International Journal of Artificial Intelligence in Education (1998), 9, 159-177.

Supporting Peer Help and Collaboration in Distributed Workplace Environments

J. Greer, G. McCalla, J. Collins, V. Kumar, P. Meagher, J. Vassileva, *Aries Laboratory, Department of Computer Science, University of Saskatchewan, Saskatoon, SK, S7N 5A9 Canada*

Abstract. Increasingly, organizations are geographically distributed with activities coordinated and integrated through the use of information technology. Such organizations face constant change and the corresponding need for continual learning and renewal of their workers. In this paper we describe a prototype system called PHelpS (Peer Help System) that facilitates workers in carrying out such "life long learning". PHelpS supports workers as they perform their tasks, offers assistance in finding peer helpers when required, and mediates communication on task-related topics. When a worker runs into difficulty in carrying out a task, PHelpS provides a list of other workers who are ready, willing and able to help him or her. The worker then selects a particular helper with PHelpS supporting the subsequent help interaction. The PHelpS system acts as a facilitator to stimulate learning and collaboration, rather than as a directive agent imposing its perspectives on the workers. In this way PHelpS facilitates the creation of extensive informal peer help networks, where workers help one another with tasks and opens up new research avenues for further exploration of AI-based computer-supported collaborative learning.

INTRODUCTION

A successful modern organization is characterized by workers who feel empowered and take ownership and pride in the things they do. It has a climate where employees at all levels help one another to meet the organization's mission. It fosters and rewards continual learning, encourages cross-training and collegial ties, and distributes the organizational knowledge across the entire workforce. But a successful modern organization doesn't just happen spontaneously. It develops as a result of organizational change where a management-created climate encourages the growth of grass-roots communities of learners and actively supports collaborative learning efforts.

One common component of collaborative learning is the "informal peer-help network". This is a social network of a worker's colleagues and acquaintances who are consulted for help and advice about work matters. Informal peer-help networks exist in every organization (Constant et al., 1996). They constitute a crucial component of training and acclimatization for new employees and represent an essential element of shared organizational memory. The goal of our project is to use technology to support collaborative learning and the expansion of informal peer-help networks in large distributed organizations, in particular those where information technology plays a central role.

In order to effectively use modern information technology a worker must possess both lower-level procedural and higher-level problem-solving and judgement skills. On the job performance support through a good on-line help system or task checklists can assist workers to overcome many simple impasses. Nevertheless, no matter the degree of training or skill, workers using information technology often need to request just-in-time help from someone in their informal peer-help network. More experienced users may make less frequent but more complex help requests, while novice users tend to make more frequent and normally less complex help requests.

The specific objective of our project is to provide assistance to workers who are learning to make effective use of a management information system in their workplace. This involves activities that range from training to performance support to just-in-time help. The approach we have taken is to develop a task-oriented performance support system that is tightly coupled with a peer help system (dubbed PHelpS). There are two novel aspects of the PHelpS approach. The first involves the use of a task-oriented performance support system to index help requests and user knowledge. The second involves knowledge-based support for locating a peer somewhere in the distributed organization who is ready, willing, and able to provide help when needed. Combining these aspects produces an effective environment for delivering just-in-time help. It also creates a fertile environment in which to study collaborative learning and organizational change. Finally, the necessity to maintain knowledge profiles of potential helpers has produced many interesting research challenges in the domain of user and learner modelling.

This paper describes the PHelpS approach, concentrating on its strengths in supporting collaborative learning through peer help and speculating on its potential to effect organizational change.

COLLABORATIVE LEARNING IN A WORKSPACE TRAINING CONTEXT

Peer help within organizations is a valuable activity for a number of reasons: peer help is a cost effective method of training; it supports the decentralization of the organizational knowledge base and thus preserves organizational memory; it builds a sense of collegiality within the organization; it reinforces the knowledge of both the helpers and the persons being helped (Chi et al., 1994, Pressley et al, 1992).

Constant et al. (Constant et al., 1996) characterize “help exchanges” within organizations. They indicate that people prefer to exchange help by way of collegial ties, which develop with physical proximity, group membership, prior relationships and demographic similarity. In contrast to the strong ties of personal friendships within an organization, they advocate the development of *weak ties* among wider collections of workers so that informal peer-help networks can be expanded. This notion is compatible with Wenger's communities of learners (Wenger, 1996), where people who share learning goals within an authentic learning environment can develop ties that reinforce learning outcomes.

Developing collegial ties within a distributed organization is a challenge, since co-workers may be strangers separated by space and time. These separations pose similar challenges to those impeding the development of collegial ties in distance learning environments. Computer support for nurturing such collegial ties can be helpful. Furthermore, the provision of peer-help and peer tutoring has the positive side effect of strengthening the knowledge and confidence of both the person being helped and the helper or tutor (Palthepu et al., 1991; Slavin, 1990; Nichols, 1993).

On the issue of providing help for workers, Carroll (1990) demonstrates the ineffectiveness of de-contextualized help and the effectiveness of task-oriented help in learning about computer and information systems. His claims are sympathetic with the emphasis placed on social context in the situated learning community. Interestingly, in workplace settings where tasks and duties are fairly well specified, it becomes relatively straightforward to define and represent task models that can be used to encourage situated learning. There is a definite connection between task models and the potential for developing collegial ties. Weak ties develop in the context of carrying out workplace tasks (Constant, 1996). Thus, understanding the nature of tasks has a dual benefit of better understanding and addressing help needs and better supporting the development of collegial ties. This support can lead to the self-reinforcing benefit of increased peer collaboration, more collaborative learning in the organization, and may eventually result in positive changes in the way workers cooperate with one another and how they think about and carry out their tasks (Hammer, 1996).

In the remainder of this paper we describe the PhelpS system, designed to encourage distributed peer help within organizations.

THE PEER HELP SYSTEM (PHelpS)

The PHelpS system (Collins et al., 1997; McCalla et al., 1997, Greer et al., 1997) has been developed, tested and deployed in the context of the Correctional Services of Canada (CSC) as part of a staff training initiative. The research is sponsored by the Canadian TeleLearning Network of Centres of Excellence program. CSC is a national distributed organization with some 11000 workers in 281 different locations. PHelpS prototypes have been tuned to suit the needs of workers in the Prairie Regional Psychiatric Centre (RPC), one of the facilities of CSC. A large CSC-wide information system called the Offender Management System (OMS) has recently been introduced within CSC and almost all workers are expected to make significant use of this system in their daily activities. PHelpS functions “on top of” OMS to provide performance support and to facilitate peer help with tasks related to OMS. Of course, the PHelpS approach is not limited to the CSC context, nor necessarily to workplace learning (in fact, we are currently working on a version for use in university courses). Nevertheless, the examples of its use presented in this paper are drawn exclusively from our experiences with its deployment at RPC.

PHelpS supports the training needs of CSC by helping workers as they carry out real tasks, on the job. The key feature of PHelpS is its ability to assist in locating an appropriate peer to help a worker who is having problems while using OMS to accomplish a task. PHelpS attempts to formalize and extend the role of peer help (in CSC) by locating suitable helpers possibly beyond those in existing informal networks.

At the heart of PHelpS is a knowledge representation scheme that captures at many levels of detail the authentic tasks carried out at CSC. By knowing the task step (either generally or specifically) at which the worker has run into trouble, PHelpS is able to find a peer helper who knows how to accomplish that step. Profiles of workers’ knowledge are kept as overlays on these task hierarchies. In addition to their role in helping PHelpS do its job, these task hierarchies can also be useful for other purposes. They can form the basis for performance support on the job, as workers check off task steps as they are accomplished. They can also be used for structuring training activities in the training centre, for allowing workers themselves to decide to learn more about a task in short interludes on the job as they see the necessity, for modelling the knowledge state of workers and helpers, and for structuring the communication during peer help interactions. PHelpS incorporates “minimalist” AI techniques in a variety of places: to assist with selecting peer helpers, to help maintain learner (and helper) models, and to help structure and mediate communication. More details about these task hierarchies are provided in the next section.

Task-Engineering

Each task commonly undertaken in CSC can be represented as a hierarchical set of steps. A fragment of a typical task (the Escorted Temporary Absence (ETA) task) is illustrated in Figure 1. The ETA task consists of a set of procedures and processes by which an offender may be granted an escorted temporary absence from the correctional facility (for example, to attend a funeral of a family member). There are approximately eighty distinct steps associated with this ETA task, some carried out using OMS and others involving individual or committee assessments and approvals. The task may span several days and possibly more than one correctional worker may be responsible for guiding the task through to a successful conclusion (either non-approval of the ETA or approval and detailed arrangements for the ETA). While policies and procedures for tasks like the ETA task exist within Corrections Canada, there was no existing detailed official task-hierarchy for any task when our research began. Our first steps involved doing task analyses of various tasks (in the sense of Shepherd, 1995) and constructing task hierarchies. Next we developed a facility called the *PHelpS Personal Assistant* (shown in Figure1) which enables workers to use task hierarchies as checklists in recording steps completed within their tasks. The task-based approach has proven effective in other domains like medicine (e.g., Vassileva's hypermedia office document system, Vassileva, 1996).

Task hierarchies are central to all activities involved in PHelpS. There are approximately twenty high-level tasks that are carried out at RPC. The current PHelpS prototype captures only two of these high-level tasks, with each task comprising up to two hundred task steps. It took about one person month to engineer each of these two tasks. It is our belief that the rest of the tasks can also be engineered in a similar time frame.

Task hierarchies provide a terse description of steps that must be carried out and the recommended sequence for achieving them. Some of these steps involve the completion of forms in OMS or consultation using information contained in OMS. For example, about half of the steps in the complete ETA task involve using OMS forms. Workers completing a task with the PHelpS Personal Assistant can utilize the task hierarchy as a checklist to record the subtasks that have been achieved. Task steps can be opened to a finer grain size, or left at a coarse grain size, depending on how much detail the worker needs to see in carrying out the task. The worker can actually check off the tasks undertaken (see the check marks in Figure 1) as a reminder of where he or she is in undertaking the task. The task checklist also is hyper-linked to OMS so that clicking on a task step can take the user to the appropriate OMS screen (see the “lightning” arrow in Figure 1). For example, the checklist item in Figure 1 containing the caption *Go to Sentence Management Screen* is hyper-linked to the relevant OMS screen. Finding the proper OMS screen without such hyper-links is a difficult task for many novice users of OMS, since it requires keying through sometimes several pages of textual menu lists of cryptic options.

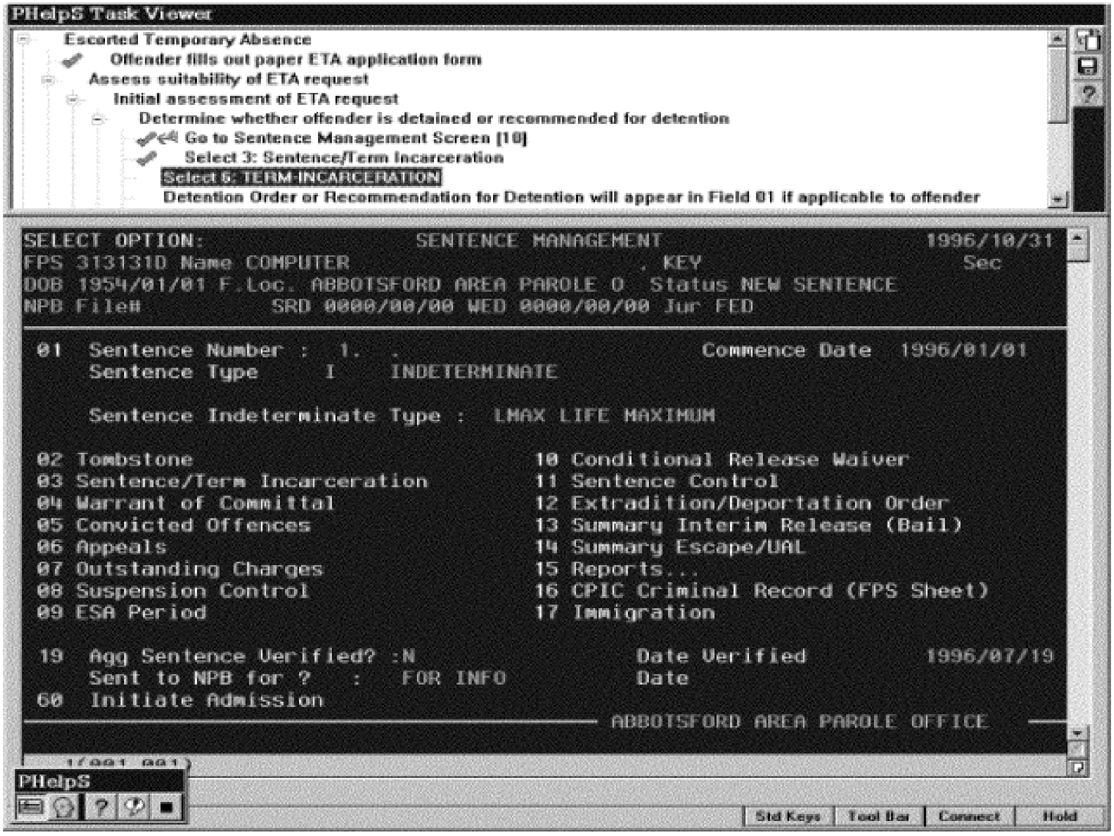

Figure 1. Personal Assistant: the Escorted Temporary Absence (ETA) Task Hierarchy.

The task-hierarchies serve as a reference point, providing context for help-requests. When a step in the checklist is causing difficulty, the user can request help in one of five forms. First, each task hierarchy item has a *hint* field, which gives a more elaborate pop-up explanation of the sometimes-terse descriptor. Second, *context specific help* can be made available in the form of links to relevant online manuals for OMS and for policies and procedures via the “Online Help” tab in Figure 2. Third, *case-based help*, where a set of relevant help cases (dialogues between workers and helpers in similar tasks) can be accessed by clicking on the “Case-Based

Help" tab in Figure 2. Fourth, *a human peer* can be contacted. Peer helpers are automatically selected subject to the constraints of willingness to help, ability to help, and availability to help. When a user clicks on the "Peer Help" button, the user initiates a process for getting help from a fellow worker in a peer-to-peer dialogue. Finally, a "Power User Help" OMS training professional or system support person may be consulted for help if necessary.

Our current work on the Help System in PHelpS has focused primarily on the fourth option above, providing peer help, with some effort directed toward case-based help, in part because it is of most interest from an artificial intelligence (AI) perspective. Figure 2 shows the Peer Helper Consultation, the most developed facility in PHelpS. When the user selects this option, the system creates a ranked list of potential peer helpers to assist with the current task. The current task step is the primary index for identifying helpers. A constraint algorithm, described in section 4.3, generates this list of candidate peer helpers by referring to the repository of helper models. It is important to note that the final choice of peer helper is left to the worker: PHelpS merely makes knowledgeable recommendations.

Working Scenario

To illustrate this peer help feature, assume a worker using the PhelpS Personal Assistant reaches an impasse at a task step in the ETA task, such as not knowing how to *select a term incarceration* (see the highlighted task step in Figure 1). He or she can mark the step as problematic (the **?** annotation-tool available on the right-top part of the screen) and request a peer helper (by clicking on the "peer helper" PHelpS button in the bottom left corner of the screen). PHelpS consults a knowledge base to locate a set of potential peer helpers within the organization who:

1. are knowledgeable about the problem area of the specific task,
2. are available to provide help in the time frame required,
3. have not been overburdened with other help requests in the recent past, and
4. have other characteristics critical to a successful peer help session, for example they speak the same language as the worker (approximately one third of CSC workers speak French as a first language, the rest English).

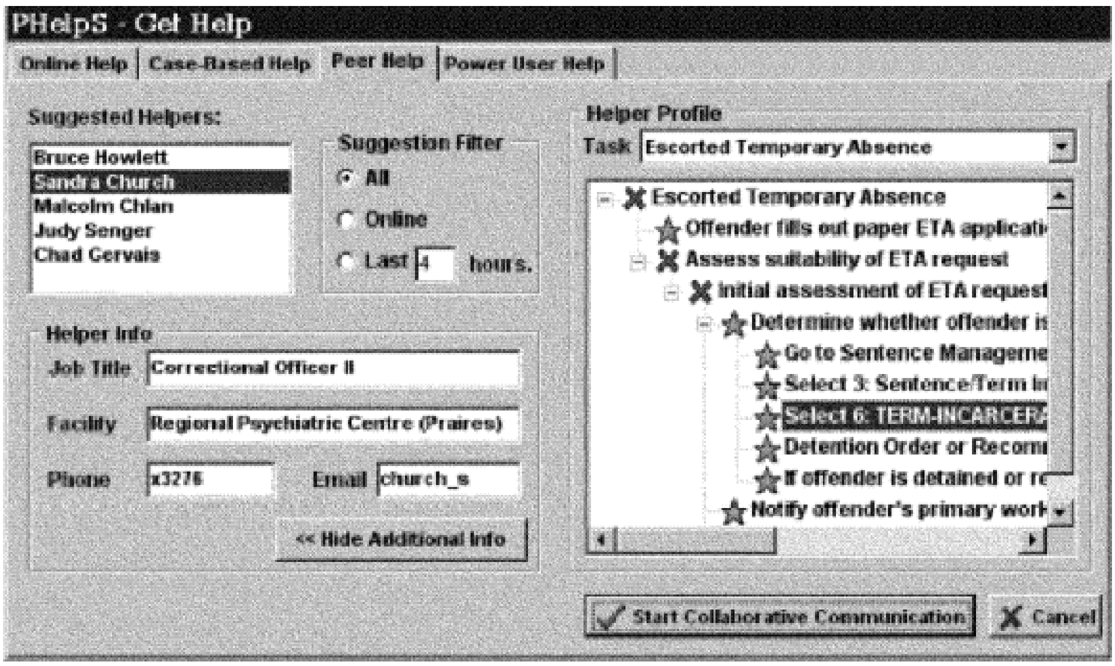

Figure 2. Peer helper suggestions.

The help request and these criteria form the inputs to a constraint solver embedded in PHelpS, which produces a set of candidate helpers ranked according to their suitability on these criteria. Figure 2 shows a ranked list of helpers resulting from the above help request. These few candidate helpers are drawn from a much larger pool.

The user selects his or her preferred helper from the candidate list (perhaps using the weights and/or other information about potential peer helpers available in the Helper Profile, see Figure 2). Once the helper is selected, a helper-helpee dialogue is begun.

The key to intelligently generating a reasonable candidate list involves maintaining knowledge profiles for every potential helper. The knowledge profiles are organized according to the tasks that need to be achieved within the organization. Thus, for each task hierarchy and each user, a hierarchically structured knowledge profile is constructed and annotated with the simple annotations *can help* or *can't help* attached to each task item. This knowledge profile along with other information about each potential peer helper (language spoken, rank in CSC, current availability, etc.) together constitute the user model of each peer helper. The process of creating and maintaining the user models is discussed in more detail in Section 4.2.

INSIDE PHelpS

In this section we will discuss the architecture, user modeling and functioning in PHelpS.

Architecture of PhelpS

The architecture of the PHelpS system is shown in Figure 3. It consists of two distinct modules: the Personal Assistant, which provides performance support, case-based help and peer help during task execution; and the Knowledge Update module, which permits prospective peer helpers and users to inspect and update the system's models of themselves. These components utilize three knowledge bases: the Task Hierarchy repository, the Learner/Helper Model repository, and the Help Case repository.

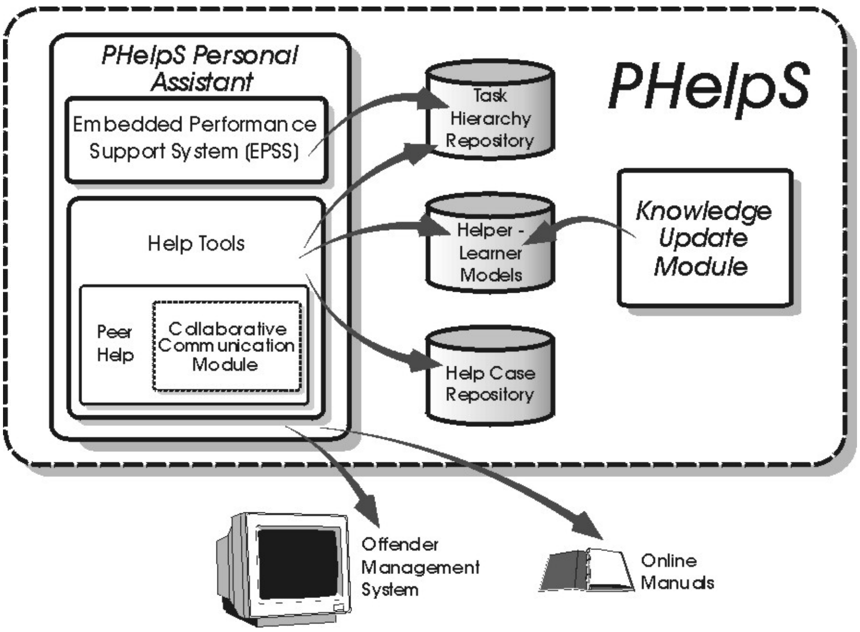

Figure 3. The PHelpS System Architecture

Personal Assistant

The Personal Assistant is an embedded performance support system (EPSS) (McGraw, 1994) with a suite of embedded help tools, including a peer help system. The Personal Assistant provides checklists of actions that correspond to specific routine tasks commonly performed within an organization. The Personal Assistant provides the means through which users can inspect and perform various task-oriented operations.

Knowledge Update Module

The Knowledge Update Module initializes and maintains the helper models as described in Section 4.2. Another use of the Knowledge Update module is in annotating task hierarchies. Decisions made while knowledge engineering task hierarchies arise from consensus among experienced end-users, and thus evolve over a period of time. As task hierarchies are evolving to their final form, users are invited to annotate various subtasks in any task hierarchy and other users are able to browse these annotations. Subtasks within task hierarchies are subject to ongoing change as regulations or policies change. Annotations in task hierarchies are one means of identifying the need for change, and also in helping to manage change.

Collaborative Communication Module

Presently PHelpS supports only passive collaboration where the system does not take an active part in supporting/enhancing collaboration. Such efforts fall under the realm of CSCW techniques.

As soon as the person needing help selects a peer helper, he / she initiates a dialogue. This dialogue can take place off-line (e.g. by telephone or personal visit) or on-line, depending on urgency and the predilections of the two people involved. Currently, PHelpS doesn’t supports on-line communication. The Collaborative Communication Module serves mainly as an architectural placeholder for the further development of the collaboration support facilities of the system.

We are currently examining how the system might be expanded to support active collaboration where the system tries to understand the needs of individual workers based on their models, to provide appropriate feedback and structured communication media, and finally, to drive peer collaboration towards a preferred model of collaboration. We are extending the collaborative communication module to support on-line dialogues between worker and peer helper as a mediated chat session with some facilities for knowledge sharing. PHelpS can, for example, provide context-based message templates to the user who can fill-up the template and send it off to the peer helper for either synchronous or asynchronous communication. Both helper and user can share the current task hierarchy and use it as an object of discussion during communication, as well as access each other's knowledge profiles and checklists (if permission of each user is given). The dialogue exchanged through the Collaborative Communication facility can be trapped for each case and becomes fodder for the case-based help facility. Capturing and indexing help cases can be supported by the task-hierarchy knowledge representation underlying PHelpS. Each completed help case will be examined by a moderator, edited to remove personal information, indexed to the appropriate subtasks where it might be applicable, sometimes generalized to include other information not in the original help dialogue, and finally committed to the Help Case repository. Of course, in situations where the Collaborative Communication module is bypassed in favour of telephone communications or personal visits, it will be impossible to automatically capture help cases.

We are also in the process of incorporating Microsoft NetMeeting software in the PHelpS communication module. In this environment, communication can include audio and video conferencing as well as application sharing and chat-facilities. Within CSC, there are network challenges that might preclude audio and video conferencing at this time and there are serious security and accountability challenges that might preclude application sharing as well. Nevertheless, other organizations might be better suited to this kind of complete collaboration support environment for peer help. The PHelpS approach in conjunction with a point-to-point collaboration tool makes a strong partnership.

User Modelling

As can be seen from the above section, user modelling is at the heart of PHelpS. In this section, more details about the PHelpS approach to user modelling are presented. Each user model contains information ranging from the task-specific (e.g., the knowledge profiles showing the tasks the peer helper can perform and the level of capability in carrying out each coarse and fine-grained step in these tasks, etc.) to the very general (e.g., the peer helper's age, gender, position in CSC, linguistic fluency, current login status; the number of times the peer helper has provided help; etc.).

Overlay with the Tasks

The task-specific information forms the kernel of the knowledge profile and is essentially an overlay on the various task hierarchies recording the degree of capability the person has with each important task and subtask. Within the ETA task hierarchy, fifteen of the steps are considered important subtasks from a helper selection perspective. The other finer-grained steps (such as knowing how to press the "esc" key or to enter an offender's name) are considered irrelevant when computing a helper's knowledge profile. Thus, skill with these 15 important subtasks would be recorded in each helper's knowledge profile for the ETA task. Factors influencing the system's belief about the helper's skill include the number of times the specific subtask has been completed in the recent past, the number of times he or she has given help on the subtask in the past, and the number of times this help was useful or not useful to the worker requesting help. These counts are combined into a composite numeric measure of helper capability for each subtask.

Inspectability

The user models in PHelpS are inspectable by a variety of users as in (Paiva et al., 1995) and (Bull, et al., 1995). Inspectability is used in two places: (i) for the worker to inspect his or her own user model; and (ii) for the worker to inspect the user model of any potential peer help partner. Figure 2 illustrates the form by which users can inspect and maintain their own user model. Each step in each task can be annotated by the user with a * or an X indicating that the user feels he or she *can help* or *can't help* on the step (see Figure 2). Users also have the option to inspect and update the components of their own user capability scores. That is, details such as how many times the user has performed a task, how often the user has given help on this task etc., can be inspected and updated or corrected by the user. Thus errors in users' overall helper capability scores for any particular task (thresholds denoted by *s and Xs in figure 2) can be corrected by the users themselves.

Similarly, a worker needing peer help who wants more information about a potential peer helper can do so by clicking on the "Show Additional Info" button. When this button is clicked the worker can see information from the user model of a potential helper. Not all information will be shown; for example, only the knowledge profile for the current task will be displayed, not the knowledge profiles of all tasks in the user model. In this way, a potential helper's level of expertise can be determined before that helper is contacted.

The knowledge sharing facilities implemented in PHelpS Talk are related to the same checklist / knowledge profile task hierarchies used in peer helper identification. That is, the helper is provided with the marked-up checklist of the worker seeking help. In this way, the current help context is easily communicated. The helper can decide where help is needed and how many steps have been completed. Only after the potential helper agrees to help, the helpee will get his/her phone number and establish a telephone connection. In addition, in the same way the worker needing help can browse a version of the helper's user model before contacting him or her, the helper can also inspect a version of the user model of the worker seeking help. The helper may browse this model and determine the level of knowledge of the worker (in terms of the worker's annotations of *can help* and *can't help* for the task at hand) to adjust explanations appropriately. That is, a helper would provide a different kind of advice to a novice totally unfamiliar with the task than he or she would provide to a very experienced worker. As a privacy measure it should be noted that PHelpS provides users with the ability to hide their profiles from inspection, if they wish.

Initialization and Updating

Information in the user model changes over time, ranging from fine-grained temporal intervals (e.g., minutes or hours for login status, or to carry out a particular task step), through longer periods (e.g., days or weeks for learning a new task, months or years for changing position in CSC, etc.).

The user model is thus multi-dimensional along two principal axes: temporal and task-relatedness. The task-relatedness dimension is explicitly hierarchical, following the natural topography of the various task hierarchies (see Figure 1). The temporal dimension is implicit, according to what particular information is in the knowledge profile at any given time. PHelpS thus must concern itself not only with initially stocking the user model for each peer helper, but also with maintaining the accuracy of the user model over time.

For time varying information, such as login status, work schedules, or number of times the peer helper has dispensed help, user model maintenance is straightforward, since this information can be gleaned by the system. It is more difficult, however, to track changes in the task-specific information in the knowledge profiles. Nevertheless, the Knowledge Update module does provide automated support for such maintenance. Since each user is also a potential helper, as users carry out various tasks and use the task checklists, the system increases its confidence in their knowledge of the task. Thus, knowledge profiles can be updated by using information from the EPSS checklists. Marking these checklists not only reifies the task steps in the mind of the peer helper, a useful reinforcement activity for just-in-time training on that task, but also can provide a sort of *temporal extension* of the peer helper's user model, recording the current state of his or her problem solving. In addition, each time a helper is called upon for assistance, the number of times helped is automatically updated. If the user requesting help finds the interaction helpful, the helper's ability scores on the tasks related to the requested tasks are increased.

The second support provided by PHelpS for peer helpers to keep their knowledge profiles up-to-date is a by-product of the peer help session itself. Since peer helpers and those being helped can inspect each others' knowledge profiles in order to better understand each other and to facilitate communication, the mutual interaction that ensues can clarify inaccuracies in the profiles of both parties.

As can be seen there is a major responsibility for PHelpS users to carry out profile maintenance activities. This, in fact, is how we "solve" the problems of user model maintenance that usually prove so difficult in many user and student modelling applications (Huang et al., 1991). We are not the first to suggest that users should help maintain their own user models (Paiva et al., 1994), but in order for this to work, the context must be right for them to do so. The widespread use of PHelpS in an organization should encourage workers to ensure that the accuracy of their own user models is maintained. They should want to keep an up-to-date record of their capabilities on various tasks (i) so that when they need help it can be targeted appropriately to take into account their strengths and weaknesses; and (ii) so that when others contact them for help, they will only be contacted on topics about which they know something. In short, an inaccurate user model will cost a worker time both when seeking help and when being sought for help.

If user model maintenance proves to be more problematic than we hope, there are some natural extensions that can be carried out to help in maintaining the knowledge profiles. If users prove unwilling to check off their task steps at a fine grain size, but are willing to do so at a coarse grain size, then it would be possible to *infer* that the fine grained steps are known if the coarse grained ones are. If the users prove unwilling to check off any task steps at all as they carry out tasks, the hyper-links from the task hierarchies to the OMS system can be traversed in reverse to update a task hierarchy by inference from particular activities carried out in OMS. Such update strategies may require the use of recognition and diagnosis methodologies from artificial intelligence. Sometimes these methodologies seem far removed from easy applicability, but, in this domain, the task-specificity of the activities should allow the reasonable adaptation of methodologies like granularity-based recognition (McCalla & Greer, 1994) and model/knowledge tracing (as in the LISP Tutor, Corbett & Anderson, 1995), which are explicitly designed for task-specific, real world domains.

Peer Selection

In addition to being multi-dimensional, the user models in PHelpS are also multi-purpose. Their principal purpose is to help find an appropriate set of peer helpers for anybody needing help on a particular task. The primary requirement to be met in the choice of an appropriate helper is to find a peer who knows the task, and in particular the task step on which the person needs help. Secondary requirements are that this person be available in the time frame in which help is required, and that the person has not been asked to help others too often in the recent past. The primary requirement is fulfilled through use of an algorithm that matches the current task hierarchy checklist of the worker needing help with the knowledge profiles of potential peer helpers. The matching function currently operates as a constraint solver, which considers the subtask (step) where help is requested, the sibling subtasks, the subsequent subtasks, and the higher-level parent subtasks. For these subtasks, a set of best candidate helpers is computed based on their ability to help, and the function returns a list of all those helpers who exceed a certain threshold.

In Hoppe’s terms (Hoppe, 1995), peers are chosen using a “complementarity criterion”, i.e., a peer of high competence on the task on which help is needed is chosen to help. Note, that unlike in Hoppe's learning domain, in our workplace environment choosing peers based on a “competitiveness criterion” is unlikely to be very effective. In future research, we may investigate incorporating information taken from the longer-term knowledge profile of the person needing help.

The list returned from the profile matching function is further pruned by removing peer helpers who are not available or who have been overused, according to the secondary requirements above. The remaining list of potential peer helpers is then presented to the person needing help, as in Figure 2.

This is where a second purpose of the user models becomes important: having the person needing help scan through the knowledge profiles of the potential peer helpers in order to choose somebody who has the specific combination of qualities that they prefer. These qualities may involve the peer helper having a particular pattern of task knowledge, or perhaps may involve some of the less task-specific characteristics, such as having the same gender, being of approximately the same age, or being in a similar position (same job title or same union). The literature on collaborative work and collaborative learning suggests that such characteristics are often important in facilitating effective collaboration (Dillenbourg et al., 1995). However, regardless of these statistical trends, we believe it is important to leave it to the workers themselves to decide if these are important to them in the current help context.

The third purpose for the user models is the dual of the second purpose, namely giving the peer helper who has been contacted the ability to look at the knowledge profile of the worker needing help (as well as the task hierarchy checklist that represents that worker's current state of task completion). The peer helper has the right to refuse to help. But, if he or she does decide to help, the worker's profile (and task hierarchy checklist) and the helper's profile can form the basis for mutual understanding between the two parties.

In particular, during their discussions the peer collaborators can adjust the level of their conversation to take into account differences in knowledge between the two, can point to various parts of the task hierarchies in their respective profiles, and can use the verbal descriptors in the task hierarchies as a shared vocabulary.

PhelpS as a Training System

While PHelpS may seem to be an EPSS with some intelligent help facilities, it can also be used explicitly for training, say in a course. Workers can use PHelpS to practice simulated tasks during this training, and thus learn away from the pressures of their day-to-day activities, but nevertheless in a supportive, situated, and authentic environment. We have begun the deployment of PHelpS in just such a way to train employees in the tasks of the so-called "Programs Module" at CSC.

The Programs Module tasks include creating treatment program descriptions, admitting offenders to programs, reporting on treatment progress, summarizing the impact of treatment when the offender completes a treatment program, etc. While comprehensive knowledge of OMS for these tasks is currently possessed by few employees, all employees are now expected to be able to perform most of the sub-tasks in the Programs Module.

PHelpS should work equally well in the training centre as it does in workplace support. Workers learning a new task can be guided through simulated cases by utilizing the PHelpS Personal Assistant. They can request help from fellow learners or from the trainer through the PHelpS peer help interface. They can also request help from workers who are on the job outside of the training classroom, a notion of a spatially distributed classroom, one without walls. Moreover, the PhelpS approach also allows the notion of a course distributed in time; that is it becomes possible for workers to engage in training at their normal workstations and at times of their own choosing. This enables a notion of just-in-time training, where learning a new task can be opportunistically scheduled.

A PHelpS prototype is currently operational and deployed on a limited basis at the Regional Psychiatric Centre of CSC. We have conducted a number of usability experiments and formative evaluations with the system and have continued to refine its interface and functionality. Preliminary experiments in using PHelpS in the training context indicate that the task-oriented nature of training that PHelpS encourages is quite different from normal classroom training used to date at CSC. Learners using PHelpS are more focused on learning the procedural skills to complete a task, have a reduced cognitive load with respect to OMS functions and thus, are more able to focus on the kinds of judgements required in performing tasks.

As one worker commented after using PHelpS: "It is really useful, guiding you step-by-step. Before they just threw the information at you and you were overwhelmed." Although we have not yet experimented with PHelpS in a self-directed learning setting, we have some evidence to indicate that it will be feasible for learners to attempt training even when physically removed from the training centre. There is also some evidence to suggest that PHelpS can assist with transfer of knowledge from one task to another. One worker, who reached an impasse while attempting to solve a not-yet-engineered task (i.e. assigning a case management officer to an offender), accessed the PHelpS task hierarchy for a related task (i.e. assigning a primary worker to an offender) and inferred a solution to the impasse without requiring any further help.

EVALUATION

A recent usability study was conducted using PHelpS at RPC in one of the OMS training courses. The goal of this experiment was to provide hands-on training with OMS and PHelpS together and to test the effectiveness of PHelpS. The study consisted of two sessions with a total of four trainees, a small but representative group. The first session lasted for 75 minutes introducing OMS and PHelpS to the trainees. Realistic simulated tasks were given to the trainees to perform during the next 75 minutes. The class was spatially distributed: the trainees were in the training room and the helpers were on duty performing their day-to-day activities. The experimenters recorded the proceedings using audio (telephone conversations and debriefing), video (of the entire proceedings), trickle files (of the keystrokes) and "observation notes".

Analysing the collected data resulted in a number of useful observations that could help direct future developments in PHelpS.

- All the trainee workers used the PHelpS task hierarchy checklists. Three different types of checkmarking modes were observed: serial, burst and random. In serial mode, the workers checkmarked a task step immediately after performing it. In burst mode, the workers checkmarked a series of task steps after performing many task steps. In random mode, the workers checkmarked task steps only when they required peer assistance. Random mode checkmarking seems to be very common. This points out that the workers perceive PHelpS as a help tool that is best used only when they are seeking peer help. This raises some doubts in the possibility of using PHelpS to track the problem-solving process, as described in Section 4.2, so probably other means of tracking the time-dependent changes in user's abilities have to be explored.

- During the experiment, there were seven help requests from the trainee workers leading to seven help sessions, out of which five were successful. The failure of the remaining two sessions was attributed to the failure to communicate the full context of the help requests. In one help session it took several minutes to establish the correct context and only a few minutes to solve the problem. Some excerpts from this help session are given below.

  Trainee: Yeah, but do I need, like, it says, to schedule six full-time participants. Do I need the names of six full time participants?

  Helper: Do you have the names of the people?

  Trainee: Ahh, not so far. No.

  Helper: Oh, OK.

  Trainee: So, OK there is a scheduled program in the system already right. What, what is the program called?
  (the trainee leads the helper astray and the context is not mutually understood)
  (5 minutes elapsed)

  Helper: They never made any mention of a list of guys or anything like that?
  (the helper tries to bring the trainee back to the correct context)
  (3 minutes elapsed)

  Helper: So they want you to assign a guy, assign six people to the relationships program.

  Trainee: Ah. Schedule. Schedule six full time participants to attend the relationships program.

  Helper: Oh. OK - ahhh *(cough).* Go back to the main menu.
  (problem is identified and the helper proceeds with the correct help.)

- All trainee workers preferred to choose "a knowledgeable peer" and were satisfied to choose the person suggested by the system, even though they were sometimes strangers. In addition, the trainee-workers preferred to ignore the job hierarchy as long as the helpers were willing to collaborate. In one of the help sessions, a worker picked a third ranked helper from a list of helpers suggested by the system. Eventually that help-session turned out to be a failure because the helper had little expertise in that particular context.

We observed some very encouraging, but also thought-provoking feedback about the PHelpS approach during our experimentation at the Regional Psychiatric Centre. Learners in post hoc debriefing sessions made comments such as

"PHelpS is very user-friendly ... like step-by-step processing, fully knowing what to do next ... hierarchy is good"

"... getting the context right is tough"

"hands-on approach is good ... some staff have more knowledge on certain topics ... it is better to contact them when stuck"

"... being able to help someone else is helping you"

"I will contact a knowledgeable stranger to a less knowledgeable friend"

"Looking forward to seeing it (PHelpS) on the units"

Our preliminary testing has shown some promise of the PHelpS approach. In the next section we discuss concerns that have arisen during our exploration of PHelpS.

DISCUSSION

There are some concerns that have to be addressed, including:

- how sophisticated the task hierarchies need to be to allow the selection of an appropriate peer helper and to capture a wide variety of tasks,
- how sensitive the matching algorithm needs to be to identify appropriate peer helpers,
- how change in tasks and workers capability in carrying out tasks can be handled,
- how communication between helper and the learner can be facilitated,
- how additional kinds of help can be incorporated,
- whether wide spread acceptance of PHelpS can be gained by both workers and management, and
- whether PHelpS, once deeply ingrained in an organization, leads to abuse of workers' privacy, and is misused to monitor workers' efficiency and effectiveness.

There are three major maintenance concerns that need to be addressed with respect to the task hierarchies. To start with, the engineered task hierarchies need to be verified for their accuracy. Second, the task hierarchies need to be updated from time to time to ensure consistency of the hierarchies across the different units within a CSC organisation, across the different CSC organisations in Canada and across the different user models that are maintained in PHelpS. Third, we are trying to identify ways to represent tasks that cannot be naturally represented in hierarchies. This would help us generalise the design of the peer help approach and replicate PHelpS-based peer support for different organisations with different work cultures from that at RPC.

We would also like to explore enhancements to the PHelpS system to incorporate more facilities to handle change. We think that belief revision techniques from artificial intelligence could be adapted to significantly improve the ability to keep knowledge profiles up-to-date in the face of changes in tasks and workers' understanding of these tasks. We are also actively experimenting with augmenting the other kinds of help the system can provide, including more sophistication in the "Frequently Undertaken Interactions" (FUI) Help Case subsystem, and more refinement in online "canned" help to capture whether such help is at the technical or policy levels, thus providing the worker with further choice as to the kind of help he or she feels is needed in the current situation. All such help is made more coherent and more useful by being indexed through the current task hierarchies to ensure its relevance to current worker goals (Carroll, 1990). A final area we are currently exploring for system enhancement is improving communication between the helper and the worker being helped. As mentioned above, we are incorporating advanced communication software such as NetMeeting. We also are working on expanding the role of the system during the help session by allowing it to actively intervene in and influence the interaction (by making use of the information in the user models of each participant in the session). The system itself becomes a learning collaborator in the interaction in the manner of Chan and Chou (Chan et al., 1995). A big remaining communications problem is coming up with techniques that help establish mutual context at the beginning of a help session (beyond sharing user profile information).

Another major concern for the success of PHelpS is the motivation of the workers to support each other. Peer help, formal or informal, is a common trend and is being encouraged in many corporate cultures. PHelpS interactions are considered informal in the sense that the workers choose their helpers without being absolutely explicit as to what kind of help they need and without knowing much about the candidate helpers. PHelpS attempts to provide structured peer support by collecting information about the workers and helpers with respect to the task hierarchies, their work habits and other related information. The workers know that there are implicit rewards associated with peer help. In addition, the organisation should come up with explicit rewards to encourage peer support. For instance, RPC has recognised a group of “power users” who volunteered to work under a new initiative called the “peer-mentor program” within RPC. Such corporate recognition can be motivating for potential peer helpers.

With respect to privacy issues, the individual user models employed by PHelpS to help workers, could also be used by management to monitor workers, perhaps to the workers' perceived or actual disadvantage. Hopefully, management would be deterred from such monitoring by the likely result that workers would immediately cease using PHelpS at the first sign that their privacy was being violated, and thus all the potential advantages of the PHelpS approach to an organization would be thrown away by its misuse. This isn't as big a problem at CSC as it would be in other organizations, since CSC workers are accustomed to the monitoring of their activities while using OMS, and in that accountability for accessing sensitive information about convicted criminals (particularly high-profile offenders) is part of the workplace culture. Nevertheless, even in more controlled environments such as CSC, worker empowerment and wholehearted worker acceptance are critical to the success of PHelpS.

RELATED WORK

There have been numerous approaches in the field of AI and Education aimed at providing peer help for the learner. Most of them, however, try to create an artificial peer, i.e. an intelligent component or agent, who collaborates with the learner, an approach that was originally proposed by Self in 1986. Examples of such artificial peers are Gilmore & Self’s (1988) or Dillenbourg & Self’s (1992) artificial co-learners, Chan & Baskin’s learning companions (Chan & Baskin, 1990), and Aimeur & Frasson’s “ troublemaker” (Aimeur et al., 1997). All of these systems are focussed on collaborative problem solving (and consequently have a very restricted domain of application). They generate help and utterances themselves (using their knowledge bases) and decide when to interfere (using their pedagogical strategies). In this sense they are traditional “Intelligent Tutoring Systems”. Our approach to providing peer help differs significantly from these classical approaches.

First, our domain is not restricted to problem solving domains like that at CSC. In fact, the subject domain of PHelpS can be highly varied: the only requirement is the existence of some kind of domain structuring (into tasks, skills, topics, or concepts) to which help-requests can be indexed. We are currently experimenting with a version of PHelpS designed for use in university courses.

Second, there is minimal automatic generation of computer-based help, so the system can perform with a less extensive knowledge base and less sophisticated reasoning mechanisms than in the learning companion approaches. Third, the system doesn’t interfere with the help dialogue and doesn’t make pedagogical decisions. It is activated only by an explicit request from the user. In this way PHelpS involves human intellect at all points which are currently considered as the “Achilles’ heels” of AI-based learning environments: the diagnosis of a student’s knowledge, pedagogical decision-making, and generating the instructional content.

PHelpS can be compared with other student model-based approaches for selecting an appropriate human helper. Hoppe's (1995) COSOFT project is the first ambitious project to address several issues related to the use of student modelling in order to parameterize human-human collaboration. These issues include the composition of a learning group from a known set of students, and especially the selection of a peer-helper, the identification of problems to be dealt with in a collaborative session or the selection of tasks that are adequate for a given learning group. Hoppe's approach has been primarily targeted at exploring possible improvements to group student modelling to support human collaboration. It focuses on a limited domain since it employs classical ITS diagnosis, representation and matching techniques. Unlike COSOFT, the user modelling approach employed in PHelpS doesn't rely entirely on computer diagnosis, but on human feedback, since the interface is task-oriented. This makes it easily transferable to a different domain.

The task-based approach employed in PHelpS can be compared with other task-based approaches for designing the interface to information systems and applied simultaneously as a basis for user modeling, most prominently HynecosUM (Vassileva, 1996). The comparatively smaller number and dynamics of the tasks performed by users distinguishes the domain of both applications. In office documentation systems, typically the organization of work assumes well defined tasks associated with actors performing them.

In HynecosUM, like in PHelpS, the relative stability of task-hierarchies has allowed minimalist AI techniques to be used avoiding the need for deeper cognitive modeling of the tasks and machine learning for maintenance of the task hierarchies in time. In HynecosUM the users designate their current task, since they interact with the information system via the task-hierarchies. The information system (i.e. the database) is detached and kept independent. This is also similar to PHelpS. There are also similarities in the user modelling approach: the user model is a qualitative overlay over the task-hierarchy, but while in HynecosUM it represents the level of experience of the user on the task, in PHelpS it represents the ability of the user to give help on this task. The updating of the user model is in both cases taking into account various external sources, rather than internal reasoning. Similarly to PHelpS, the evaluation of HynecosUM has showed a better user performance with the task-based interface than with interface organized around the data-structure.

FUTURE DIRECTIONS OF RESEARCH

We believe that peer help is a valuable and cost-effective resource not only in corporate training, but also in a university environment. Thus, we are exploring how to apply PHelpS in a university setting. Universities, experiencing large growths in student enrollment, are faced with the difficult problem of providing adequate help resources for their members i.e. the staff, faculty and students. Help resources are needed at an institution-wide and also at a course-specific level, due to the limited time of instructors to provide help and answer questions. Computer technology offers several approaches to facilitating and providing the necessary personalized help resources that can be made available to a mass audience. By deploying IntraNets in Universities different kinds of resources (lecture notes, exercises, quizzes, syllabii, etc.) can be made available on request to any student. There are numerous positive examples of implementing on-line course materials and discussion groups at other universities, for example, the Virtual-U Project (Virtual-U, 1997), WebCT (Goldberg, 1997), Quorum (Canas et al., 1995).

However, merely providing access to appropriate material via a network doesn't solve the problem of providing help. One way to decrease the load of teachers is to provide conditions for students to help each other. Peer help has many pedagogical advantages. For example, it promotes the socializing of students in the context of work and increases their motivation by giving social recognition for their knowledge and helpfulness. Computer technology can be applied to support peer help. There are many Computer Supported Collaborative Work (CSCW) tools that facilitate communication among peers. However, they rarely ever provide personalized help on demand. We have integrated PHelpS with an existing discussion forum, FAQ server, a set of on-line materials and adaptive help facility, in order to create an "Intelligent Helpdesk" for our students. We are currently using this Helpdesk in the Spring Semester of 1998 to support about 400 students in an introductory course in computer science at the University of Saskatchewan. We hope that this will provide a huge amount of experimental data, which, when evaluated, may point to new directions for future improvement and development of PHelpS.

Currently, we envisage two main trends for future development of PHelpS: extending the support of the system from peer help to full peer collaboration and increasing the intelligence of this support. More specifically, we plan to pursue the following directions:

- *Ensuring common workspaces for helper and helpee in PHelpS.* By ensuring a shared clipboard, whiteboard, internet phone and providing the possibility for sharing applications, the context in which the user's problem has arisen will be more readily shared between the user and the peer helper. This will make the communication among all participants easier. Also in this way the system will be able to track individual interactions for later analysis.

- *Providing pedagogical support for helpers.* Based on the user model, the system can provide pedagogical support for the helper. As a simple example, it can suggest what kind of communication channel would be more suited for the peer-help session (e.g. shared whiteboard, or internet phone, or just a chat environment) based on knowledge of the computational environment and past technological sophistication of the helpee. It can also suggest a level of explanation based on how many times the helpee has asked for help (people who constantly ask for help could be given very general help in order to force them to try to solve the problem themselves). One of the co-authors of this paper, (Kumar, 1997) is investigating a two-component model for providing support to the helper. The "Librarian" component provides a "short list" of appropriate reference materials (web-pages, FAQ-questions/answers, etc.) for the helper to use in answering the request. The "Pedagogue" component provides advice about how to handle the request for best effect (e.g. to point the helpee to an appropriate FAQ vs. answering his/her request). Both components rely on knowledge of the user requesting help as well as the particular help request, which allows appropriate resources to be indexed and retrieved.

- *Integrating collaboration tools with PHelpS.* Collaborative tools such as Microsoft NetMeeting would be integrated with PHelpS to support collaborative activity and teamwork in corporate training. On the basis of the user models, the collaboration component could select not only one peer helper, but several partners to form a team for solving the problem or answering a help-request. Also, given a catalogue of roles and assignments (problems), this component could create teams of users and select appropriate collaborative roles for them to cope with the assignment (problem). The selection of partners for a team could be based on the principles of *complementarity* or *competitiveness* of skills proposed by Hoppe (Hoppe, 1995), and implemented by (Mühlenbrock et al., 1997), (Ikeda et al., 1997).

- *Decomposing the architecture into smaller intelligent components* (plug-ins, or agents). PHelpS readily lends itself to an agent architecture (Nwana, 1996), (Vassileva, 1997), where the components can pursue their own goals. For example, there could be an agent associated with every user who filters help requests from the agents of other users. Each personal agent would be equipped with pedagogical knowledge and knowledge about the user (a detailed user model). Based on this knowledge and the user's current goal or request, the personal agents would search for appropriate other agents that can fulfill the user goal. The agents can communicate via a broker or could be completely decentralized. The matching of requests and offers would happen via negotiation among the agents.

In summary, we are pursuing a simultaneous development of PHelpS in two main directions. “In breadth”, we are applying the system in new domains, for example, in University teaching; we are integrating it with other tools like on-line adaptive help, course-materials, discussion forum and FAQ facility; and we are employing Computer-Supported Collaborative Work tools to facilitate sharing workspaces and applications and rebuilding the system in a highly flexible and modular way. “In depth”, we are using artificial intelligence techniques to amplify the abilities of the HelpDesk in diagnosis, pedagogy and collaboration support. We believe this provides us with a broad and interesting research perspective, that should result in the construction of flexible, usable, robust and sophisticated tools to support human learning that are characterized by their ability to react to individual differences among learners.

CONCLUSIONS

The PHelpS approach has the potential to give real substance to the notion of the “learning organization”. By using technology to support human-human collaboration, the spread of knowledge takes place naturally as the informal peer help networks supported by PHelpS distribute new ideas and techniques throughout the organization. Learning is *leveraged* by the technology, not *imposed* through some external agent such as a traditional ITS system or CAI-based teaching styles. Learning happens authentically in the context of real tasks. Learning happens collaboratively, both through being helped and through helping. In some sense the organization itself can be said to be learning, since the knowledge of how to carry out tasks is rapidly distributed and soon becomes ubiquitous. Interestingly, unlike many information technologies which become bogged down if the number of users grows too big, PHelpS works better the more people who use it, since a large database of potential peer helpers allows a finer grained selection of appropriate ready, willing, and able helpers.

Acknowledgements

We would like to thank the management and staff of the Prairie Regional Psychiatric Centre of the Correctional Services of Canada for their active support and participation in the PHelpS project to date. In particular, we thank Ray Tkatch and Brenda Parkinson for their active involvement in helping to learn about RPC problems and Marcel Chaisson, the Executive Director of the Prairie Regional Psychiatric Centre, for his personal support for our PHelpS work. We would also like to acknowledge the Canadian TeleLearning Network of Centres of Excellence for financial support of the project.

References

Aimeur E. & Frasson, C. (1996) Analysing a New Learning Strategy according to Different Knowledge Levels”. *Computer and Education, An International Journal*, **27**, 115-127.

Bull, S., Brna, P., and Pain, H. (1995) Extending the scope of the student model. *User Modeling and User Adapted Interaction* **5**: 45-65.

Canas, A., Ford, K., Hayes P., Brennan, J., Reichherzer, T. (1995) Knowledge Construction and Sharing in Quorum, in Greer J. (Ed.) *Artificial Intelligence and Education, Proceedings AIED’97*, Charlottesville, VA: AACE, 218 – 225.

Carroll, J.M. (1990) *The Nurnberg Funnel*. Cambridge, MA:The MIT Press.

Chan T.W. & Baskin A. (1990) Learning Companion Systems. In C. Frasson & G. Gauthier (Eds.) *Intelligent Tutoring Systems: On the Crossroads of AI and Education*. Norwood, NJ: Ablex, , 6-33.

Chan, T., Chou, C. (1995) *Simulating a learning companion in reciprocal tutoring systems*. In Proceedings of the International Conference on Computer Support for Collaborative Learning. [On-line] Available:http://www-cscl95.indiana.edu/cscl95/chan.html

Chi, M.T.H., de Leeuw, N., Chiu, M.H., La Vancher, C. (1994) Eliciting self-explanations improves understanding. *Cognitive Science* **18**, 439-477.

Collins, J. Greer, J., Kumar, V., McCalla, G., Meagher P., Tkach, R. (1997) Inspectable User Models for Just in Time Workplace Training. In A. Jameson, C. Paris, C. Tasso (Eds.) *User Modelling, Proceedings of the UM97 Conference*, Wien-New York: Springer, 327-337.

Constant, D., Sproull, L., Kiesler, S. (1996) The Kindness of Strangers: The Usefulness of Electronic Weak Ties for Technical Advice. Organization Science, **7**, 119-135.

Corbett, A.T. and Anderson J.R. (1995) Knowledge tracing: modeling the acquisition of procedural knowledge. *User Modeling and User Adapted Interaction*, **4:**253-278.

Dillenbourg, P. & Self, J. (1992) A Computational Approach to Socially Distributed Cognition. *European Journal of Psychology of Education*, **VII**, 353-372.

Dillenbourg P., Baker, M, Blaye, A., O'Malley, C. (1995) The evolution of research on collaborative learning. In Reimann, P and Spada, H. (eds) *Learning in Humans and Machines*. London: Elsevier

Gilmore, D. & Self, J. (1988) The application of machine learning to intelligent tutoring systems. In J. Self (Ed.) *Artificial Intelligence and Human Learning: Intelligent Computer Assisted Instruction*, New York: Chapman and Hall, 179-196.

Goldberg M (1997) WebCT – Word Wide Web Course Tools [On-line] Available: http://homebrew1.cs.ubc.ca/webct/

Greer J., McCalla, G., Kumar, V., Collins, J., Meagher, P. (1997) Facilitating Collaborative Learning in Distributed Organizations. In R.Hall, N.Miyake, N.Enyedy (Eds.) *Computer Support for Collaborative Learning*, Proceedings CSCL'97, 73-82.

Hammer, M. (1996) Beyond Reengineering. How the process centered organization is changing our work and our lives. *Harper Business*.

Hoppe, H.-U. (1995) The Use of Multiple Student Modelling to Parameterise Group Learning, in J. Greer (Ed.) *Artificial Intelligence and Education, Proceedings of AIED'95*, Charlottesville, VA: AACE, 234-241.

Huang, X, McCalla, G, Greer, J, Neufeld, E (1991) Revising deductive knowledge and stereotypical knowledge in a student model. *User Modeling and User-Adapted Interaction* **1**: 87-115.

Ikeda, M., Go S., Mizoguchi, R. (1997) Opportunistic Group Formation, In B. duBoulay and R. Mizoguchi (Eds.) *Artificial Intelligence and Education, Proceedings of AIED'97*, Amsterdam: IOS Press, 167-174.

Kumar V. (1997) CSCL Environment for Distributed Collaboration, presented at the Doctoral Consortium of the *Conference of Computer Supported Collaborative Learning CSCL'97*, Toronto, December, 1997.

McCalla G.I. and Greer, J.E (1994) Granularity-based reasoning and belief revision in student models. In *Proceedings of User Modeling'94*, 39-62.

McCalla, G.I., Greer, J.E., Kumar, V.S., Meagher, P., Collins, J.A., Tkatch, R., Parkinson, B. (1997*) A Peer Help System for Workplace Training. In Proceedings of the 8th World Conference on Artificial Intelligence in Education, AI-ED 97*, Amsterdam: IOS Press, 183-190.

McGraw, K.L. (1994) Performance Support Systems: Integrating AI, Hypermedia and CBT to Enhance User Performance. *Artificial Intelligence in Education*, **5**:3-26.

Mühlenbrock M., Tewissen F., Hoppe H.U. (1997) A Framework System for Intelligent Support in Open Distributed Learning Environments, In B. duBoulay and R. Mizoguchi (Eds.) *Artificial Intelligence and Education, Proceedings of AIED'97*, IOS Press: Amsterdam, 191-198.

Nichols, D. (1993) Intelligent student systems: Learning by teaching. In P.Brna, S. Ohlsson and H. Pain (Eds.) *Artificial Intelligence and Education: Proceedings of AIED'93*, Charlottesville, VA: AACE, 576.

Nwana H. (1996) Software Agents: An Overview, *Knowledge Engineering Review*, **11**, 1-40.

Paiva, A., Self, J., Hartley, R. (1994) On the dynamics of learner models. *In Proceedings of the 11th European Conference on Artificial Intelligence*, 163-167.

Paiva, A. Self, J., and Hartley, R. (1995) Externalizing learner models. In J. Greer (Ed.) *Artificial Intelligence in Education: Proceedings of AI-ED 95*, Charlottesville, VA: AACE, 509-516.

Palthepu, S., Greer, J.E., McCalla, G.I. (1991) Learning by Teaching. In L.Birnbaum (Ed) *Proceedings of the International Conference on the Learning Sciences*, 357-363.

Pressley, M., Wood E., et al. (1992) Encouraging mindful use of prior knowledge: Attempting to construct explanatory answers facilitate learning, *Educational Psychologist*, **27**, 91-109.

Self, J. (1986) The application of machine learning to student modelling. *Instructional Science*, **14**, 327-388.

Shepherd, A. (1995) *Task Analysis*. Monk, A.F., Gilbert, G.N. Editors, *Perspectives on HCI: Diverse Approaches*, London: Academic Press, 145-174.

Slavin, R.E. (1990) *Cooperative Learning: Theory, Research and Practice.* Prentice Hall, Englewood Cliffs, NJ: Prentice Hall.

Vassileva, J. (1996) A task-centered approach for user modeling in a hypermedia office documentation system. *User Modeling and User-Adapted Interaction* **6**: 185-223.

Vassileva J. (1997) Goal-Based Autonomous Social Agents Supporting Adaptation and Teaching in a Distributed Environment, to appear in *Proceedings of ITS'98*, San Antonio, Texas, 15.08.-20.08.1998.

Virtual-U (1997) Simon Fraser University, Virtual-U Research Project [On-line] Available: http://virtual-u.cs.sfu.ca/vuweb/

Wenger, E. (1996) Communities of Practice: The Social Nature of Learning*. Healthcare Forum Journal*, July/August, 20-26.