Lessons from Deploying I-Help

Julita Vassileva, Ralph Deters,

University of Saskatchewan
57 Campus Drive
Saskatoon, S7N 5A9 Canada
{jiv, ralph }@cs.usask.ca

ABSTRACT

In this paper, we describe the multi-agent infrastructure underlying I-Help, an internet-based peer-help system that was deployed over the last two years with over 600 students at the University of Saskatchewan. The system contains a variety of learning resources, most prominently, public discussion forums, on-line materials, and a chat-tool for private discussion between peer-learners. In this paper we focus only on the private discussion tool (called also "I-Help 1-to-1"). We present some of the lessons learned in developing and deploying this part of the system.

1. INTRODUCTION

I-Help is an internet-based peer-help system designed to assist learners as they engage in problem-solving activities. It locates resources (both electronic materials and humans able to help) that are suited to a learner's help request. The I-Help project [6] has explored a number of interesting research issues, especially in the areas of learner modelling and agent technology. In the last two years we have deployed various versions of I-Help in large-scale experiments involving hundreds of learners. This has led to many challenges and lessons learned.

2. MULTI AGENT ARCHITECTURE

To illustrate the functionality of I-Help we will use an example scenario. Imagine that a student working on a programming assignment has a question. The personal agent of the student asks matchmaker agent to find another student who is currently on line and is competent on the topic of the question. The matchmaker maintains profiles of the knowledge and some other characteristics of all the users. The matchmaker creates an ordered list of the users who qualify and sends it to the personal agent of student who asked for help. The personal agent starts negotiation with the personal agents of the users from the list, trying to find one that would agree to help at a satisfactory price in I-Help credit units (ICUs), the virtual currency of the underlying I-Help economy. Once the negotiation process has succeeded, the agent of the potential helper notifies its user and asks her if she would be willing to help or not. If not, the personal agent has to negotiate with other agents from the list of suggested helpers. If the helper is willing to help, a communication channel is opened between the two users (a simple chat tool), and a help session starts. After one of the parties closes the chat window, an evaluation form pops up in which the student has to evaluate the other one. This information is used to update the knowledge profiles as well as some of the other characteristics of users maintained by the matchmaker agent.

I-Help is based on a multi-agent architecture [7], consisting of personal agents and application agents (see figure 1). These agents use a common ontology and communication language. Each agent has a model of the resources of the user or application it represents. Personal agents keep a model of the knowledge level of the learner about domain topics, as well as some individual features, like eagerness, helpfulness, class ranking [3]. Application agents keep model of the topics addressed by the instructional materials belonging to an application (e.g. an educational web-site). The agents use their resources to achieve the goals of their users, their own goals, and goals of other agents. Thus the agents are *goal-driven*. In their goal pursuit the agents can also use resources borrowed from other agents, i.e. they are *collaborative*. For this they have to negotiate. The agents communicate with each other and with matchmaker agents to search for appropriate help resources for their users, depending on the topic of the help-request. If an electronic resource is found (represented by application agents), the personal agent "borrows" the resource and presents it to the user in a browser. However, if a human helper is located, the agents negotiate the price for help, since human help involves inherent costs (time and effort) for the helper. The result of a successful negotiation is an agreement from a competent learner to help in exchange for a negotiated payment in virtual currency (actually the payment rate/min is negotiated). Help is negotiated entirely by the personal agents thus freeing the learner from the need to bargain. In this way the personal agents trade the help of their users on a virtual help market [2, 8]. In this way, we achieve a distributed (multi-user, multi-application) adaptive (self-organized) system that supports users in locating and using help resources (other users, applications, and information).

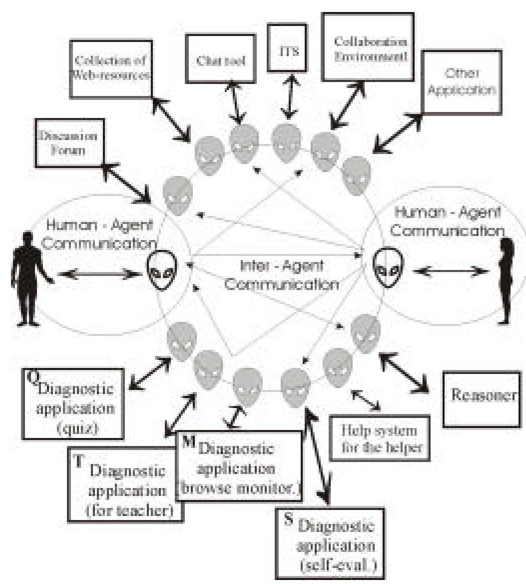

Figure 1**: The multi-agent Architecture of I-Help**

We explored off-the-shelf FIPA-conform agent frameworks, but they turned out to be too limited, involving one process per agent, thus making scalability to hundreds of agents an impossible goal. Thus, we created our own multi-agent infrastructure, and this has proven to be critical to our success in getting over 400 distinct personal and application agents working at the same time. CORBA is used as an object sharing protocol, since it is the standard and easiest way to ensure a scalable system. The infrastructure of the system consists of an agent for each user and a user host, a database connection and servlet engine for communication. The servlets ensure the connection of the clients with the other parts of the implementation. In addition a user host is introduced that is responsible for handling all user data and also served as a cache for user specific web pages. Each module is implemented in a way that one main process (master) controls various sub-processes e.g. there is one database connection main process, which controls several database connection sub-processes. This technique ensures scalability by having several agent hosts and database connection processes. By spreading the processes over several machines resource conflicts are avoided. In this way a stable multi-agent infrastructure was achieved, which was able to serve a large number of users (up to 400). When more than 400 users are given personal agents, however, the CORBA object-request broker can not support the load. This limits the scalability of the system.

We are currently are experimenting with the next version of the agent infrastructure that involves a fully distributed multi-processor implementation with automatic load balancing across many processors. New CORBA brokers and processes are spawned automatically as required on under-utilised processors. If one processor fails, the entire set of agents that it supported migrates onto a new processor without interruption. This implementation offers also a rule-based expert system shell on board each agent, permitting the agents to be "programmed" in more flexible ways. As we incorporate more and more functionality into the I-Help 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, autonomous agents means that any notion of "correct" behaviour is very difficult to define. This suggests there may be no way to predict whether or not a system will scale up without building it first. In fact even after it is built and tested with simulated workloads, it is sometimes hard to predict the kind of workload that real users might apply. Furthermore, simulated workloads that represent realistic situations with multi-user distributed systems are themselves very time-consuming and difficult to build. Often the deployment itself works as the first real load test, so on the first day, when hundreds of students simultaneously start logging on, there is a real risk of a bad surprise.

3. DEPLOYMENT LESSONS

We discuss three deployments of I-Help in classes at the University of Saskatchewan.

Deployment 1 of I-Help (Sept.-Dec. 1999) used a synchronous chat environment. At that time, I-Help sought the single best helper, according to their knowledge of the topic. The first deployment was available to 100 students, but there was very little usage. The reasons for this were technical; there were problems with the network speed, but there were also social reasons, as will be explained later in the paper.

In deployment 2 (Jan.-Apr. 2000), synchronous/asynchronous messaging replaced the chat, because the previous version was dependent on the selected helper being online, 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. In addition to knowledge level, the learner helpfulness (as evaluated by previous helpees) and eagerness (a number, calculated from the student's pattern of online activity) were also modelled, and this information was used 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. Users could similarly construct 'banned' lists – people with whom they did not wish to interact. In this way each personal agent has a model of the social relationships of its user. The price for help was calculated centrally by the matchmaker depending on the difficulty of the topic and the amount of knowledgeable users currently on-line. In deployment 2 I-Help was available to 322 first year computer science students for three weeks. Of these, 76 individuals registered to use the system. 62% of them actually used the peer-help search – some extensively; others rarely. There were 86 help requests in total over this three-week period.

In deployment 3 (Sept.-Dec. 2000), a negotiation mechanism [4] was incorporated in the personal agents. It allowed a dynamic calculation of the price for help depending on the priorities of the helper and helpee. The system was available during Term 1 (3 months) to 251 second year engineering students. Of these, only 2 used the systems few times a week, 13 weekly, 31 seldom, and 205 – very rarely. This usage involved only the discussion forums and the on-line resources. There were no requests for peer-help.

In the remainder of this section we discuss the factors and issues influencing the variable usage of I-Help.

3.1 Technical Factors

Technical factors had a large impact on I-Help usage. One of the reasons for the relatively low level of usage 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 long waiting tried to log-off and 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 students using I-Help in deployment 3 usually accessed the system from a lab where the software required for their course assignments (Visual C++) is installed, rather than from their own computers at home. Unfortunately, the lab is using old and slow computers (Pentium I). Running 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.

3.2 Social Factors

A number of social factors affect I-Help usage. The choice of group had a strong influence on the level of use. Often smaller or more cohesive groups do not need the system. The first deployment of I-Help 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 deployment 3, 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.

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 will be effective.

There might be also deeper reasons for the little usage of the peer-help facility. Other studies [8] show that many students lack the metacognitive skill to know when to ask for help and how to make most of the help available. This suggests the user of more pro-active and pedagogically intelligent agents that monitor the student behaviour and discover opportunities to offer help.

Motivation is another social issue of importance. Our effort to motivate students to offer help led to the introduction of the economy underlying I-Help. We 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 regulating the demand and supply of resources at different times (e.g. before an assignment deadline) and for load-balancing among helpers. 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.

Has the economy worked? In deployments 1 and 3 the amount of use of I-Help was minimal, suggesting the economy was not particularly motivating. I-Help 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. 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 them 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 of I-Help to be "glory", that they became recognised as "authorities" among their peers. Some students mentioned that they hoped by posting on the forum to attract the attention of the instructor, another form of recognition. Perhaps I-Help needs to map the currency onto fame and social prestige, as some adaptive web-sites (www.thevines.com, or www.slash.org) already do. 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 [5]. Though our 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.

4. CONCLUSIONS

In this paper we outline the multi-agent architecture of I-Help, a distributed Internet based system for peer-help and we focused on discussing the lessons learned from several deployments of the system. As these lessons show, developing a scalable, dependable and usable multi-agent system is a complex endeavor, which can fail in many different ways. However, as virtual learning environments span the web, and physical presence becomes expensive or impossible, we believe that environments such as I-Help will become a valuable communication tool to support learning. Exploring the social conditions for success of such environments is equally important as pushing the technical frontiers in scalable multi-agent systems infrastructures.

5. REFERENCES

[1] Aleven V. Koedinger, K. Limitations of Student Control: Do Students Know when they Need Help? ITS'2000, Springer, 292-303.

[2] Kostuik K, Vassileva J. (1999) Free Market Control for a Multi-Agent Based Peer Help Environment, Autonomous Agents '99 Workshop on Agents for Electronic Commerce and Managing the Internet-Enabled Supply Chain.

[3] McCalla, G., Vassileva, J., Greer, J. & Bull, S. Active Learner Modelling, ITS'2000, Springer, 2000, 53-62.

[4] Mudgal, C. & Vassileva, J. Bilateral Negotiation with Incomplete and Uncertain Information, CIA'2000, Springer, 107-118.

[5] 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.

[6] Vassileva, J., Greer, J., McCalla, G., Deters, R., Zapata, D., Mudgal, C. & Grant, S. A Multi-Agent Design of a Peer-Help Environment, Proc. AIED'2001, IOS Press, 38-45.

[7] Vassileva, J. Goal-based Autonomous Social Agents Supporting Adaptation and Teaching in a Distributed Environment, ITS'98, Springer, 564-573.

[8] Winter, M. (1999) The Role of Trust and Security Mechanisms in an Agent-Based Peer-Help Environment, Autonomous Agents '99, Workshop on Deception, Trust, and Fraud in Agent Societies, 139-149.