NuggetMine: Intelligent Groupware for Opportunistically Sharing Information Nuggets

Jeremy Goecks[1]
Graphics, Visualization, and Usability Center
Georgia Institute of Technology
Atlanta, GA 30332, USA
jeremy@cc.gatech.edu

Dan Cosley
Dept. of Computer Sciences and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
cosley@cs.umn.edu

Abstract
NuggetMine is an intelligent groupware application that collaborates with a workgroup to increase information nugget sharing among the group. Information nuggets are small amounts of self-contained information, such as the URL of an interesting news article, a book title, or the time and location of a local art event. NuggetMine and the workgroup work together to build, maintain, and utilize a repository—or "mine"—of information nuggets. Group members submit nuggets to NuggetMine, which organizes and augments the submitted nuggets and provides a desktop interface to each group member. This interface makes it easy for group members to submit nuggets, view nuggets, and explore the mine. NuggetMine distributes the tasks necessary to share nuggets between it and the workgroup so as to best utilize the skills of each collaborator. In this paper, we describe the NuggetMine application and interface and present a pilot study of the application.

Keywords
Intelligent groupware, information nuggets, information sharing, human-computer collaboration, associative networks, informal communication, lightweight interfaces

INTRODUCTION
Picture this scenario, which often occurs in a research lab. Two researchers, Joe and Kathy, are conversing about their current research. Upon hearing about Joe's latest ideas, Kathy mentions that she recently read an article on the Internet that challenges some of Joe's ideas; she recommends that Joe read the article. She notes that an upcoming lecture sounds like it might pertain to Joe's ideas as well. Joe makes a note about the two pieces of information that Kathy gave him and thanks Kathy for the information, and they move on to discuss another topic. The two pieces of information that Kathy conveyed to Joe—in essence, the URL of an article and awareness of a local event—are quite small and are useful and pertinent outside the current context; we define such pieces of information as information "nuggets".

As the above scenario illustrates, information nuggets are a common and valuable currency among people. However, face-to-face conversation does not always support sharing of these nuggets. We can all recall times when we discovered too late that a coworker possessed a useful nugget of information that never came up in conversation. Likewise, we can also recall instances where a nugget did come up, but we forgot it or never acted on it. (How often has someone told you "you've got to see this movie"? Did you?) Finally, we have all experienced situations in which only part of a group knows an interesting tidbit of information because "everyone knows that". These instances indicate that conveying information nuggets only during conversation limits nugget sharing among a workgroup and limits individual usage of shared nuggets.

We created NuggetMine in an effort to address these limitations. NuggetMine is an intelligent groupware application that collaborates with a workgroup to increase information nugget sharing among the group. NuggetMine and the workgroup work together to build, maintain, and utilize a repository—or "mine"—of information nuggets. Individual group members submit nuggets to NuggetMine when they encounter them; the interface encourages submissions by minimizing the amount of effort and information required to contribute a nugget. When a user submits a nugget, NuggetMine fills in information missing from a nugget submission, such as the title of a URL. It then integrates the nugget into the associative networks that NuggetMine maintains. These networks are based on simple, intuitive associations (e.g. author, keyword) which users can exploit to explore the repository.

NuggetMine makes it easy for users to view and access nuggets in the repository. NuggetMine displays individual nuggets opportunistically and unobtrusively via a scrolling desktop interface. When a user sees an interesting nugget, he can easily stop the scrolling to view the nugget. In addition, NuggetMine provides access to nuggets related to

IUI'02, January 13-16, 2002, San Francisco, California, USA.

[1] This research done while at the University of Minnesota Dept. of Computer Science and Engineering.

the current nugget via the associative networks mentioned above. Hence, for each nugget that scrolls across the desktop interface, the user has ready access to that nugget *and* all its related nuggets. In sum, NuggetMine increases information nugget sharing among a workgroup by working with group members to create a mine of nuggets and by providing an interface for group members to easily view nuggets in the mine.

WHY NUGGETMINE EXISTS

Short, informal communication plays an integral part in workgroup productivity. Group members use informal communication to quickly share information and to maintain a shared knowledge base [26][14]. Much research has focused on understanding and developing software applications that augment such communication within a workgroup. There are studies of common applications that support group communication such as email [27] and instant messaging [18]. Experimental groupware applications such as TeleNotes [28] and NotePals [5] have been developed to provide additional support for informal communications such as context management and note sharing, respectively.

However, we observe a void in the space populated by existing groupware applications that augment informal workgroup communication. This void is due to the fact that existing groupware lacks support for opportunistic, asynchronous sharing of information nuggets. An application filling this void—a groupware application that opportunistically and asynchronously captures and displays nuggets—would help a workgroup better share and utilize nuggets known by individual group members.

We do not differentiate between nuggets that address work-related topics and those that do not because we believe both types are valuable. Sharing nuggets related to work topics increases the knowledge of the workgroup, and an increase in workgroup knowledge very often helps the group to be more productive. Sharing nuggets unrelated to work provides group members with the opportunity to share their recreational interests with the group and find like-minded members with whom to discuss these interests. Thus, "off-topic" nuggets provide a means for group members to develop additional communication channels to other people and ultimately increase social capital within the group.

Conversations Are Non-Optimal for Nugget Sharing

In the introduction we discussed a scenario where Kathy, a researcher, shared some information nuggets with her colleague Joe during a conversation. Consider the process by which Joe acquired the nuggets which Kathy possessed. First, Joe and Kathy started conversing; second, their conversation turned to a topic that was related to some nuggets Kathy knew; third, Kathy consciously recalled the nuggets when they were relevant; fourth, Kathy conveyed the nuggets to Joe; fifth, Joe remembered or recorded the nuggets; and finally (a step we didn't discuss in the introduction), Joe acted on the nuggets. (e.g. Joe found the story on the Internet that Kathy mentioned and attended the lecture Kathy discussed). Figure 1 summarizes this process.

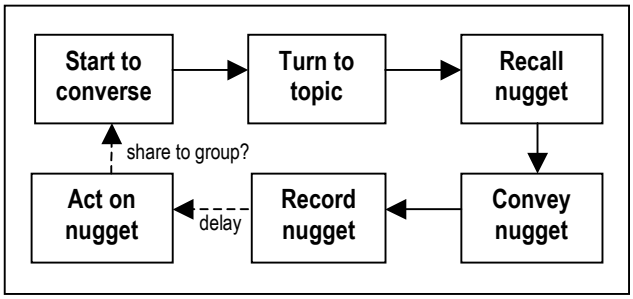

Figure 1. Information nugget sharing via conversation. Problems include the synchronous chain of steps, the delay between recording and acting on a nugget, and the need to repeat the process for all group members.

Although the process was successful in the above scenario, it is clear from this discussion that it has many weaknesses and thus is quite fragile. Most of the steps in the process are synchronous. They must occur in the order given and failure at any step results in failure to share the information. The process is asynchronous in a particularly bad place: there is a delay between remembering and acting on the nugget, which may cause it to go unused. Recording the nugget is also problematic, as human memory is fallible and cocktail napkins may be lost. Finally, most workplace conversations are not meetings; rather, they involve only two people and are short and informal [26]. Hence, to propagate the nugget to all interested group members, the nugget-sharing process must happen many times without failure during these short, two-person conversations. Such repetition is unlikely given the weaknesses just described.

Beyond the first nugget shared, there are often additional nuggets related to the first nugget that might be shared as well. Nuggets can be related by content (e.g. keywords, category) or by attribute (e.g. time captured, author), and people often use such associations to acquire and remember information [23]. Thus, individuals likely have a better chance of remembering a number of related nuggets if they are all shared during the same conversation. In order to share several related nuggets during a conversation, however, the "recall nugget" step and its subsequent steps must occur several times. As discussed above, this repetition has no guarantee of success. In addition, group members not involved in a particular conversation may know additional nuggets related to the current nugget. If the members are not participating in the conversation, these nuggets will go unshared because they are unaware that anyone is interested in them.

This analysis indicates that conveying nuggets during informal conversation is a tenuous process that can fail at many points and in many ways. Over the course of many conversations within a workgroup, the failure to share nuggets becomes significant; ultimately, the workgroup significantly underutilizes the information nuggets its group

members possess because group members do not share nuggets as effectively or efficiently as they might.

NuggetMine to the Rescue

NuggetMine addresses a number of the issues discussed above in an effort to increase nugget sharing among a workgroup. It provides an intelligent, lightweight desktop interface that supports three main tasks. First, users can quickly and easily submit nuggets to the "mine" while they are working at their desktop computer. Second, the interface displays captured nuggets unobtrusively to the user when she is working at her desktop, allowing her to become aware of nuggets at opportune times without disturbing her work. Finally, NuggetMine automatically induces content-based and attribute-based associative networks among captured nuggets; the interface uses these networks to help users explore the mine of captured nuggets and find nuggets of interest.

The essence of NuggetMine is in its collaboration with the workgroup to achieve the goal of sharing nuggets within the group. In particular, it distributes the tasks necessary to enable sharing so as to best utilize the skills of each collaborator. Group members, who know what information is interesting and relevant to the group better than NuggetMine does, locate information nuggets and submit them to NuggetMine. NuggetMine, which is faster, more patient, and about as good as humans at organizing and disseminating information, augments and organizes the submitted nuggets and makes the submitted nuggets readily available and navigable.

Contrast the process of using NuggetMine to share a nugget, illustrated in Figure 2, with that of communicating a nugget in conversation. Sharing a nugget via NuggetMine requires fewer steps. Users do not need to recall a nugget at a certain time during a certain conversation; instead, they simply submit a nugget when they remember or discover it. Also, since the nuggets go into a repository accessible to the entire group, the problems of sharing a nugget among the group through repeated conversations disappear. For recipients, there is no burden to record/remember and then later act on a nugget, as the recording is automatic and viewing nuggets is asynchronous with respect to sharing them. Users can opportunistically view nuggets at any time and, if one is interesting, immediately pursue it—as well as any nuggets related to it.

Finally, it is important to point out that NuggetMine stimulates nugget sharing differently than does face-to-face conversation. Conversation stimulates nugget sharing via association: people recall and share nuggets associated with the current conversational topic. NuggetMine does not work this way; instead, it simply asks users to remember that it is always available and ready to accept nuggets. Since NuggetMine provides a complementary way to share nuggets, we believe that its users will share more information than they would otherwise share. We also believe that it is generally simpler for users to remember

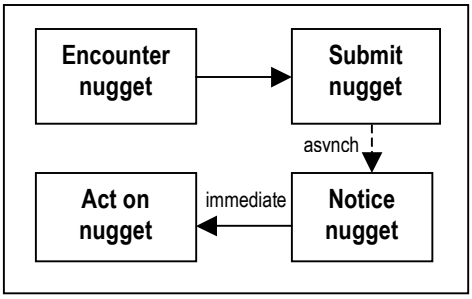

Figure 2. Nugget sharing in NuggetMine. Advantages include the asynchronous nature of submitting vs. viewing nuggets, the automatic recording of nuggets, and the ability to immediately act on a nugget.

that NuggetMine is always available and submit nuggets to it immediately rather than to commit nuggets to their memory and recall them during relevant conversation. Finally, we believe that NuggetMine would be even more effective if it also provided an interface for capturing nuggets that do come up in conversation. In this paper, however, we focus on a desktop interface for capturing nuggets that come to mind not during a conversation but during a user's day-to-day activities, nuggets that would otherwise be lost. We briefly discuss interfaces for capturing conversational nuggets in the future work section.

THE NUGGETMINE INTERFACE

We developed NuggetMine with the objective of creating a groupware application that collaborates with a workgroup to collect and use interesting information nuggets. Several design goals fall out of this objective; we discuss these goals below. We then explore how users interact with NuggetMine, focusing on how the interface expresses these goals while supporting users' interactions.

Design Goals

Our belief is that for users to make effective use of NuggetMine, its interface must be a natural and intuitive part of their work environment. "Natural" and "intuitive" are famously vacuous terms in user interface design, so we refined them to include five primary characteristics:

- **Integrated**. NuggetMine's interface must feel like an integrated part of the work environment, rather than a separate application that users must start, attend to, make use of, and stop.
- **Lightweight**. The interface must require a minimal amount of effort for the services it provides.
- **Complementary**. The interface must perform tedious or difficult tasks as much as possible, while still giving users control over the outcome.
- **Opportunistic**. The interface must not intrude on or otherwise interrupt users. It should work with their natural behaviors to provide services when appropriate.
- **Simple**. NuggetMine provides a basic, simple service, and its interface should have controls and features to

match. Too many controls or options would make the interface less lightweight and increase user effort.

The Primary Interface

Figure 3 shows NuggetMine's normal interface at actual size (about 320 by 110 pixels). The interface cycles through a series of potentially interesting nuggets at a user-defined interval; this cycling behavior is somewhat similar to ticker-tape news displays such as the My Yahoo! Ticker.[2] We chose to model nuggets as either URLs or small pieces of text because most nuggets that come up in conversation are either pointers or small facts. For each nugget, the interface displays summary information (the nugget's "headline"). Clicking on a nugget headline brings up the nugget's URL or text in a web browser. Users can also expand the interface in order to get more information about the nugget and access other nuggets related to the current nugget. Finally, users can easily submit nuggets to NuggetMine by dragging and dropping them on the interface. We discuss this operation in detail below.

This interface supports all five of our goals. It is integrated, as the small window runs continually in the background. It is lightweight and simple, with few controls and no actions that take more than one or two mouse clicks. It is complementary, choosing a set of potentially interesting nuggets to display while allowing users to decide what *is* interesting and giving them power to find out more. The interface is also opportunistic. It waits in the background for users to take natural breaks in their work, hoping to catch their interest in a non-obtrusive way rather than interrupting them with what it thinks is useful information.

The cycling of fresh, personalized nuggets enhances the opportunistic and assistive qualities of the interface, making the most of the times when users do glance at it. The current personalization is crude: NuggetMine displays new nuggets first, then a selection of nuggets sampled from all the nuggets stored with a bias toward newer nuggets and nuggets that have been viewed more often. Ideally, the interface would be smarter when personalizing nuggets. For instance, it might build a keyword profile based on nuggets viewed by a user and use that profile to choose nuggets.

Capturing and Organizing Nuggets

Our primary goal when capturing nuggets was to make the process lightweight. We believe that every additional bit of effort required to submit a nugget results in a drop in the number of nuggets submitted and hence decreases the utility of the NuggetMine. To overcome users' inertia to share nuggets, we condensed the capture of a nugget to its essence by utilizing the drag-and-drop interaction technique. When a user wants to submit a nugget, she simply drags-and-drops some text data or a URL onto the interface. This brings up the capture interface illustrated in

[2] http://my.yahoo.com/ticker.html

Figure 3. The NuggetMine interface in its normal, unobtrusive mode (actual size).

Figure 4, pre-loaded with the dragged data. (Clicking on the main interface's Submit Nugget button also brings up the capture interface, but the data field is blank.) She can now submit the nugget with one click on the Submit button. The interface also allows her to categorize the nugget and to relate it to the nugget that was displayed when she captured the new nugget.

In addition to being lightweight, the capture interface also expresses our other goals. The process of capturing nuggets is both integrated and opportunistic. Users can submit nuggets whenever they come across them during their everyday activities, regardless of what they are doing or whether a nugget is immediately relevant to anyone in the group. Capturing nuggets is simple, requiring only a drag-and-drop and a mouse click. However, we have been unable to develop or locate a GUI widget that affords the drag-and-drop capabilities of the interface. The target icon from Figure 3 and word-of-mouth were the best options we could think of.

Organizing Nuggets

One key feature of NuggetMine is that it builds a repository of nuggets. We would like users to be able to explore this repository via associations; for example, if a user sees a nugget about the IUI 2002 conference, he should be able to easily discover a nugget about an upcoming AAAI conference that is in the repository. However, forcing users to manually create relationships between nuggets, or even to specify several attributes of a nugget, would violate all of our design principles.

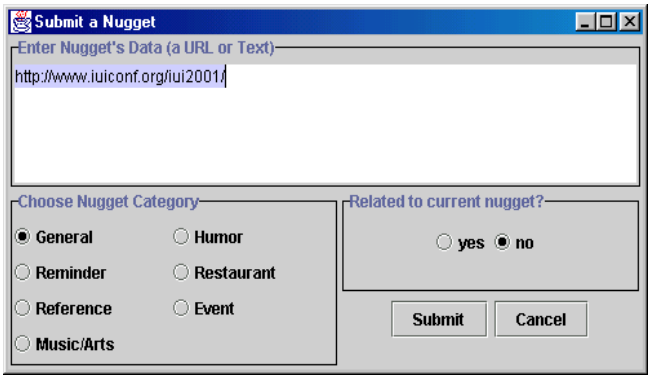

Figure 4. The nugget submission interface (not actual size) requires only one click to submit nuggets after a drag-and-drop.

Instead, NuggetMine creates these associations and records these attributes on its users' behalf. When a user submits a nugget, the NuggetMine interface sends the nugget to a server. The server records the submitter and time of submission. It automatically assigns the nugget a headline; the nugget headline is the page title for URLs or the first few words of the nugget for text data. It also selects relevant keywords for the nugget using Salton's TF/IDF algorithm [22] and revisits each URL every few weeks to recompute the keywords. Finally, if the user specified a category or a relationship with another nugget, the server records this information as well.

We considered allowing users to manually specify the keywords and headline. We also debated about whether to allow users to submit both a nugget and some accompanying text. For example, someone might want to submit a URL along with a sentence or two about why people should view it. This was tempting, but our primary goal was to make the process of submitting nuggets lightweight in order to minimize user effort. We finally decided that even including these features as options in the interface might confuse users and discourage them from sharing nuggets, so we left the features out.

Once the server has determined a nugget's attributes, NuggetMine automatically creates content-based and attribute-based relations between all captured nuggets. Essentially, NuggetMine builds a number of interconnected associative networks, each based on a different relation, which link nuggets in the repository to each other. Users can explore the repository by navigating among and within the associative networks.

Exploring the Repository

A user may access the meta-information captured and assembled by the server by clicking the "More" button to reveal the expanded mode of the interface. Figure 5 shows the interface displaying most of what it knows about a nugget. Users can see the nugget's keywords, author, and time of submission, and use this information to decide whether to pursue the nugget.

NuggetMine also provides hyperlinks to nuggets with similar keywords, to nuggets with similar overall content, to nuggets with the same author, and to nuggets in the same category. Each hyperlink provides access to nuggets in a different associative network, allowing users to rapidly explore the repository via one or more of the networks. Currently, NuggetMine displays the other nuggets in a particular network using a web browser. As the size of the repository grows, this method will become less practical. We envision allowing users to explore parts of an associative network by repeatedly clicking on the same type of link (e.g. repeatedly clicking on the content hyperlink navigates among nuggets in the content network).

These hyperlinks are an important feature of NuggetMine's interface because they support opportunistic and intuitive exploration of the nugget mine. The main interface does not draw attention to itself, but when users do attend to the interface, it can easily provide a large amount of information about a nugget and a gateway into the entire repository. Users can efficiently and easily obtain all nuggets related to the current nugget by using the hyperlinks, yielding much information with little effort. The interface is also simple to use, with all hyperlinks becoming visible after at most two mouse clicks. The simplicity of the interface and use of the web browser to view nuggets help keep the interface lightweight and minimizes the learning curve. Finally, exploring the repository via these networks is simple and natural for users because the associations NuggetMine uses are themselves simple and natural.

RELATED WORK

NuggetMine draws on prior research in several areas. It draws inspiration from work in the areas of human-computer collaboration and interfaces for informal communication. It also utilizes techniques from the areas of information filtering and personalization. We sample from the relevant research in these fields and contrast NuggetMine with this work.

Human-Computer Collaboration

The IUI community has recently begun producing interfaces that support human-computer collaboration. In

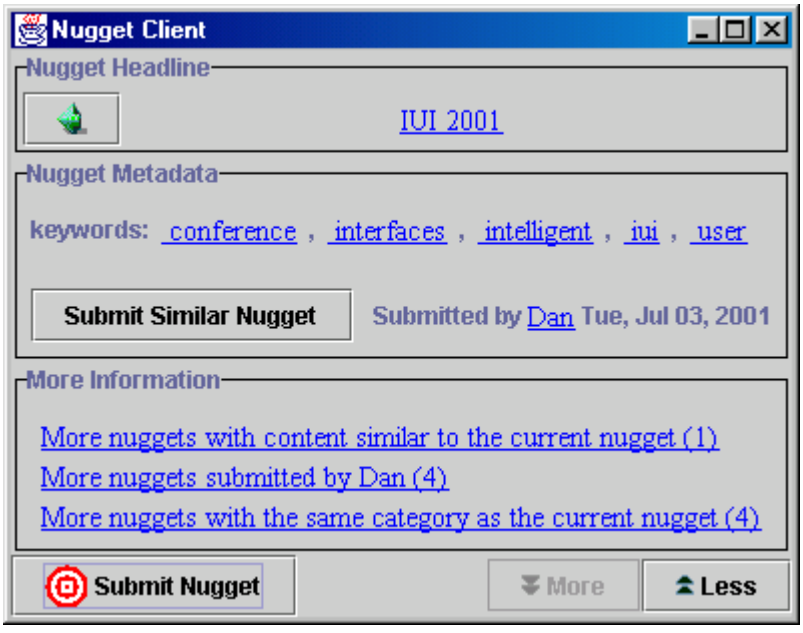

Figure 5. The maximum-size NuggetMine interface (actual size). Metadata tells users more about a nugget, while hyperlinks afford opportunistic viewing of related nuggets.

such collaboration, both the computer and the user work together to perform tasks that are essential to realizing the ultimate goal. Collagen [21] views human-agent collaboration through the lens of discourse theory and uses an API to support such collaboration in the general case. Allen and colleagues have developed mixed-initiative systems that support collaborative planning by the user and the system via natural, spoken language discourse [2][7]. These interfaces do not support collaboration with a group and do not support collaboration that extends beyond one interactive session to accomplish a goal. In contrast, NuggetMine collaborates with an entire group of users over many interactive sessions to accomplish the goal of sharing information nuggets. NuggetMine also distributes tasks among itself and its collaborators according to who is best able to perform the tasks; the interfaces discussed above do not distribute tasks this way.

Interfaces for Informal Communication

Informal communication is communication that is unplanned and unscripted; in other words, neither the time and place (unplanned) nor the structure (unscripted) of the communication is established beforehand. An extensive study of lightweight interactions—brief, two-person, informal communications—among members of a co-located workgroup guided the development of the TeleNotes [28] application. The Montage [24] application uses desktop video conferencing to support lightweight interactions between members of a geographically distributed workgroup. Both applications provide a means to communicate a short asynchronous message to another person. These messages appear as a sort of "sticky note" on the recipient's screen. NuggetMine takes a different approach, displaying nuggets opportunistically in the background. NuggetMine also distributes information to an entire group, whereas TeleNotes and Montage facilitate information exchange between only two people at a time.

Information Filtering and Personalization

Researchers have developed a number of web surfing assistants such as WebWatcher [12], Letizia [15], and Margin Notes[19]. These assistants use a single user's interests to recommend pages related to the page currently displayed in the browser. Let's Browse [16] is similar to Letizia but works for groups of users browsing synchronously. NuggetMine also recommends web pages to members of a group, but these pages are contributed by other group members and are viewed asynchronously. NuggetMine has more in common with recommender systems, which use input from group members to make recommendations for other group members [21]. Directly recommending items to other members of the group (as NuggetMine users do when they submit nuggets) is most like active collaborative filtering [17].

PILOT STUDY

We deployed NuggetMine to a research lab at the University of Minnesota to qualitatively assess its benefits and weaknesses. There are six people in the lab, with some undergraduate and some graduate students. All lab members had met the other lab members in person; both authors were lab members. The study took place over two weeks—a limiting factor—because of time constraints. Our goals for this study were to obtain qualitative feedback about (a) how useful NuggetMine was to the group and how we could improve its usefulness and (b) how readily NuggetMine satisfied its design goals.

The feedback we obtained suggests that NuggetMine can fulfill its principal goal of increasing information nugget sharing among members of a group. When asked whether NuggetMine accomplished its goal, one user remarked:

> "Yes. [It] did accomplish its goal of sharing information. I felt that there are often lots of pieces of information that I would like to share with others, but that I do not. Most of these pieces of info have a rather low priority and could even be considered 'trivial.' I feel that the conventional ways of transmitting this info (excluding person-to-person direct contact) add extra weight or importance to this info that I didn't want to convey. ('Ohh, he sent me an email about it… it might be important.')"

Overall, user interviews from this study suggest that NuggetMine was successful and well received by users. Users liked the intelligent features of NuggetMine; in fact, one user felt that NuggetMine's most useful feature were the associative networks that it maintained. This user said:

> "I experienced the nugget client to lie somewhere between bookmarks and instant messaging clients. I am not sure that sharing of information was the key attribute, rather, I felt that the key attribute was the automatic grouping of nuggets. This is the element that both bookmarks and IM clients are missing."

This study illuminated some problems with the NuggetMine interface as well. We may have erred in the direction of too much simplicity. Users wanted more control of the nuggets displayed. Users wanted the option to never see a nugget again and to view only nuggets in a certain category. Users also wanted an 'expiration date' for nuggets so that nuggets relevant only until a certain date (e.g. a conference, a local event) would disappear after that date. These features would probably make it into a second version of the interface, although we still worry about the balance between features and ease of use.

Users consistently said that the most problematic issue for the interface is that the background window sometimes became buried under other windows. The Adjusting Windows technique [3] might be a better way to display nuggets. In Adjusting Windows, the system displays information by temporarily taking over the bottom portion of the active window (similar to the way sports broadcasts display the scores of other games). This interface might prove better, but it is likely that the ultimate solution for

viewing nuggets lies in having several interfaces, both on and off the desktop.

CONTRIBUTIONS

NuggetMine is not overly intelligent and its intelligence is not novel. Building and navigating associative networks has been done before (e.g. [13][9][4]), and most of its IR and personalization techniques are rudimentary. That said, we believe that NuggetMine demonstrates a fundamental but often overlooked IUI design principle and does so in a novel application.

We designed NuggetMine using the following fundamental IUI design principle: the interface should (a) intelligently perform tasks that it can perform well or that users find difficult and/or tedious and (b) help users perform tasks that users excel at or that computers find difficult. In short, the interface should do what it does well, and it should support users in tasks that humans do well. NuggetMine performs only certain tasks intelligently, such as organizing the nuggets contributed by group members. This would be difficult and tedious for group members, but NuggetMine performs the task quickly and accurately. NuggetMine also attempts to intelligently select interesting nuggets to display for each particular user. Although people would surely do a better job of examining the entire repository and picking out the most interesting nuggets, doing so would also fall into the realm of tedium.

Conversely, NuggetMine does not attempt to contribute nuggets to the repository because NuggetMine would find it very difficult to determine what nuggets group members would be interested in. This difficulty takes two forms. The first is that almost all machine learning and information filtering algorithms use past interests to predict future interests. Unfortunately, these algorithms do not handle novel information well, and we believe that novel information is a large part of the opportunistic, informal communication that NuggetMine supports. The second difficulty is that of avoiding the obvious. If a user submits a nugget that points to the official Moxy Früvous website, he probably does not want NuggetMine to pull in 800 other fan sites. We fear that by using the information submitted in the past, NuggetMine would select many commonplace, boring nuggets related to those already in the repository.

NuggetMine also demonstrates the utility of incorporating intelligence into a groupware application. With a few exceptions in the areas of automatic meeting scheduling [6] and information filtering [25], groupware applications include almost no intelligent behavior. Requiring users to perform all application tasks places a heavy burden on users. Including some intelligence in groupware application interfaces could ease that burden considerably and improve the utility of the groupware.

Organizing submitted nuggets is the most difficult and tedious aspect of sharing them; by performing this task itself, NuggetMine significantly reduces its users' burden. Keeping its users' burden minimal is especially important for NuggetMine because it is a non-essential, opportunistic application that relies on user utilization, interest, and contributions to succeed. Hence, any excessive burden to users will result in users not utilizing NuggetMine to their fullest benefit. NuggetMine's success in the pilot study suggests that we have effectively employed intelligent behavior to minimize the application's burden on the user and improve the user experience.

FUTURE WORK

We hope that NuggetMine inspires others in the IUI community to incorporate intelligence into new and existing groupware applications in efforts to improve the utility of groupware. NuggetMine itself can be extended in several directions. An obvious step would be to deploy it to more and larger groups with varied characteristics. Observing how different groups interact with NuggetMine would provide insight into when, where, and why it is most effective. Such observation would also allow us to investigate patterns of use among group members. What is the critical mass? Are there few members who contribute most of the nuggets? Does it change the use of other tools (e.g. e-mail, IM) in the group?

We would also like to make NuggetMine more intelligent. It should use smarter algorithms when selecting nuggets for a user. Despite the danger described earlier, developing techniques that allow NuggetMine to contribute to the repository could also be profitable. Perhaps NuggetMine could search for nuggets related to those in the repository and ask active users to submit these nuggets on its behalf. A social/persuasive component [8] that encourages users to share more nuggets would also be interesting. Including social visualizations in NuggetMine (e.g. who is contributing, how the repository is growing over time) to stimulate participation would be another interesting project.

Finally, we would like to address a particular weakness of NuggetMine: it is currently bound to the desktop. Many nuggets occur in conversation and we should capture them—but conversations often take place away from a computer. Ubiquitous computing technology [1] could capture and display nuggets beyond the desktop. Picture a PDA with a voice recorder that continuously monitors the conversation and records the last *n* seconds of conversation on command. An assistant could transcribe these snippets and submit them to NuggetMine. The PaperPDA system [10] could easily assist in the capture of nuggets, while peripheral and ambient displays [11] placed in a group common area could assist in their dissemination. Group members in these common areas—e.g. a lounge, a snack or water cooler area—are unlikely to be engaged in a task, and we hypothesize that these areas are excellent locations for opportunistically displaying nuggets.

CONCLUSIONS

NuggetMine is an intelligent groupware application that facilitates opportunistic sharing of information nuggets among a workgroup. NuggetMine collaborates with the

workgroup, dividing tasks so that each collaborator performs the tasks he/it does best. Group members submit nuggets to NuggetMine, and NuggetMine manages submission, organization, and display of the submitted nuggets via an opportunistic desktop interface. NuggetMine performed relatively successfully in a limited pilot study.

ACKNOWLEDGEMENTS

We thank the other four users who participated in the pilot study. Thanks also go to the participants of the University of Minnesota Spring 2001 HCI seminar for valuable discussion of issues related to this work. Finally, we thank the anonymous reviewers of this paper for their comments.

REFERENCES

[1] Abowd G., Mynatt, E. (2000). Charting the Past, Present, and Future: Research in Ubiquitous Computing. *ACM Transactions of Computer-Human Interaction, Vol. 7, Issue 1* (March 2000), pp. 29-58.

[2] Allen, J., Ferguson, G., Stent. A. (2001). An Architecture for more Realistic Conversational Systems. In *Proceedings of IUI 2001*, pp. 1-8.

[3] Bailey, B., Konstan, J., Carlis, J. (2000). Adjusting Windows: Balancing Information Awareness with Intrusion. In *Proceedings of the 6th Conference on Human Factors and the Web*, 2000.

[4] Chung, Y., Pottenger, W., Schatz, B. (1998). Automatic Subject Indexing using an Associative Neural Network. In *Proceedings of 3rd ACM Conference on Digital Libraries*, pp. 59-68.

[5] Davis, R., Landay, J., Chen, V., Huang, J., Lee, R., Li, F., Lin, J., Morrey III, C., Schleimer, B., Price, M., Schilit, B. (1999). NotePals: Lightweight Note Sharing by the Group, for the Group. In *Proceedings of CHI '99*, pp. 338-345.

[6] Erlich, S. (1987). Social and Psychological Factors Influencing the Design of Office Communication Systems. In *Proceedings of CHI '87*, pp. 323-329.

[7] Ferguson, G., Allen, J. (1998). TRIPS: An Integrated Intelligent Problem Solving Assistant. *Proc. Fifteenth National Conference on AI (AAAI-98)*, pp. 567-573.

[8] Fogg, B. (1998). Persuasive Computers: Perspectives and Research Directions. *Proc. CHI '98*, pp. 225-232.

[9] Giles, C., Bollacker, K., Lawrence, S. (1998). CiteSeer: an Automatic Citation Indexing System. In *Proceedings of 3rd ACM Conference on Digital Libraries*, pp. 89-98.

[10] Heiner, J., Hudson S., Tanaka, K. (1999). Linking and Messaging from Real Paper in the Paper PDA. In *Proceedings of UIST '99*, pp. 179-186.

[11] Ishii, H., Ullmer, B. (1997). Tangible Bits: towards seamless interfaces between people, bits, and atoms. In *Proceedings of CHI '97*, pp. 234-241.

[12] Joachims, T., Freitag, D., Mitchell, T. (1997). WebWatcher: A Tour Guide for the World Wide Web. *Proceedings of IJCAI97*, Nagoya, Japan, August 1997.

[13] Jones, S., Paynter, G. (1999). Topic-based Browsing within a Digital Library using Keyphrases. In *Proc. 4th ACM Conference on Digital Libraries*, pp. 114-121.

[14] Kraut, R., Fish, R., Root, B., Chalfonte, B. (1993). Informal Communication in Organizations. In *Groupware and Computer Supported Cooperative Work*, R. Baecker, Ed. M. Kaufman, San Mateo, CA.

[15] Lieberman, H. (1995). Letizia, An Agent for Web Browsing. *International Joint Conference on Artificial Intelligence,* Montréal, August 1995.

[16] Lieberman, H., van Dyke, N., Vivacqua, A. (1999). Let's browse: a collaborative Web browsing agent. In *Proceedings of IUI 1999,* pp. 65-68.

[17] Maltz, D., Ehrlich, K. (1995). Pointing the Way: Active Collaborative Filtering. In *Proceedings of CHI '95,* pp. 202-209.

[18] Nardi, B. A., Whittaker, S., Bradner, E. (2000). Interaction and outeraction: instant messaging in action. In *Proceedings of CSCW 2000*, pp. 79-88.

[19] Rhodes, B. (2000). Margin Notes. In *Proceedings of IUI 2000*, pp. 219-224.

[20] Resnik, P., and Varian, H.R. (1997). Introduction to the special section on recommender systems. *Communications of the ACM* 40(3), pp. 56-59.

[21] Rich, C., Sidner, C., Lesh, N. (2001). COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction. To appear in *AI Magazine, Special Issue on Intelligent User Interfaces, 2001*.

[22] Salton, G. (1991). Developments in Automatic Text Retrieval. *Science, Vol 253*, pp. 974-97.

[23] Sternberg, R. (1999). *Cognitive Psychology*. Harcourt Brace & Company, Orlando FL USA, 1999.

[24] Tang, J., Issacs, E., Rua, M. (1994). Supporting Distributed Groups with a Montage of Lightweight Interactions. In *Proceedings of CSCW '94*, pp. 23-34.

[25] Vivacqua, A., Lieberman, H. (2000). Agents to Assist in Finding Help. *Proceedings of CHI 2000*, pp. 65-72.

[26] Whittaker S., Frohlich D., Daly-Jones O. (1994). Informal workplace communication: What is it like and how might we support it? In *Proceedings of CHI '94*, pp. 130-137.

[27] Whittaker S., Sidner C. (1996). Email overload: Exploring personal information management of email. In *Proceedings of CHI '96*, pp. 276-283.

[28] Whittaker, S., Swanson, J., Kucan, J., Sidner, C. (1997). TeleNotes: managing lightweight interactions in the desktop. *ACM Transactions on Computer-Human Interaction, Vol. 4, Issue 2* (Jun. 1997), pp. 137-168.