on a page is clicked on (say) 35 out of 100 times, the 2nd ranked link 20, the 3rd ranked link 15, etc. Now, if we see that on some pages the 2nd ranked link is never clicked on, or the 8th ranked link is clicked on 25 times out of 100, we know this is anomalous, and we can investigate these links.

DISCUSSION: ISSUES IN INVOLVING USERS IN CONTENT DESIGN

We have opened up the design of PHOAKS content to a multitude of anonymous, remote users. This seems to fits well with the PHOAKS approach, based as it is on using information gleaned from millions of Usenet messages posted by hundreds of thousands of people. However, newsgroups feature social pressures that to a certain extent constrain the topic and quality of posted information. We cannot be sure a priori whether equivalent constraints will govern user tailoring of PHOAKS content. We can identify a number of important issues that arise in this situation.

First, and most fundamental, we don't even know the identity of our users. This makes it impossible to institute a "one person one vote" protocol.

Second, and perhaps most important, is the credibility of remote users. To what extent can we trust people to tailor shared information content? Our analysis of user submitted ratings cannot make us optimistic.

Third, even if all the feedback we receive were credible, is it representative? Probably the people who offer feedback are just the "squeaky wheels"; we are not sure whether the opinions of the 800 or so people who have sent us email are typical of the 400,000 people who have used PHOAKS.

Fourth, how should we weight the opinions of PHOAKS users versus the original producers of recommendations? For example, options for weighting a "does not belong" rating range from decrementing the score of a web resource to disqualifying it.

Fifth, how can we differentially empower users to modify content? For example, we may want to give modification privileges to original producers of PHOAKS content or active participants in a newsgroup or frequent visitors to PHOAKS pages. One way to solve the problem of user credibility is to identify a small minority of users who are highly likely to be credible.

Sixth, how can we communicate "rough draft" status of PHOAKS information and then communicate "polished draft" status to pages once a significant number of readers have corrected any machine classification errors? More generally, when pages contain a mixture of information produced in Usenet messages, organized by PHOAKS algorithms, and modified by PHOAKS users, how do we make the source and status (and thus, perhaps the merit) of all the information clear?

REFERENCES

1. Ackerman, M.S. & Malone, T.W. Answer Garden: A tool for growing organizational memory, in Proceedings of the Conference on Office Information Systems, ACM Press, 31-39.
2. Ackerman, M. and McDonald, D., Answer Garden 2: Merging organizational memory with collaborative help, in Proceedings of CSCW'96 (Boston MA, November 1996), ACM Press, 97-105.
3. Allen, R.B. User models: Theory, method and practice. International Journal of Man-Machine Studies, 32 (1990), 511-543.
4. Girgensohn, A., Redmiles, D.F., and Shipman, F.M. Agent-based support for communication between developers and users in software design, in Proceedings of KBSE'94 (Monterey CA, September 1994), IEEE Computer Society Press, 22-29.
5. Goldberg, D., Nichols, D., Oki, B.M. and Terry, D. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35, 12 (1992), 51-60.
6. Hartson, H.R., Castillo, J.C., Kelso, J., and Neale, W.C. Remote evaluation: the network as an extension of the usability laboratory, in Proceedings of CHI'96 (Vancouver BC, April 1996), ACM Press, 228-235.
7. Hill, W.C., Hollan, J.D., Wroblewski, D., and McCandless, T. Edit wear and read wear, in Proceedings of CHI'92 (Monterey CA, May 1992), ACM Press, 3-9.
8. Hill, W.C. and Hollan, J.D. History-enriched digital objects: prototypes and policy issues. The Information Society, 10 (1994), 139-145.
9. Hill, W.C., Stead, L., Rosenstein, M. and Furnas, G. Recommending and evaluating choices in a virtual community of use, in Proceedings of CHI'95 (Denver CO, May 1995), ACM Press, 194-201.
10. Hill, W. C. and Terveen, L. G. Using frequency-of-mention in public conversations for social filtering, in Proceedings of CSCW'96 (Boston MA, November 1996), 106-112.
11. Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A. and Cohen, M.D. Intelligent information sharing systems. Communications of the ACM, 30, 5 (1987), 390-402.
12. Maltz, D., and Ehrlich, K. Pointing the way: active collaborative filtering, in Proceedings of CHI'95 (Denver CO, May 1995), ACM Press, 202-209.
13. Muller, M. J. and Kuhn S. (eds.). Participatory Design. Communications of the ACM, 36, 4 (entire issue).
14. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. GroupLens: an open architecture for collaborative filtering of netnews, in Proceeding of CSCW'94 (Chapel Hill NC, October 1994), ACM Press, 175-186.
15. Shardanand, U., and Maes, P. Social information filtering: algorithms for automating "word of mouth", in Proceedings of CHI'95 (Denver CO, May 1995), ACM Press, 210-217.
16. Terveen, L.G., Selfridge, P.G., and Long, M.D. Living Design Memory: Framework, Implementation, Lessons Learned. Human-Computer Interaction, 10, 1 (1995), 1-38.
17. Terveen, L.G., Hill, W.C., Amento, B., McDonald, D., and Creter, J. PHOAKS: a system for recognizing and sharing recommendations. Communications of the ACM, 40, 4 (1997), 59-62.
18. Terveen, L.G., Hill, W.C., Amento, B., McDonald, D., and Creter, J. Building task-specific interfaces to high volume conversational data, in Proceedings CHI'97 (Atlanta GA, March 1997), ACM Press, 226-233.
19. Wittenburg, K., Das, D., Hill, W., Stead, L. Group asynchronous browsing on the world wide web, in Proceedings of the 4th International WWW Conference, (Boston MA, December 1995), O'Reilly & Associates, 51-62.
20. Wroblewski, D., McCandless, T., Hill, W. Advertisements, Proxies and Wear: Three Methods for Feedback in Interactive Systems, in Dialogue and Instruction, Beun, R., Baker, M., and Reiner, M. (eds.), 1995, Springer-Verlag, 336-347.