Lessons Learned in Deploying a Multi-Agent Learning Support System: The I-Help Experience

Jim Greer, Gordon McCalla, Julita Vassileva, Ralph Deters, Susan Bull, Lori Kettel
ARIES Laboratory, Department of Computer Science, University of Saskatchewan, Canada.

Abstract. In this paper we look at the lessons learned from several large-scale real world deployments of the I-Help agent-based peer-help learning support system. These lessons divide into two main categories: software engineering lessons and usage lessons. In the deployments of I-Help to date we have learned a number of important things about the technology needed to support widespread use of a distributed learning support system. In particular accessibility, dependability, and scalability are critical needs. We have also learned a number of things about how, why, and even whether students will use a system like I-Help. There are technical and social dimensions to the usage issue. The paper briefly overviews I-Help, and then describes the various deployments. The software engineering and usage lessons are then elaborated, drawing on data gathered by I-Help itself during its various deployments and on questionnaires handed out to student users at the end of two of the deployments. These lessons are, we believe, useful not just in the I-Help context, but for any AIED researchers who plan to deploy a complex system in a real world for a large number of users.

1. Introduction

I-Help is a peer-help system designed to assist learners as they engage in authentic problem-solving activities. It works by locating resources (both online and human) that are particularised to a learner's help request. The I-Help project has been ongoing for a number of years, with descriptions of various aspects appearing in the research literature. The research has explored a number of interesting AIED research issues, especially in the areas of learner modelling and agent technology. In the last few years we have moved beyond research prototypes and have begun to deploy various versions of I-Help in large-scale experiments involving hundreds and sometimes thousands of learners. This has led to a whole new set of challenges and lessons learned. The focus of this paper is on these large-scale deployments and what we have learned from them.

While on the surface I-Help resembles a simple environment for sharing messages in public and private discussion areas with the help of a personal agent, underlying I-Help there is a significant and complex system. There are many personal agents that communicate with each other and with application agents of various sorts; there are learner models that are spread across the many agents in the system; and there are inference mechanisms to process the learner models to locate appropriate helpers. It is a huge effort to build such a complex system and at the same time make it robust, scalable, useable, and usefully intelligent and adaptive to individual learning needs. The first part of this paper explores the software engineering lessons that we have learned through several deployments of I-Help. The second part explores the lessons we have learned from these deployments about how, why, and whether students use the system. Drawing on performance data and post-hoc questionnaires we explore students' actual usage of the system and draw some preliminary conclusions about student use of I-Help. First, however, we introduce I-Help and describe the various deployments that we have carried out.

2. The I-Help System

I-Help has two components: public discussion (I-Help Pub) and private discussion (I-Help 1-on-1):

> *Public Discussion:* In I-Help Pub, learners can post questions, comments and responses to forums. These postings are shared with their peers. Forums are clustered into groups and group memberships. A person who is a member of a group can access the forums created for that group. I-Help Pub is used asynchronously.
>
> *Private Discussion:* The second I-Help component supports one-on-one private discussions (or help dialogues) between a learner – the helpee, and a single peer (or expert) – the helper. These dialogues may be synchronous or asynchronous. The following illustrates the sequence of events for a help request in I-Help 1-on-1: 1. A learner contacts their personal agent to issue a help request; 2. The learner's agent negotiates with the agents of other learners, to locate potential helpers; 3. The top N matches are notified that there is a help request waiting; 4. The first of the contacted helpers to accept the request starts a one-on-one interaction with the helpee. Requests to other potential helpers are cancelled; 5. Upon completion of the interaction, each learner receives a brief evaluation form through which they evaluate their partner, for student modelling purposes.

Multiple fragmented student models underlie the 1-on-1 system [1]. Each person "owns" a personal agent, their representative in the system, and this personal agent keeps a model of its "owner" as a source of information as it acts on the owner's behalf. These models are used by personal agents when negotiating help sessions with other users in order to determine the best helpee-helper matches [2]. User model information is obtained from the learner (through stated availability and self-assessment of knowledge of different topics); from the short peer evaluations; from a determination of whether or not the student is currently or frequently online; and from I-Help's observations of student participation in both the public and private discussions. The public and private discussions may be used together, or the two components may be used independently. Whichever is used, the obvious educational benefit to students is that those requiring help receive assistance at the time they need it. Furthermore, peers providing help should also benefit from the reflection necessary to formulate an acceptable explanation.

3. I-Help Deployments

We discuss three deployments of I-Help in classes at the University of Saskatchewan: 1. Sept.-Dec. 1999; 2. Jan.-Apr. 2000; 3. Sept.-Dec. 2000. In the first two, I-Help Pub and I-Help 1-on-1 were separate systems. They were integrated in deployment 3. Given their history as distinct sub-systems, we discuss I-Help Pub and 1-on-1 deployments separately below.

Deployments 1 and 2 of I-Help Pub allowed students to post questions and answers in threaded forums, having a structured, organised environment as a benefit to the learner. Deployment 1 had around 600 users. Deployment 2 was available to around 1000 users, but was actually used by about 750. Since email notification reminders to visit forums can increase usage, this was introduced for deployment 3. However, rather than emailing messages with reference to all new postings [e.g. 3], deployment 3 allowed notifications in reaction to *postings of interest* (users can request email notification of new postings in a particular forum, by a particular author, on a particular topic, and responses to a particular posting), to increase the utility of notifications. The other major innovations for deployment 3 were addition of: multiple views (users can create their own sets of forums, each view forming a single perspective through which to access forums); a search facility (searches can be performed according to topic, keywords or author); choice of English or French interface. Deployment 3 was available to 1600 students – all undergraduate courses in the Department of Computer Science, and to 100 students in two courses in Law, also at the University of Saskatchewan.

Turning to the private discussion component, deployment 1 of I-Help 1-on-1 used a synchronous chat environment. At that time, I-Help sought the single best helper, according to their knowledge of the topic. Knowledge was organised in detailed concept maps. The system was able to support about 50 personal agents, and was offered on a voluntary basis to 100 students, but there was very little usage. In deployment 2, synchronous-asynchronous messaging replaced the chat because the previous version was dependent on the selected helper being online at the time, and willing to engage in the help session. For the same reason, I-Help located the top five potential helpers to increase the likelihood of a quick response. Simple topic labels replaced the concept maps, because students did not want to maintain such a detailed learner model. In addition to knowledge level – helpfulness (as evaluated by previous helpees) and eagerness (online activity) were modelled, and this information was used alongside knowledge level in matching partners. Learners could also create a 'friends' list – people from whom they would particularly like to receive help, and to whom they would offer a discount in the event that they required help (I-Help agents and students are motivated to interact through a virtual currency – see Section 5). Users could similarly construct 'banned' lists – people with whom they did not wish to interact. Topics could also be banned. The number of personal agents that could be supported was scaled up to about 200. In deployment 2 I-Help 1-on-1 was offered to 322 first year computer science students for almost three weeks. Of these, 76 individuals registered to use the system. Among these, some used the 1-on-1 facility extensively; others used it rarely. There were 86 help requests in total over this three week period. Of those who were registered for both I-Help 1-on-1 and I-Help Pub at that time, 31% used the 1-on-1 facility only; 38% used I-Help Pub only; and 31% used both.

Major extensions were produced to the 1-on-1 system for the third deployment. As stated above, I-Help 1-on-1 and Pub were fully integrated for the first time. Eagerness, helpfulness and knowledge level were still criteria for matching in I-Help 1-on-1; however activity in I-Help Pub (number of postings read, replies made, etc.) now also contributed to the eagerness measure. The friends list had two sections – friends who receive a discount, and preferred helpers who receive a premium. The banned list was similarly divided – users could ban individuals as helper, helpee or both. In addition to the previous attributes, learners were able to provide a greater range of information to their agent, for student modelling – they could indicate how frequently they were willing to be contacted as helper; the maximum number of sessions in which they were prepared to be involved at one time; the importance of earning currency; and their ability to help. (The latter was used alongside peer evaluations of helpfulness.) For the role of helpee, the learner could indicate the relative importance of the following in a helper: knowledge level, helpfulness, speed of response, cognitive style and currency. These attributes were then weighted appropriately before the initiation of agent negotiations. Deployment 3 had over 400 agents. Due to this technology limit, the fully integrated I-Help system (with 1-on-1 and Pub) was made available to 326 students in 2 courses. As discussed further in section 5, there was very little usage by the students in one course of either component of I-Help, while usage in the other course was focussed mainly on I-Help Pub.

4. I-Help: Software Engineering Lessons

In this section we discuss some of the architectural and software engineering issues that have arisen as one deployment of I-Help has led to the next. We start with design requirements for I-Help. We then provide an historical overview at the technology level of the various versions of I-Help, showing how technological challenges have led to interesting solutions as I-Help has become ever more sophisticated. We conclude the section with a brief overview of some of the main software engineering lessons that we have learned.

Through its various versions I-Help has had three basic requirements: to be accessible, dependable and scalable. To avoid lack of use due to the accessibility problem sometimes experienced early in a project [e.g. 4], I-Help had to be widely available. Since it is required to operate in a highly heterogeneous environment, the best solution to the accessibility problem was to make I-Help available from a simple web-browser. The main http-clients targeted have been Netscape and Internet Explorer. Dependability is the second requirement. It has been crucial to ensure that the services offered to students are available, reliable, secure and safe, and that the system does not crash. The third requirement is that I-Help is able to scale up to allow more students to use it in a wider variety of contexts.

Even before the large-scale deployments discussed in section 3, there were several "proof of concept" prototypes of both I-Help Pub and I-Help 1-on-1. Early I-Help Pub prototypes used a public-domain database, ODBC and Perl-cgi scripts. Every page was generated by the server and almost every click required a screen refresh. The early tests with users resulted in such slow performance that they would not use the system. To achieve scalability and reasonable system performance, it became clear that a commercial database with direct web support was required. After several failed attempts to build a reliable Oracle-based application (due to the steep learning curve associated with Oracle application development), finally a stable and scalable I-Help Pub was built. This allowed deployment 1 to proceed.

The first I-Help 1-on-1 "proof of concept" prototype took a single-process server approach. It was written in Java (jdk1.1) and designed to run on a single PC. The application consisted of three modules: a simple communication module (ComServer), an agent host and a module to handle the database connection issues. The agents used in this implementation were simple Java threads that reacted to incoming messages. Small applets embedded in the page ensured a connection of the clients with the application. While all tests indicated a stable system, the first real usage ended in disaster. The sudden load caused by simultaneous login of over 60 users within a minute, led to a temporary high demand of processor power by the DB-Connection module. This meant that the agents had too little power, which led to slow creation of web pages. The reaction of the students to the decreased performance was a series of logoff-login commands, which lead to an extremely high load, which, in turn, resulted in total collapse of the application. With this first disappointing experience in mind the students refused to work with improved versions that year. We clearly had to do better if we were to go beyond a proof of concept prototype.

Thus, the version of the I-Help 1-on-1 architecture in deployment 1 attempted to overcome the problems of resource conflicts by using of RMI to distribute the server-side application. Each module became an independent process. In addition more complex agents that were able to communicate via KQML messages were introduced. These contained simple goal-queues and rudimentary planners. Further, the agents were enabled to observe the current load and plan their activities accordingly. Using this approach it was discovered that the use of applets led to serious problems (because of different Java versions supported by different browsers and hardware platforms). In addition it turned out that memory leaks (which do happen in Java!) led to crashes of the agent host. Monitoring the system and restarting it periodically before memory consumption reached critical levels ensured a minimal degree of stability. Unfortunately, usage of the system peaked on weekends before assignment-deadlines, which resulted several times in crashes at the time of greatest need. The students reacted to this instability by avoiding the tool.

The next implementation of the I-Help 1-on-1 architecture (which underpinned both deployments 2 and 3) represented a complete re-implementation of all parts. CORBA was adopted as an object sharing protocol, since it promised the best standard and the easiest way to ensure a scalable system. This version of the system consisted of a database connection and servlet engine for communication, as well as an agent for each user and a user host. The servlets ensured the connection of the clients with the other parts of the implementation and replaced the ComServer. In addition a user host was introduced that was responsible for handling all user data and also served as a cache for user specific web pages. Each module was implemented in a way that one main process (master) controlled various sub-processes. This technique ensured scalability by having several agent hosts and database connection processes. By spreading the processes over several machines, resource conflicts were avoided. This was the first stable version, which was able to serve up to 400 users.

Looking at overall software engineering lessons learned in the development of the various I-Help prototypes, one important decision was to use a database for most system information, an idea explored first in I-Help Pub. This decision has led to enhanced dependability and robustness. It is also easy to add new information and to find out information for a variety of purposes beyond peer matching (for example for our empirical studies). However, ORACLE has a very steep learning curve, and since it is a proprietary product, the portability of I-Help is restricted.

Another decision that stands out is to embed I-Help in an agent architecture, ideal for scalability and many other things. Off-the-shelf agent solutions were explored but most solutions were too limited, involving one process per agent, thus making scalability to thousands of agents an impossible goal. We therefore created our own multi-agent architecture named MAGALE [5], and this has proven to be critical to our success in getting 400 distinct personal and application agents working at the same time. In fact, the MAGALE architecture is an important ingredient to our future plans for this system. As we incorporate more and more I-Help functionality into the multi-agent paradigm, it becomes easier to modify a particular agent's capability and watch its effects on the system.

There is a down side to agents, however. The nature of emergent behaviour resulting from large numbers of interacting, semi-autonomous agents means that any notion of "correct" behaviour is very difficult to define. This suggests that there may be no way to predict whether a system will scale up without building it first. In fact even after it has been built and tested with simulated workloads, it is sometimes hard to predict the kind of workload that real users might apply. Further, simulated workloads that represent realistic situations with multi-user distributed systems are themselves very time-consuming and difficult to build. Often the deployment itself is the first real load test, so on the first day, when hundreds or thousands of students simultaneously log on, there is a real risk of an unpleasant surprise (the sad story of many "dot coms" whose servers failed to handle the load on day one of operation).

Another software engineering lesson learned in this project is that a system in constant evolution must be carefully managed during major deployments. Change management and version control are important issues. There is a great temptation to apply partially tested hot-fixes to code in the running environment. This has caused embarrassment to our developers on many occasions and caused confusion to our users when new features (or new bugs) or subtle changes began to appear without adequate explanation. One of the goals of experimental work with deployed systems is to compare functionality by offering different versions to different sub-groups of users. For example, two different agent negotiation algorithms were being used in deployment 3 of I-Help 1-on-1. The difference in behaviour between the two algorithms would be imperceptible to users, but would provide different candidate helpers for a given situation. Adding this kind of new functionality is relatively simple if the system is well designed. Clearly, version management is crucial in all of these situations. An important lesson learned in this area was to obtain traces of user behaviour and snapshots of learner model states over time so that post-hoc off-line experiments could be run to simulate real effects.

5. I-Help: Usage Lessons

During the various deployments of I-Help, the main goals were to determine whether or not: 1. the system helps in supporting student learning; 2. it stimulates more and better learning interactions among the students; 3. people learn through helping/explaining to other people. While the final proof that we achieved these goals requires data from many more deployments, sufficient data has been obtained to support these hypotheses and to reveal interesting insights on educational and social issues. Various kinds of data have been collected. In all deployments trace data has been collected by I-Help as learners interact. This data has become increasingly fine-grained from deployment to deployment as we have traced the additional functionality. In addition we distributed questionnaires after deployments 2 and 3. In deployment 2, we surveyed only the 76 students who registered for I-Help 1-on-1, receiving 64 responses. As stated previously, the 1-on-1 registrants were fairly evenly split between primarily using I-Help 1-on-1, I-Help Pub and both. (However, 86% felt that the availability of both components was useful – despite the lack of integration at this stage.) In deployment 3 we surveyed some of the first and third year courses to obtain opinions from students at different levels, and from courses with different usage patterns. Of our 538 responses, 308 came from students who stated they had sometimes, frequently or very frequently used I-Help (others used it only rarely (141) or never (89)). The analysis below is based on the trace data in the three deployments to date, as well as responses to the questionnaires collected in deployments 2 and 3.

Table 1: I-Help Pub usage, deployment 3

Course	total learners	total threads	total replies	total reads	threads by learners	replies by learners	reads by learners
CS 100	343	257	318	23601	173 67%	117 37%	22108 94%
CS 111	348	796	1306	158112	762 96%	837 64%	151789 96%
CS 116	251	28	27	3402	24 86%	18 67%	3071 90%
CS 330	75	162	263	21809	149 92%	189 72%	20511 94%
CS 370	135	260	147	17043	65 25%	61 41%	14277 84%

Table 1 shows I-Help Pub usage in the selected courses in deployment 3, taken from trace data. CS100, CS111 and CS116 are first year computer science courses, but have differences in focus and audience (and, as it turned out, differences in their use of I-Help): CS100 is a service course with an eclectic audience and only some technical depth; CS111 is a first year course for CS majors with an emphasis on programming; CS116 is a first course in programming but offered exclusively to second year Engineering students. CS330 and CS370 are key third year courses for CS majors. CS330 has heavy technical content; CS 370 focuses less on programming. The columns in Table 1 show the number of new postings (threads) in all forums during deployment 3; the number of replies; and the number of times any posting or reply was read. Since many of the threads and replies were made by instructors, teaching assistants and paid helpers, we have in the last three columns extracted the amount of student usage alone.

We believe that level of usage can be considered as an important evaluation criterion: if the system brings value to the students, it will be used; if it is not valuable, students will abandon it. However, timing of introduction is also important – many learners offered I-Help 1-on-1 in deployment 2 stated that they would have used it, or used it more, had it been available from the start of term. Implementation delays also resulted in late introduction of I-Help 1-on-1 in deployment 3 in CS116 and CS330, the two courses in which it was available. Avoidance in these courses was high. We observed drastically different levels and patterns of usage of I-Help Pub in different courses, despite its availability from the outset of all courses. In some it was used extensively, in some it was barely used at all. For example in deployment 3, I-Help Pub was well used in CS100, CS111, CS330, and CS370, but rarely in CS116, as illustrated in the access statistics in Table 1.

Usage patterns across courses were different. CS111, CS116 and CS330 comprised mostly learner questions with mainly peer responses, sometimes expert replies. The higher proportion of expert contributions in CS100 and CS370 are explained as follows. In CS100 the instructors used I-Help extensively for giving information, making announcements, offering hints about assignments, etc. In CS370 it was used heavily by instructors for providing feedback on assignments, and the creation of new threads in an attempt to stimulate discussion. Neither CS100 nor CS370 contain such technical content where students run into impasses. These classes dwell less on programming than do CS111, CS116 and CS330. It appears from the data analysed to date, that technical focus is a greater predictor of learner participation than is course level. In our computer science settings this is probably because I-Help Pub was promoted as a help facility, rather than a course collaboration tool. (Some of the planned future uses have a more collaborative focus.) In general, in courses with lower levels of use, usage increased before assessment due dates. In courses with higher usage, this tended to be more consistent across time. Other issues, including the low usage level in CS116, are discussed below.

It is hard to find direct evidence for the effect of the system on students' learning. Correlating grades with I-Help usage is one measure, and an analysis of deployment 2 data suggests there is a correlation between performance in the course and use of I-Help [6]. Further, prestige measures based on frequently read postings suggest that the most widely read Pub comments have been posted by the highest achievers (although not all high achievers post). However, it is not clear whether the higher grade comes as a result of the high usage of I-Help, or it is pre-condition for the high usage.

With reference to I-Help Pub, most students in deployment 3 responded (on a 5 point scale) that reading postings helped their learning; most found answers received useful; many found the act of writing their question helped them solve the problem themselves; many found that answering other people's questions helped in their own learning. The results in Table 2 are from the questionnaires of students who logged on sometimes, frequently or very frequently.

Table 2: I-Help Pub questionnaire results, deployment 3

	very frequently	frequently	sometimes	rarely	never
reading postings helped my learning	21 10%	52 24%	98 46%	35 17%	7 3%
answers received were useful	18 12%	58 40%	51 35%	14 10%	4 3%
writing out question helped me solve problem myself	7 5%	21 14%	70 48%	40 28%	7 5%
answering other people's questions helped my learning	2 2%	20 22%	41 45%	20 22%	8 9%

Other reasons for using I-Help, as expressed in deployment 3 responses to an open ended question, included the ability to access useful information; accessibility 24 hours a day; usefulness as a place to find hints for solving problems; the many perspectives provided in the range of answers to posted comments; the ability to confirm that a student is on the right track; the chance to compare one's progress to that of peers; the confidence boost resulting from seeing that others have similar problems. In contrast, deployment 2 questionnaires indicated different reasons for using I-Help 1-on-1 – mainly the greater depth of interaction that can occur in a private dialogue.

Table 1 also illustrates the utility of browsing for some students who were not active posters. Questionnaire results indicate that of those who logged in at least sometimes (308), 239 (78%) never posted a question. Of these 239, only 11 (0.05%) ever answered a question. It can be inferred that the remainder were reading postings because they found them useful. (This also indicates some overlap between those who were answering questions and those who posted them.)

Since the deployments took place with different groups of users, and used different versions of the environment, it is important to analyse the complex factors that contributed to higher level of usage in some courses versus others. There were within-group differences, as commonly reported [e.g. 7], but our widespread deployments also enabled identification of three groups of external factors that played a major role in usage levels. First of all, there are technical factors, concerning the stability of the system and the interface-organization. Second, there is a “knowledge investment” factor, which accounts for the amount of initial knowledge and help provided by the instructor and paid tutors in the form of posting hints, suggestions, additional information and answering student help requests until students develop trust in the system usefulness and learn how to use it. Finally, there is a social factor that determines to a large extent the patterns of usage and generally, the success of such a “social” tool. Below we discuss each of these three groups of factors in more detail, drawing lessons from all three deployments.

Technical

Technical factors have a large impact on I-Help usage. One of the reasons for the relatively low level of usage of I-Help 1-on-1 in deployment 1 was the slow response time of the system, especially off campus, due to slow network connections during this period. It must be pointed out that the slow response was due to reasons independent of the system (the local phone company was upgrading the network connection to campus). The coincidence of this maintenance with the introduction of the system to the course was unfortunate. Many students tried to log into the system, after endless waiting tried to log-in again, and when this failed too, they never tried using the system again. The speed of connection is important, and so is the type / power of computer used. For example, the CS116 students using I-Help (integrated version) in deployment 3 usually access the system from a lab where the software required for their course assignments is installed, rather than from their own computers. Unfortunately, the lab is using very old and slow computers (Pentium I). Running the programming environment required for the course (Visual C++) simultaneously with a browser consumes the processor power entirely, which slows down the performance in both I-Help and the programming environment. These two examples show the critical importance of such "low level" technical factors for the usage of the system. There are sometimes unexpected difficulties in implementing complex distributed multi-agent systems, due to very basic "low-level" problems, completely unrelated to the proposed technology.

An important factor influencing usage is interface design, which made interaction with the personal agent somewhat cumbersome in the first two deployments. In those deployments, different interfaces for I-Help Pub and 1-on-1 were used. A seamless integration of the two interfaces was difficult, but necessary and it was achieved for deployment 3. Approximately 70% of development effort on deployment 3 was required for user interface development.

There is an obvious trade-off between the rich functionality that is provided to students to, for example, create personal views, subscribe to certain forums, activate a notification mechanism about a reply to a particular posting or person, search postings etc., in I-Help Pub, and the need for a simple interface that is seamless and easy to learn. Questionnaire responses from deployment 3 indicate that 56 users created new views, 48 (86%) of whom found the views useful. 117 people set notifications, of which 89 (76%) found them useful. 130 tried the search facility; 115 (88%) found it helpful. Some students in deployment 3 (66 in total) had prior experience with WebCT or other forum tools. Of these, 29 preferred I-Help, 21 preferred other tools, 16 were indifferent. We observed that some students with more limited use of I-Help preferred other discussion forums (like those provided by WebCT or newsgroups) for their simplicity. However, students who participated in many active forums appreciated the value added by I-Help's functionality.

Knowledge Investment

As also found in other environments [e.g. 8], knowledge investment by the "authorities" in a course seems to be an important influence on usage. In all three deployments to date, the participation of the instructor and of paid tutors (who initially contribute learning materials not available elsewhere, answer questions promptly, and make themselves available) is critical at the outset of a course to stimulate usage of the system. The relatively minimal usage of I-Help in CS116 in Table 1 can in part be explained by a lack of such investment, as can the lack of use of I-Help 1-on-1 which was available to CS116 and CS330 in deployment 3. (While course authorities were active in I-Help Pub in CS330, they did not participate in I-Help 1-on-1.) It appears likely also that had the experts been less active in I-Help Pub in CS370 in particular, usage levels would have quickly dropped. The greater usage in the other courses was encouraged by serious knowledge investment by the course authorities. However, we also found in general that after I-Help begins to be used more extensively, a culture of usage develops among the learners in a course. Moreover, such initial knowledge investment by the paid helpers pays off in more direct ways: it greatly reduces the number of email requests for help to course authorities, since usually these requests are about the same problems, and providing answers in a public forum saves the effort to answer each one individually. An instructor who used I-Help Pub extensively in a large multi-section course (CS111 in deployments 2 and 3) claimed to have reduced email interactions with individual students by 90% due to I-Help. The instructor time to deal with student problems via I-Help was less than half that of prior offerings of the course without I-Help. A similar result was obtained in a 700-person distance learning course in Taiwan [9] where the number of paid teaching assistant hours has been reduced by 2/3 due to the more efficient use of tutor time with their system.

Social

A number of social factors affect I-Help usage. As has been found elsewhere [e.g. 10], choice of group had a strong influence on the level of use. Although there are uses for I-Help in some small group interaction settings [11], often smaller or more cohesive groups do not need the system. The first deployment of I-Help 1-on-1 was with 3rd year students who knew each other well, had established multiple ways of interacting in course and in the labs, and hence did not find any need to login to the system. The reasons for this choice were purely pragmatic: time until the beginning of term was short and implementation for this course required the least adaptation effort, as the domain representation and student modelling were already developed. A similar effect appeared in CS116 with the integrated I-Help, with a large group (3 parallel sections) of second year Engineering students. Due to the culture of the College of Engineering, involving much group work and extra-curricular activities, students knew each other well and had established knowledge networks. They shared laboratory space so there was ready access to face-to-face help. 68% of those who stated in the deployment 3 questionnaires that they had never logged on, were from this group. (Other reasons for low use were technical – poor computers in their labs, as mentioned before; and the lack of knowledge investment by course authorities).

Having knowledge-level differences within a group also encourages I-Help usage. If all the students are at approximately the same level of knowledge, it is less likely that the selection of competent helpers for I-Help 1-on-1 will be effective. Such uniformity will also result in a passive audience for I-Help Pub, with students mainly reading postings contributed by the teacher or tutors and contributing little. This occurred to some extent in the service course (CS100), where only 37% of replies came from students. In deployment 3 some students were able to access I-Help forums for several different courses at the same time. With deployment at this scale, there is the potential for cross-fertilisation among courses, either in I-Help Pub when students in more than one course adapt responses read in one course for use in another, or in I-Help 1-on-1 where potential helpers could be selected from students in other courses. We hope to explore these issues in future deployments.

Motivation is another social issue of importance. Our effort to motivate students to offer help led to the introduction of an I-Help economy [2] underlying I-Help 1-on-1 in all three deployments to date. The main idea is that those who request help have to pay (in I-Help credit units, a virtual currency) and those who give help get paid for the effort. A special negotiation mechanism [12] among the agents has been incorporated (in deployment 3) to facilitate the selling and buying of help. The I-Help economy is intended to create a dynamic help market, which is important not only for encouraging a reasonable level of help requests and help responses, but also for load-balancing among helpers.

Has the economy worked? In deployments 1 and 3 the amount of use of I-Help 1-on-1 was minimal, suggesting the economy was not particularly motivating. I-Help 1-on-1 was more extensively used in deployment 2, but it is not clear that the economy was the motivating factor. Respondents to the questionnaire administered after deployment 2 were evenly split as to whether they found the virtual currency motivating. Some mentioned that it would be good to be able to exchange the accumulated help-currency for marks towards their final grade in the course. Two people were particularly negative, stating that the currency was stupid! One problem may be that the currency exchange of I-Help credit units into things of value in the real world is not favourable (minimal prizes have been given for top helpers). Another problem may be that rewarding students solely on the level of their bank account does not take into account the quality of the help. It might be important also to take peer evaluations of helpfulness into account, and to see whether users have banned helpers. For example, in deployment 2, one student who was involved in many help sessions in the role of helper (17), left 5 of his helpees with an unanswered question – i.e. he abandoned these helpees during an ongoing discussion. Finally, perhaps the currency has to be converted into other things than material goods. Several students revealed their main motivation for posting answers on the public discussion forum to be "glory", that through I-Help Pub they became recognised as "authorities" among their peers. Some students mentioned that they hoped by posting on I-Help Pub to attract the attention of the instructor, another form of recognition. Perhaps I-Help 1-on-1 needs to map I-Help currency onto fame and social status, not prizes. In fact, it seems to be generally recognised that social recognition is an efficient reward system also in many newsgroups and on the internet for the developers of free software [13]. Though our I-Help data is inconclusive, we believe that some form of reward is useful to stimulate student participation. The crucial question is the choice of the real world equivalent. The reward should be based on the social values of the group.

We do not suggest that I-Help should necessarily be used by all students. It may not be the most effective method of obtaining help for some. For example, users registered in deployment 2 of I-Help 1-on-1 who did not make use of I-Help gave the following reasons as responses to an open-ended question: asked friends; preferred face-to-face interaction; preferred working in a small group; asked lab assistant; asked teacher; checked textbook/references; preferred solving own problems; gave help in person; never needed help. These are all valid reasons for not using I-Help 1-on-1. Similar reasons were given for not using I-Help Pub or 1-on-1 in deployment 3.

6. Conclusions

We are still analysing the data from deployment 3, and expect to find other interesting patterns, gleaned from both the quantitative and qualitative information available. We plan to continue to deploy complete versions of I-Help at an increasingly large scale and in an ever widening range of contexts, and to learn the real world lessons from this. We are also involved in new research to extend the capabilities of I-Help. Many graduate student and other research projects are investigating topics like fragmented learner modelling for distributed environments, social networking in peer help systems, the development of computational environments that must work with a massive number of agents, the extension of distributed environments to be available pervasively and ubiquitously, the modelling of affect as well as content, the intelligent formation of groups and their maintenance, the capture and use of cognitive style in selecting appropriate peers, the impact on privacy of systems like I-Help and how to constrain invasions of privacy, visualisation of learner models, etc.

The research model underlying the I-Help project is a good one, we believe. Long term research (often carried out by graduate students) into basic AIED and other scientific issues leads to new ideas and/or "proof of product" prototypes that shed light on important possible directions for I-Help. These ideas, then, are incorporated into new I-Help sub-systems and integrated into the latest about-to-be-deployed full I-Help system. The lessons learned from the large scale deployment of this system then feed back into the longer term research effort, providing useful data about what really works and what does not in the pragmatic brutality of the real world. The cycle continues, with the scientific, engineering, cognitive and social dimensions feeding off one another to provide greater insight.

Acknowledgements: We thank the Canadian TeleLearning Network of Centres of Excellence, whose funding has been critical to the development of I-Help. Also Canada's Natural Sciences and Engineering Research Council, whose funding of Greer, McCalla, and Vassileva has been essential, especially for basic research directions explored. Finally, funds provided by the University of Saskatchewan and the Canadian Foundation for Innovation have proven useful in funding the technological infrastructure underlying I-Help.

References

[1] McCalla, G., Vassileva, J., Greer, J. & Bull, S. Active Learner Modelling, in G. Gauthier, C. Frasson & K. VanLehn (eds), Intelligent Tutoring Systems, Springer-Verlag, Berlin Heidelberg, 2000, 53-62.

[2] Vassileva, J., Greer, J., McCalla, G., Deters, R., Zapata, D., Mudgal, C. & Grant, S. A Multi-Agent Design of a Peer-Help Environment, in S. Lajoie & M. Vivet (eds), Artificial Intelligence in Education, IOS Press, Amsterdam, 1999, 38-45.

[3] Akers, R. Web Discussion Forums in Teaching and Learning, Horizon: horizon.unc.edu/projects/monograph/CD/Technological_Tools/akers.asp, accessed Feb. 2001.

[4] Hmelo, C.E., Guzdial, M. & Turns, J. Computer Support for Collaborative Learning: Learning to Support Student Engagement, Journal of Interactive Learning Research 9(2), 1998, 107-129.

[5] Vassileva, J. Goal-based Autonomous Social Agents Supporting Adaptation and Teaching in a Distributed Environment, Proceedings of Third International Conference on Intelligent Tutoring Systems, San Antonio, Texas, 1998, 564-573.

[6] Bowes, J. Knowledge Management through Data Mining in Discussion Forums, MSc Thesis, Department of Computer Science, University of Saskatchewan, 2001.

[7] Woodruff, E., Brett, C., Macdonald, R. & Nason, R., Participation in Knowledge-Building Communities to Promote Teaching Competency in Mathematics, Presented at Annual Meeting of Canadian Society for Studies in Education, 1998, available at www.oise.utoronto.ca/~cbrett/csse98.

[8] Light, V., Nesbitt, E. & Light, P. CMC in HE, Education-line,1998, www.leeds.ac.uk/educol/documents/000000693.htm.

[9] Chen, G.D. Web Portfolios: Tools for Monitoring and Assessing Learning Process (invited talk), in S. Young, J. Greer, H. Maurer & Y.S. Chee (eds), Proceedings of ICCE/ICAI (Vol. 1), National Tsing Hua University of Taiwan, 2000, 73.

[10] Guzdial, M., Hmelo, C., Hübscher, R., Nagel, K., Newstetter, W., Puntambekar, S., Shabo, A., Turns, J. & Kolodner, J. Integrating and Guiding Collaboration: Lessons Learned in Computer-Supported Collaborative Learning Research at Georgia Tech, Proceedings of Computer Supported Collaborative Learning, 1997, 91-100.

[11] Bull, S. & Greer, J. Peer Help for Problem-Based Learning, in S. Young, J. Greer, H. Maurer & Y.S. Chee (eds), Proceedings of ICCE/ICAI (Vol. 2), National Tsing Hua University of Taiwan, 2000, 1007-1015.

[12] Mudgal, C. & Vassileva, J. Bilateral Negotiation with Incomplete and Uncertain Information, in Klusch & Kershberg (eds) Proceedings of Workshop on Cooperative Information Agents, Springer-Verlag, 2000, 107-118.

[13] Raymond, E. Homesteading the Noosphere, the Cathedral, and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, O'Reilly and Associates, 1999, available at www.tuxedo.org/~esr/writings.