Answer Garden 2: Merging Organizational Memory with Collaborative Help

Mark S. Ackerman
David W. McDonald
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717
{ackerman, dmcdonal}@ics.uci.edu
http://www.ics.uci.edu/CORPS/ackerman.html

ABSTRACT
This research examines a collaborative solution to a common problem, that of providing help to distributed users. The Answer Garden 2 system provides a second-generation architecture for organizational and community memory applications. After describing the need for Answer Garden 2's functionality, we describe the architecture of the system and two underlying systems, the Cafe ConstructionKit and Collaborative Refinery. We also present detailed descriptions of the collaborative help and collaborative refining facilities in the Answer Garden 2 system.

KEYWORDS: computer-supported cooperative work, organizational memory, community memory, corporate memory, group memory, information refining, information retrieval, information access, information systems, CMC, computer-mediated communications, help, collaborative help, CSCW

INTRODUCTION
Many user communities have a problem with delivering help and general assistance. Unfortunately, the user is often left to sift through reams of documentation, find his way through mail archives, or pursue answers through trial and error. Normally, one attempts to examine the documentation or other help sources, and then wanders out into a hallway in search of friendly colleagues.

The problem becomes acute, however, in distributed communities. We take for our example the astrophysics community, although this problem exists in most scientific communities. In the astrophysics community, the users may be spread across the world, they may work in isolation, and they may have need of relatively specialized help. What we would like is a surrogate for this hallway talk. Such a solution must avoid the broadcast problem of flooding everyone's electronic mail basket with thousands of questions. Instead, this work reports on a system to narrow-cast a question to the appropriate others, whether those others are experts or colleagues.

This research, then, examines a collaborative solution to a common problem. Earlier work, a system called Answer Garden, allowed organizations to develop databases of commonly asked questions that grow "organically" as new questions arise and are answered. The subsequent Answer Garden 2, the focus of this paper, continues this work. It is a second-generation architecture for the same design problem, investigating some of the issues encountered in field studies of the original system. The new architecture provides a customizable and adaptable set of software components that allow a variety of organizational and informational configurations. Furthermore, it offers a generalized solution to the problem of finding help for any information system. We report here on the new architecture and its responses to the context and authoring issues.

The paper begins with a brief introduction to the help and memory problems, as well as a brief overview of the original Answer Garden application and its field study results. Answer Garden 2 is then introduced. After an explanation of its architecture, the paper analyzes two particular features of Answer Garden 2. These two features, collaborative help and collaborative refining, are explained at length. Collaborative help mechanisms provide the necessary context for information, and collaborative refining mechanisms provide support for authoring. The paper concludes with a survey of related CSCW systems and some conclusions about these design considerations.

FRED'S PROBLEM
Fred (not his real name) is an astrophysicist at the Harvard-Smithsonian Astrophysical Observatory. He, like many scientists, does not want to know anything about software systems or his hardware. He wants to do his scientific work, free of the multitude of computer problems that seem to get in the way.

In the "old days," everyone sat around in a common room, using their computer consoles with the mini-computer. If Fred had a question, he could ask one of the half-dozen to dozen colleagues and programmers sitting in the room.

Computer Supported Cooperative Work '96 , Cambridge MA USA
© 1996 ACM 0-89791-765-0/96/11 ..$3.50

Everyone had to hear the answer, so the community learnt from the problems of each individual.

Now, Fred sits in his office with his workstation near his desk. It is quieter, but much more isolated. If he has a problem or a question, he can look through the documentation or send electronic mail for help. If he sends electronic mail, he may not get an answer from the programmers for some unknown period of time, or he may be given a response that makes him feel stupid for not knowing the answer. Often he resorts to wandering through the hallways, looking for people who might know the answer. He then tries various possibilities until he finds a solution or he gives up.

Any community, institution, or organization of any size often has a problem with answering questions in a timely manner. Yet, solving problems and completing tasks are often dependent on obtaining timely answers to specific questions.

Fred's problem is the dual problem of help and of collective memory. We will use the term *collective memory* to denote the common attributes of organizational, institutional, and community memory. (The term has a related, but slightly different meaning in the historiographical and critical literatures, but there is no better term to denote memory in a range of collectivities.)

Within an organization or community, individuals' information seeking requires finding the right part of the collective memory. Typically, collective memories include information repositories (e.g., information databases, filing cabinets, documents). It can also include people (e.g., other organizational personnel) [25]. The collective memory to which Fred has access includes at least the documentation, the system programmers, and his colleagues. However, he may have great trouble finding the right piece of the collective memory that has the answer he needs. In other words, his access to the collective memory should be augmented.

Answer Garden and Fred's problem

Previous work, reported in [4] and [2], considered one way of doing this augmentation. This work revolved around a system called Answer Garden. Field studies of its use uncovered a number of important problems in providing collective memory and help to users such as Fred. Before discussing these problems, and our subsequent investigations, it will be useful to briefly describe Answer Garden. This application still plays an important part in our current work.

Answer Garden supports organizational memory in two ways: by making recorded knowledge retrievable and by making individuals with knowledge accessible. In the standard configuration of Answer Garden, users seek answers to commonly asked questions through a set of diagnostic questions or other information retrieval mechanisms. Figures 1 and 2 show Answer Garden reimplemented in the World Wide Web. (Other, third-party versions exist in the Web [22] and in Lotus Notes.) Diagnostic questions guide the user through Web pages. Alternatively, the user may use a number of other information retrieval mechanisms to find the pages that may contain the answer.

If the user cannot find an answer or the answer is incomplete, the user may ask the question through the system. (This is the result of the user pressing the "I'm Unhappy" link in Figure 1.) In the original Answer Garden, the system would then route the question to an appropriate human expert. (This has been changed in Answer Garden 2 as will be discussed below.)

In the original Answer Garden, the expert would then answer the user through electronic mail. If the question was a common one, the expert could insert the question and its answer back into the information database. Thus, users were not limited to the information in the system; if the information was not present, they could tap the organization's experts. As a result, the organization would gain a corpus of information, an organizational memory. Users could obtain expert advice without a high organizational cost. Other interesting properties of the system are discussed in [2].

Open research issues

Field studies of Answer Garden's use ([2], [3]) uncovered a number of issues. While the system was held to have worked, two issues were uncovered that are critical to the success of similar memory or help systems:

- Tying the social network into the system in a more natural manner. Answer Garden's dichotomy between experts and users was problematic. While there was nothing in the underlying technology to force this dichotomy, it was a simplifying assumption in the field study to have separate user and expert groups. Real collectivities do not function this way. Most people range in their expertise among many different skills and fields of knowledge. Fred knows things about systems and his tasks, even though he may not be able to answer specific questions. We would like to allow everyone to contribute as they can, promoting both individual and collective learning.

  However, mechanisms to allow each person to contribute must not overwhelm the other people who use the system. For example, broadcasting each question to every person in an organization or community will fail. AG2 offers several mechanisms to ameliorate the overload problem while allowing and providing for a range of expertise.

- Providing for the contextualization of answers, thus providing for the user's understanding of an answer. In the Answer Garden field study, most users either did not need contextualized information or were able to contextualize it themselves. However, a significant portion of the participants did need more context.

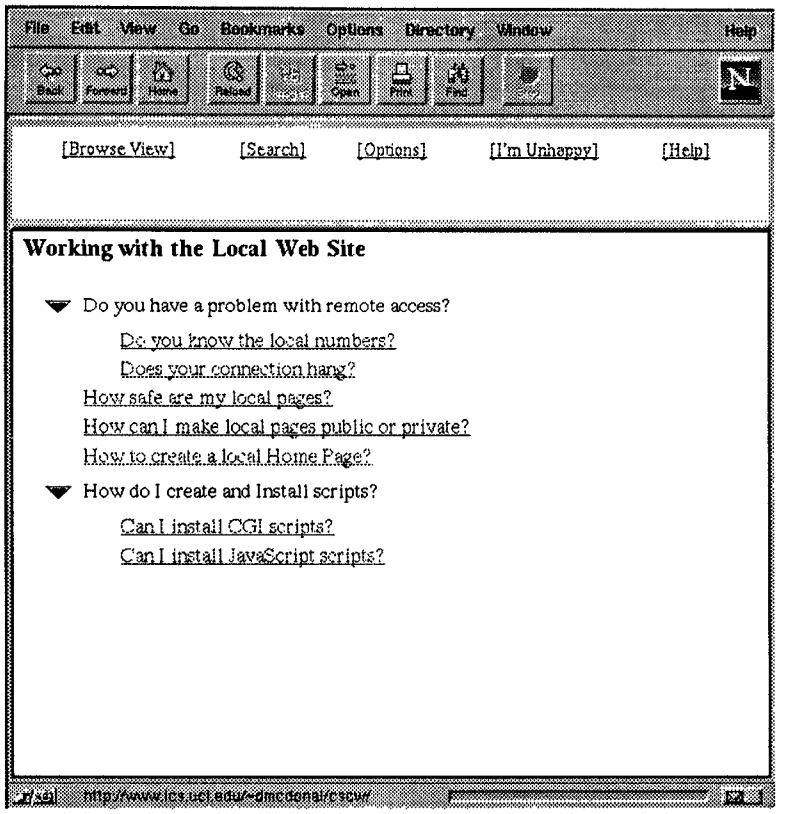


Figure 1: Answer Garden functionality in the Web

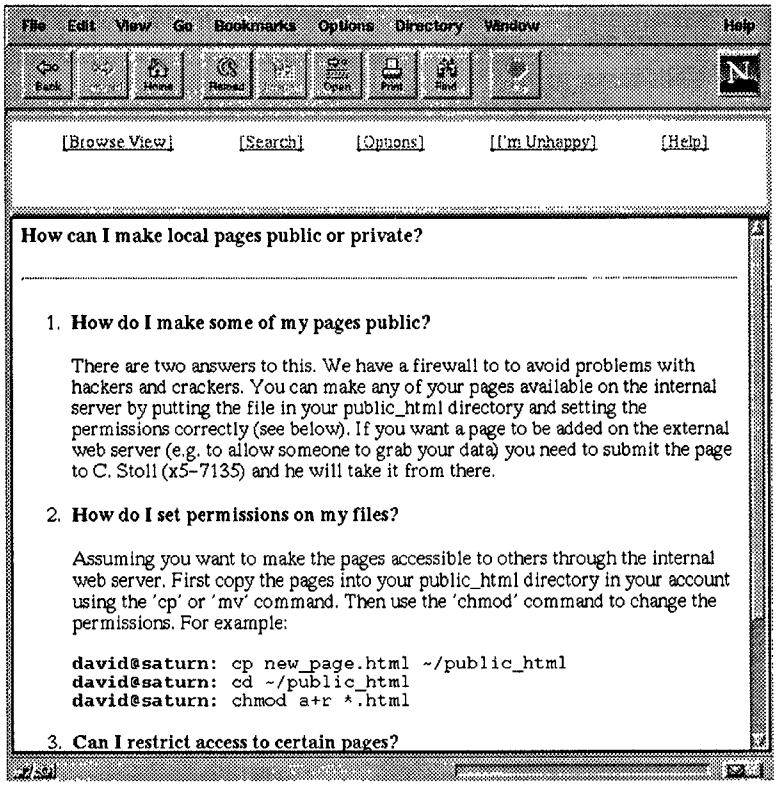


Figure 2: An Answer Garden answer page

In Fred's case, the answer to a question may be present in the documentation. However, he may lack the required expertise to infer an answer or to even use an explicit answer without additional situational information.

Providing the proper context is, unfortunately, difficult. We will return below to one way of potentially providing this context at low cost. Our mechanism also ameliorates the problem of providing answers at the right level and length of explanation.

- Easing the authoring burden. To obtain answers, the cost of authoring must be minimized. Furthermore, authoring answers, as an individual activity, promulgates the distinction between experts and everyone else. The composing content of answers takes as long as any writing takes, but we may be able to ease the mechanics of the process.

One might expect these issues to become increasingly problematic as the information becomes non-technical or the users become less sophisticated in the domain. For example, only astrophysicists can understand the scientific analysis tasks that create their questions about software systems. Astrophysicists will vary in their computer expertise, but few wish to spend time inferring the answer from substantial system documentation before continuing with their analysis tasks. And, the programmers who must currently compose the answers may not even understand the domain or its tasks.

Additionally, the field studies uncovered a number of technical issues, such as the need to use varying "front-end" systems such as the Web or Notes, to consider additional methods of finding experts, and to find better ways of maintaining the information database. These technical issues and the above social issues led us to reconsider the architectural design.

ANSWER GARDEN 2 (AG2)

Answer Garden 2 (AG2) consists of a second generation system architecture for organizational memory and collaborative help support. There are several advantages to this architecture.

First, the design cleanly separates the front-end of Answer Garden (i.e., the user client) from back-end needs. More importantly, it also decomposes the Answer Garden functionality into a set of distributed software services. This provides a high level of organizational flexibility; the services can be mixed and matched in order to provide additional flexibility. For example, by attaching an anonymity service, users of the system can send their questions anonymously. By attaching an anonymity service at another point in the distributed architecture, the experts answering the questions can also be anonymous. Or by not having an anonymity service at all, all users and experts can be known to one another.

Finally, the change in architecture makes much of the help functionality *possible from any information system.* This work, then, is generalizable to any information system.

System components

AG2 is built upon two underlying systems, both of which provide a set of services. These services create the collaborative help and collective memory functionality. The two underlying systems are:

- The Cafe ConstructionKit (CafeCK). CafeCK is a CSCW toolkit for supporting sociality and information use in collaborative environments [6]. CafeCK provides a set of reusable objects that include message transport for asynchronous and synchronous communication (including a Zephyr-like system, NetNews, and email), parsing for a variety of semi-structured protocols, private and public channels for narrowcast communication, message filters, and message retrieval by a variety of semi-structured methods. By selecting from the set of available components (or by extending it) and by writing a simple Tcl program, an application writer can create a set of distributed processes to handle information retrieval, information access, or electronic communications. CafeCK is implemented in C++, Tcl, and Tk.
- Collaborative Refinery (Co-Refinery). Co-Refinery provides mechanisms for handling individual and joint information spaces. Central to Co-Refinery is the ability to individually and collaboratively view and manipulate Answer Gardens and other information

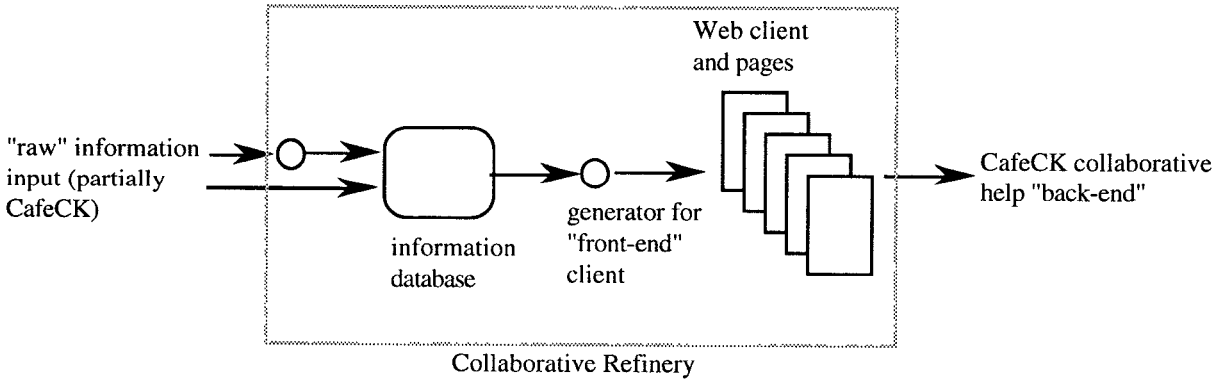
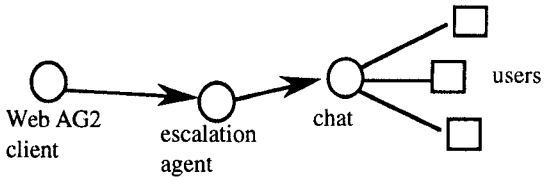
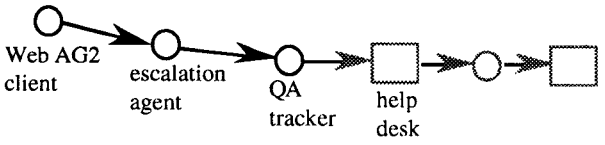


Figure 3: Answer Garden 2 (AG2) architecture



(a) The user's first attempt to get an answer goes to a chat channel.



(b) The user's *j*th attempt to get an answer gets escalated to a help desk.

Figure 4: Two possible escalations for a question

collections. It is especially useful in situations where one wants to refine and distill collections of materials as shared artifacts. It will be described extensively below.

Co-Refinery components include objects for managing a collection archive of materials, constructing and maintaining a database of relationships for those materials, and generating a suitable presentation. Output from Co-Refinery's presentation generator can be HTML, Notes documents, files, or e-mail. Co-Refinery is implemented in C++, and the Web portion relies upon HTML 3.0 and Netscape HTML extensions.

These two components are used together as in Figure 3. Raw information comes into the collection archive through CafeCK processes (such as News filters), by being explicitly sent to the archive through e-mail, or through filtering agents. It may be partially processed, and then is moved into the information database. At snap-shots or upon explicit queries (depending on a site's tailoring of AG2), the materials are built into Web pages, Notes documents, or flat files. In turn, the AG2 Web or Notes clients can send mail to CafeCK back-end processes that then handle the details of obtaining help. These CafeCK help processes will be described next.

COLLABORATIVE HELP

The problem as a duality

AG2's "back end" can be viewed either as a collective memory system or as a collaborative help system. (We use *collaborative help* to denote those help systems that use people as information sources, for example, through Computer-Mediated Communication systems.) Each of these views is the dual of the other. By duals, we invoke the language of linear programming, where two forms exist for each particular problem. Both forms are valid, and users are free to solve the form that provides them with the most analytical tractability. By considering the "back-end" organizational memory problem in terms of its dual, collaborative help, we believe we have found mechanisms for reducing the context problem.

Above, it was noted that an open research issue was how to alleviate the users' need for contextualized information in solving their problems and finishing their tasks. This issue can be ameliorated by using collaborative help in a controlled manner. Collaborative help functionality also provides help to users at their own explanation level and potentially with iterative diagnosis.

Staying local

Providing help from other people -- such as colleagues on the same hall or other group members -- allows people to seek help first from the people most likely to know the local context. Colleagues can judge a person's abilities, expertise, and situation, and can try to provide suitable information to solve the person's problem. Local participants are also more likely to information, since personal social ties are key motivators in providing assistance [7, 19].

Always asking one's colleagues is, however, problematic. First, it is still costly to ask other people. AG2's repository of previously-asked questions and frequently-required information, however, attempts to reduce that problem. More importantly, one's colleagues may not know the answer. While staying local is important, it can also be organizationally dysfunctional [10] when there is no local expert available. In these situations, a means for escalating answers past the local group is required.

Escalation

Using the facilities of CafeCK, we were able to simply construct an escalation agent for questions in AG2. This component allows the user to decide what to do if the question is not answered. It allows the user to consider whether to get answers from chat systems, bulletin boards, software agents, or other people.

The typical way that we envision the system being used is to gracefully escalate the help request until it can be answered. Because the escalation agent is a CafeCK process, the escalation can be quite flexible. The agent is currently programmed to follow organizational rules on the order of escalation, although this is under user control. It would be a simple matter to change this to provide different organizational rules, complete user control, or even heuristics (such as avoiding the chat facility when no other users are logged into their machines). No doubt other mechanisms could be found; this is a potential research question.

In our prototype, the user poses a question through his application. In the example of Fred, the user client is an AG2 front-end, but it can be any application that has asynchronous or synchronous communication capabilities. The application merely connects to a CafeCK process through, for example, e-mail. This CafeCK process, the escalation agent, is semi-autonomous, since it can be triggered either by the user or automatically.

As an example, imagine the following scenario. (Note that the underlying components for AG2 provide an enormous flexibility, so users' actual practices can vary widely from this.) The user, Fred again, has a question about his data analysis package, and he would like to know how to correctly massage his data. He first looks through the existing questions and answers, either in a stand-alone AG2 information database or in an AG2 component of his data analysis application. Assuming that the answer is not there, or that he does not understand how to apply the information that is there, he composes a question and mails it off through his Web browser.

Instead of the question going to an expert, as it would have in the original Answer Garden, the question goes to his escalation agent. This is, of course, invisible to Fred. The question is first sent to a synchronous chat system (Figure 4a). We envision the chat system being set up with channels or subchannels for each work group, hallway, or other social grouping. If someone on the chat system can answer Fred's question, and is inclined to do so, Fred gets his answer immediately. As mentioned above, nearby colleagues (as measured by geographical, social, or intellectual distance) are most likely to answer his question with the correct and sufficient context.

In our prototype, after 5 minutes, the system pops up a window on the screen. In our scenario, the dialog box asks Fred whether he got an answer to his question, and if not, whether he would like to continue (Figure 5). If Fred says to continue, the system routes the question to a NetNews bulletin board. (It is also conceivable that it would route it to a chat channel with a wider distribution, but the point is the same.) After another period of time, perhaps 24 hours, the agent again pops up a window on Fred's screen, asking whether he has received an answer. The process continues, perhaps routing the question to an expertise engine to find a suitable human expert, to a help desk (Figure 4b), or to agents that search for information on the Web or in proprietary information sources (such as Dialog or Nexus). One can even imagine agents that hire outside consultants if the need is great enough.

In this manner, Fred or any other user is more assured of receiving a usable answer. Staying local lowers the cost, since organizational-level experts need not be used immediately; increases the chance of getting an answer, since colleagues may be more motivated to answer; and is more likely to provide context, since colleagues know the local situation. To be sure, this approach is not a panacea. While it does help provide the proper amount of contextual

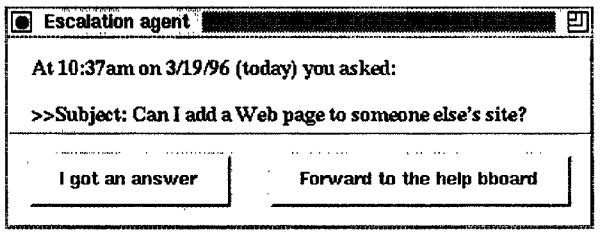


Figure 5: The escalation agent

information to make the answer meaningful, and while there is a greater likelihood that the answer will be at the right explanation level and length, there are difficult issues surrounding the social organization of channel groupings and the like. Colleagues' time is hardly free.

Nonetheless, staying local (but thinking global) does allow group members to help one another, while preserving the capability to ask larger groups as well as experts. Furthermore, the dichotomy between experts and users is largely broken down.

Other collaborative help services

AG2 requires a number of other CafeCK services. In addition to the basic communication services (chat, NetNews, e-mail) and to the escalation agent, AG2 requires services to find experts, to provide basic statistical services, to make users anonymous, and to track users' questions. The capability to find a suitable expert is required, and AG2 currently uses a rudimentary rule-based finding mechanism. This is clearly a bottle-neck for real use, but other researchers are developing better mechanisms for handling this problem (e.g., [17]). The anonymity service allows users to ask questions anonymously. Organizations or communities might not want this service, in which case the service is merely omitted. The statistics service notes which communication mechanisms are used, and also tracks the use of pre-existing answers. In a production system, users' questions should be tracked; otherwise, questions can slip away.

The design of CafeCK allows these components to be mixed and matched in a building block manner. Different organizational arrangements can be created through the architecture of the software components. Furthermore, each service can be tailored through its internal Tcl programs.

Currently, only a simple expertise engine and anonymity service have been implemented. The others are planned.

In summary, viewing the problem as one of collaborative help allows one to remove the requirement that organizational memory merely be a set of information repositories. Staying local, with the possibility of escalation, provides for a range of help from the people around the information seeker. However, AG2 also includes stronger support for building information repositories as well. This support will be described next.

REFINING A MEMORY REPOSITORY

> On my shelves were tons of unwinnowed material.... In the present shape it was of little use to me or to the world. Facts were too scattered; indeed, mingled and hidden as they were in huge masses of debris, the more one had of them the worse. ...To find a way to the gold of this amalgam...was the first thing to be done. (Bancroft [8], 1891, p. 135)

Central to the Answer Garden application is iteratively building a repository of commonly requested answers and other information. If this is to be accomplished, low

overhead is required for organizational or community members.

The original Answer Garden design assumed that building such a memory repository would occur through the everyday interaction of users asking questions and experts providing answers. However, authoring was still a significant task. The effort of writing explanations and formulating answers cannot be minimized. Nonetheless, Answer Garden provides mechanisms for iteratively developing an answer. AG2 provides additional mechanisms for refining answers from very raw information sources as well as removing unnecessary context. We developed Co-Refinery to provide these mechanisms. The goal is to enable groups of people to collaborate in jointly or individually building answers and information repositories over time.

Collaborative refining

AG2, through the underlying Co-Refinery system, supports an authoring process that includes four general activities: collecting, culling, organizing, and distilling. We assume that any of these activities, as well as authoring, may be done iteratively or in any order. Each activity is clearly important, although the major research contribution here is the support for collaborative distilling:

- *Collecting* is the phase in which information is gathered. In Co-Refinery, automatic collecting can be set up for certain types of information streams that occur in AG2, such as NetNews, synchronous chat channels, or distribution lists. In addition, manual collecting allows individual items to be submitted through the system directly or by e-mail. Collecting places items into the archive.
- After collecting the material, one must cull the collection for interesting material, and the lesser material must be discarded or ignored. *Culling* is a selection mechanism, identifying themes or threads that occur within a collection. A sizable reduction of material may be possible through culling the collection, making subsequent organizing and distilling easier. Culling reduces the apparent size of the archive, although in our current implementation, items are unreferenced rather than deleted.
- *Organizing* allows one to group materials according to some classification schemes so to enhance their retrievability and understandability. Our current prototype relies heavily on outlining, user-defined indexing, and keyword indexing, but other classification mechanisms are clearly possible. In Co-Refinery, retrievability is enhanced by making the culled subset a fully identifiable element in the collection. In this way, organizing results in an addition to the collection.
- The most important part of refining is *distilling* -- boiling down the existing (and culled) materials in order to uncover the answers or knowledge. As with chemical or liquor distilling, the results should be a more concentrated or concise form of the original information. Creating or editing a summary or synopsis, for example, removes much of the tedious work of wading through extraneous or erroneous information.

  The result of distilling is a *distillate*. Support is provided for generating the raw material for authoring a distillate, but it assumed that only users can fully distill and refine the material. Co-Refinery currently supports a number of distillates. For example, a useful intermediate distillate consists of merely concatenating selected NetNews messages. This allows an author/editor to further prune the selected information into one final distillate consisting of an authoritative answer. This behavior is very similar to what people currently do when they compile a FAQ. Another useful distillate is temporally based; items in it vanish after a short period of time. Technical hotlines often have runs of questions, and this distillate solves the problem of communicating the temporarily needed answers.

As mentioned, we assume that users move fluidly among these activities. We also assume that the refining is done iteratively and incrementally. In doing so, the system allows groups of users to interact in creating shared information artifacts and a common information space. This system represents an alternative to many current attempts to completely automate the refining process.

Figure 6 shows the results of refining in AG2, and can be considered as a snapshot of an iterative process. In Figure 6, the leaves are raw information, perhaps NetNews or e-mail messages. Authors and editors have created distillates for five of the threads, winnowing the material into answers for frequently-asked questions. One of these answers was shown in Figure 2 above. After being finished, some distillates can then take the place of the raw material in the archive. Note that some of these distillates could be intermediate; i.e., not shown to the public because they are under construction. Additionally, all distillates can be iteratively revised.

The economics of distilling

Refining is not a complete solution to authoring. There will always be effort required to compose explanations of complex technologies and tasks. Refining reduces the overhead for that task, and simultaneously reduces information overload.

We do not believe that all materials will be refined. It does not make sense in all cases to move data through information to organizational knowledge. For example, information that has a short-shelf life will not be refined, especially if the information also has a high throughput velocity. The cost would be prohibitive. Therefore, we have attempted to define some distillates that are suitable for temporally limited information. These distillates do not boil down the material, but they do make finding and retrieving the material easier.

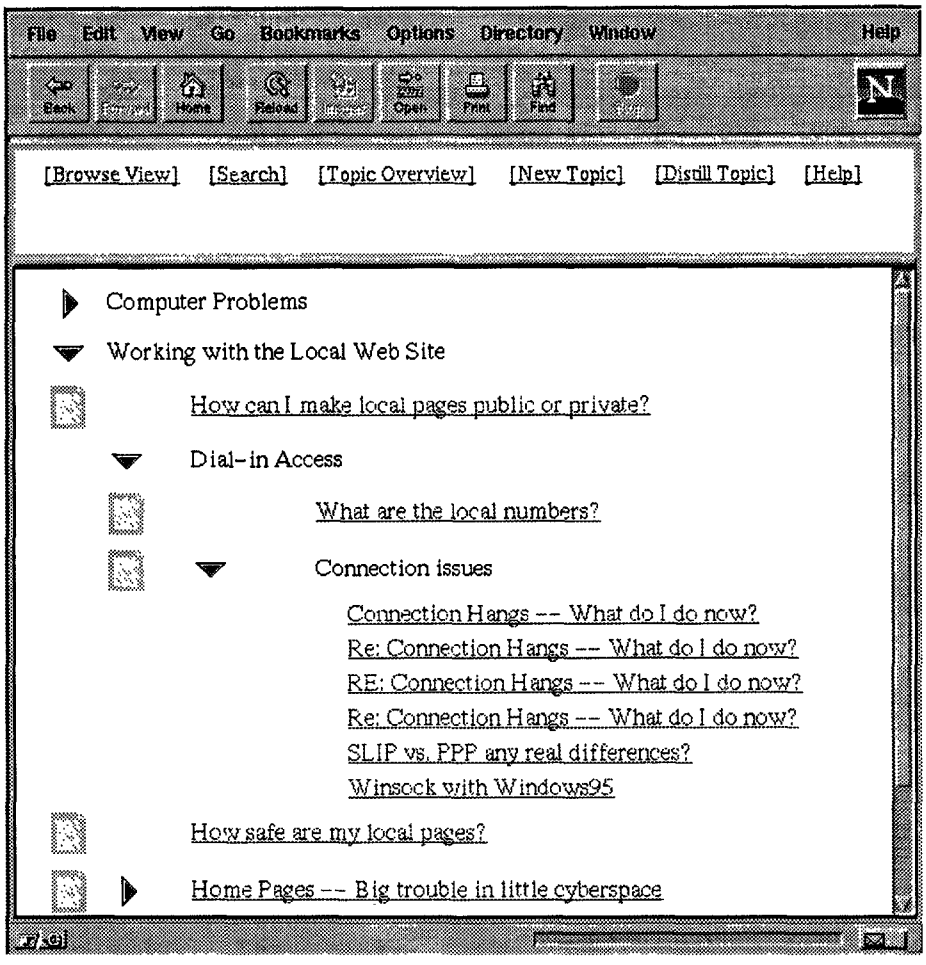


Figure 6: Co-Refinery screen with raw information nodes and distillates

In summary, Co-Refinery and AG2 provide new mechanisms for authoring and editing large amounts of information. By using the Co-Refinery collecting, culling, organizing, and distilling facilities, people can find new ways of providing answers. If people do so collaboratively, AG2 can further reduce the separation between users and experts, allowing for additional organizational learning.

RELATED SYSTEMS

AG2 is clearly related to a number of HCI systems. Help systems in general have been extensively studied in the HCI literature (e.g., [12], [18], [1]). However, while collaborative help is a widely used organizational and social mechanism, it has not been properly considered within the literature. (But see [23], [15], and [21] for notable exceptions to this.) Sproull and Kiesler [23] demonstrate the utility of asynchronous communication mechanisms for providing help within organizations. Our previous work [5] indicated the utility of synchronous chat systems, like Zephyr [14], for help. To our knowledge, no other system has escalation agents or the same range of help services.

There are a number of related CSCW systems that attempt to deal with collaborative information handling. Closest to AG2 are Grassroots [16], Community Memory [13], Spider [11], Designer Assistant (Living Design) [24], Group Memories [20], and BSCW [9]. Grassroots has different mechanisms for collecting, culling, and organizing information. These mechanisms are complimentary to those in AG2. However, Grassroots is lacking distilling features, since sharing the original documents is the major thrust of the work. Community Memory, like AG2, attempts to build a collaborative information space, but it also lacks explicit support for refining answers or obtaining help. Spider argues for collaborative sense-making and discussion. We have adopted its general philosophy, but its major technical thrust is not on the help problem. Designer Assistant, similarly, provides for iterative information gathering and sense-making, but its major thrust is on design rationale. BSCW provides an alternative shared workspace environment, but it lacks support for the refining activities in AG2. None of these systems, as far as we know, have collaborative help facilities.

SUMMARY

Returning to poor Fred's problem, all he wants is help to solve his computer questions. The institution in which he has an office and the community to which he belongs, however, want to reduce the cost and effort of answering those questions. Fred's problem, then, becomes one of

augmenting the collective memory in such a way that it benefits Fred as well as all of the social collectivities of which he is a part.

This paper has examined some technical mechanisms for augmenting the collective memory accordingly. We have tried to show that collaborative help and collaborative refining techniques can be of benefit. These techniques can ameliorate, respectively, the context and authoring issues. Moreover, we have especially tried in this work to correct the problematic dichotomy of experts and users, promulgated in our earlier work.

While we believe that these technical mechanisms offer great potential for organizational and community memory, proof must await field studies of AG2's use.

Acknowledgments

This project has been funded, in part, by grants from NASA (NRA-93-OSSA-09), the UCI Committee on Research, Interval Research Corporation, and the California Department of Transportation. We want to especially acknowledge Eric Mandel for his continuing insights into the needs of user communities, especially the astrophysics community. This project has benefited greatly from conversations with Tom Malone, Paul Resnick, Wanda Orlikowski, JoAnne Yates, Debby Hindus, Jonathan Grudin, Rob Kling, John King, Takeshi Ishizaki, Leysia Palen, Paul Dourish, Kate Ehrlich, Christine Halverson, and many others. The other members of our research group, Brian Starr, Jack Muramatsu, Wayne Lutters, and Andy Tipple, contributed to this understanding of collective memory and collaborative help.

References

1. Aaronson, A. and J. M. Carroll. Intelligent Help in a One-Shot Dialog: A Protocol Study. *Proceedings of CHI + GI '87*, 1987: 163-168.
2. Ackerman, M. S. Augmenting the Organizational Memory: A Field Study of Answer Garden. *Proceedings of CSCW'94*, 1994: 243-252.
3. Ackerman, M. S. Definitional and Contextual Issues in Organizational and Group Memories. *Information Technology and People*, 1996, 9(1): 10-24.
4. Ackerman, M. S. and T. W. Malone. Answer Garden: A Tool for Growing Organizational Memory. *Proceedings of ACM Conference on Office Information Systems*, 1990: 31-39.
5. Ackerman, M. S. and L. Palen. The Zephyr Help Instance: Promoting Ongoing Activity in a CSCW System. *Proceedings of CHI'96*, 1996: 268-275.
6. Ackerman, M. S. and B. Starr. Social Activity Indicators: Interface Components for CSCW Systems. *Proceedings of ACM Symposium on User Interface Software and Technology (UIST)*, 1995: 159-168.
7. Allen, T. *Managing the Flow of Technology*. MIT Press, Cambridge, MA, 1977.
8. Bancroft, H. H. *Literary Industries: A Memoir*. Harper and Brothers, New York, 1891.
9. Bentley, R., T. Horstmann, K. Sikkel and J. Trevor. Supporting Collaborative Information Sharing with the World Wide Web: The BSCW Shared Workspace System. *Proceedings of WWW Conference*, 1995: manuscript.
10. Blau, P. *The Dynamics of Bureaucracy: A Study of Interpersonal Relations in Two Government Agencies*. University of Chicago Press, Chicago, 1955.
11. Boland, R. J., Jr., R. V. Tenkasi and D. Te'eni. Designing Information Technology to Support Distributed Cognition. *Organization Science*, 1994, 5(3): 456-475.
12. Campagnoni, F. R. and K. Ehrlich. Information Retrieval Using a Hypertext-Based Help System. *Proceedings of SIGIR '89*, 1989: 212-220.
13. Chaplin, D. Community Memory. Department of Computer Science, Lancaster University, manuscript, 1994.
14. DellaFera, C. A., M. W. Eichin, R. S. French, D. C. Jedlinsky, J. T. Kohl and W. E. Sommerfeld. The Zephyr Notification Service. *Proceedings of Winter 1988 Usenix Technical Conference*, 1988: 213-220.
15. Finholt, T. A. Outsiders on the Inside: Sharing Information through a Computer Archive. Carnegie Mellon University, Ph.D. thesis, 1993.
16. Kamiya, K., M. Roscheisen and T. Winograd. Grassroots: Providing a Uniform Framework for Communicating, Sharing Information, and Organizing People. *Proceedings of CHI'96*, 1996: 239-240.
17. Kautz, H., A. Milewski and B. Selman. Agent Amplified Communication. *Proceedings of AAAI Symposium on Information Gathering from Heterogeneous, Distributed Environments*, 1995.
18. Kearsley, G. *Online Help Systems : Design and Implementation*. Ablex, Norwood, NJ, 1988.
19. Kraut, R. E., C. Egido and J. Galegher. Patterns of Contact and Communication in Scientific Research Collaboration. In Galegher, J., R. E. Kraut and C. Egido (ed). *Intellectual Teamwork*. Lawrence Erlbaum, Hillsdale, NJ, 1990.
20. Lindstaedt, S. N. Group Memories: A Knowledge Medium for Communities of Practice. Department of Computer Science, University of Colorado, Ph.D. proposal, 1996.
21. Okamura, K., W. J. Orlikowski, M. Fujimoto and J. Yates. Helping CSCW Applications Succeed: The Role of Mediators in the Context of Use. *Proceedings of CSCW'94*, 1994: 55-65.
22. Smeaton, C. The AnswerWeb. *Proceedings of WWW Conference*, 1995: http://www.cee.hw.ac.uk:80 /~calum/AnswerWeb /paper.html.
23. Sproull, L. and S. Kiesler. *Connections: New Ways of Working in the Networked Organization*. MIT Press, Cambridge, MA, 1991.
24. Terveen, L., P. Selfridge and M. D. Long. Living Design Memory: Framework, Implementation, Lessons Learned. *Human-Computer Interaction*, 1995, 10(1): 1-38.
25. Walsh, J. P. and G. R. Ungson. Organizational Memory. *The Academy of Management Review*, 1991, 16(1): 57-91.