MULTI-AGENT MULTI-USER MODELLING IN I-HELP*

JULITA VASSILEVA, GORDON MCCALLA, JIM GREER
University of Saskatchewan, Computer Science Department, Saskatoon, S7N 5A9 Canada
E-mail: {jiv, mccalla, greer}@cs.usask.ca

Abstract
This paper describes the user modelling approach applied in I-Help; a distributed multi-agent based collaborative environment for peer help. There is a multitude of user modeling information in I-Help, developed by the various software agents populating the environment. These "user model fragments" have been created in a variety of specific contexts to help achieve various goals. They are inherently inconsistent with one another and reflect not only characteristics of the users, but also certain social relationships among them. The paper explores some of the implications of multi-agent user modelling in distributed environments.

Key words: distributed user modelling, decentralized, multi-agent systems, agent negotiation, just in time user modelling, modelling interpersonal relationships, help-desk, expert finding, evaluation

1. Introduction

It is still common parlance to speak of a "user model" and to mean by this a single global description of a user maintained by some application ("the system") to judge the user's level of experience, understanding of domain content, mastery of tasks, preferences, attitudes, or cognitive styles. In this paper we discuss user modelling issues in a distributed, multi-user, multi-agent environment. Our discussion is based on our experience with I-Help, a multi-user, multi-agent environment that has been in use for 2 years in Computer Science courses at the University of Saskatchewan (and in a few other courses as well). The current version consists of more than one thousand agents. I-Help is one of the very few deployed large-scale multi-agent systems (MAS), which by itself is an advance in the state of the art of this technology. In the course of two years of I-Help deployment we have accumulated a significant wealth of usage data, with over 3000 users during that time.

User modelling (UM) plays a crucial role in I-Help. Many agents hold user model information about any user. At any given moment of time there is no one consistent model of a user, but many "snapshots" taken by various agents, in different contexts, containing totally different information. Achieving an ultimate, consistent user model in this distributed world seems hardly feasible and in many cases not even useful (Kono et al., 1994).

This paper describes the architecture and deployment of I-Help. It also introduces a new framework for discussing the issues of multi-agent multi-user modelling and applies this framework in analyzing the various aspects of user modelling that take place in the system

---

*to appear in *User Modelling and User Adapted Interaction* (2002) E. Andre and A. Paiva (eds.) Special Issue on User Modelling and Intelligent Agents.

2. A Multi-Agent Multi-User System: I-Help

I-Help (Greer et al., 1998) provides seamless access for students to a variety of distributed help resources (human resources, like peer help and expert advice, as well as electronic resources, like threads in discussion forums, FAQ entries, and web-resources). The system evolved from a centralized peer-help system for workplace training, called PHelpS (Collins et al., 1997) into a large-scale multi-agent environment for university students that allows them to receive and give peer help, both synchronously and asynchronously. Both PHelpS and I-Help use a matchmaking services based on user models that find an appropriate peer-helper online who can help when a request is made.

In the remainder of this section we discuss I-Help in order to provide a context for the distributed user modelling framework introduced in section 3.

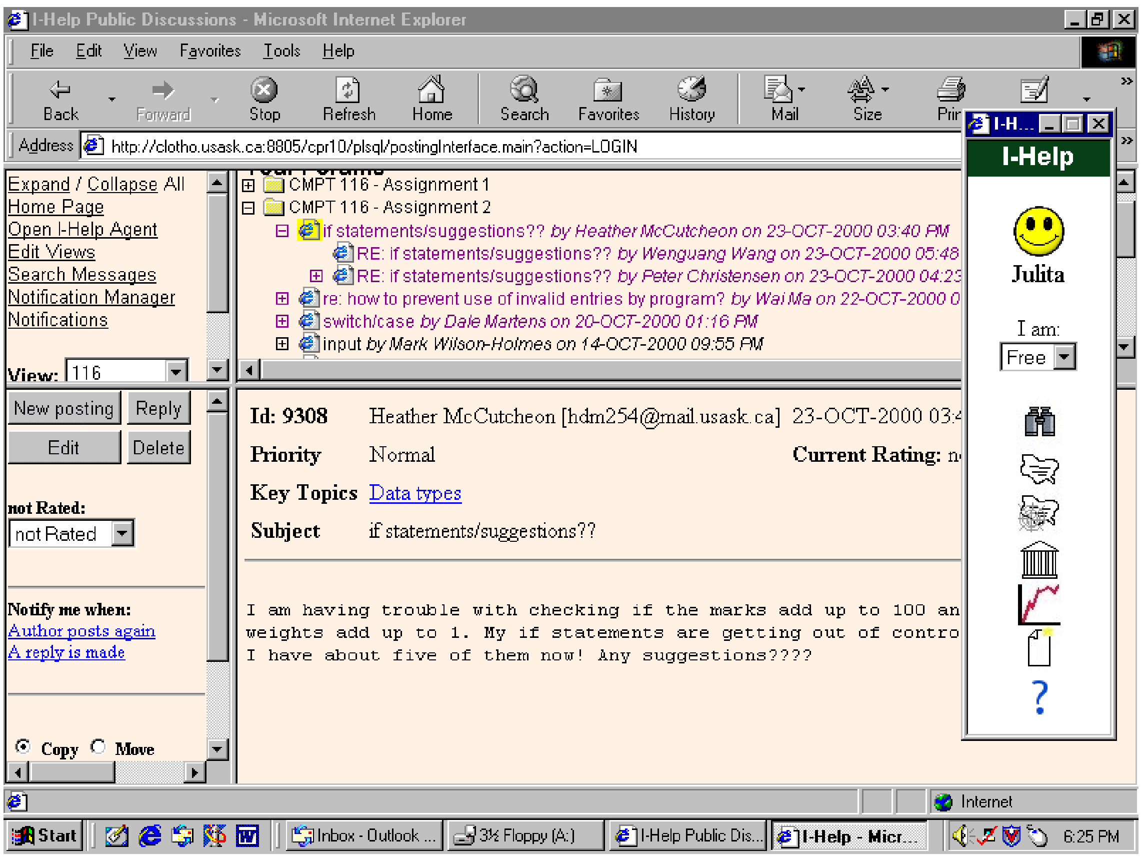

Figure 1: The personal agent (on the right) has found a relevant thread to the help request ("Assignment 2") in the public discussion forum. The four parts of the window comprise the interface of the public discussion forum. They allow viewing the postings in a hierarchical way (organized along the topic taxonomy), sending a new posting, replying to a posting, deleting postings, voting on the quality of postings, and automatic e-mail notifications.

2.1 FUNCTIONALITY: AN EXAMPLE SCENARIO

To illustrate the functionality of I-Help we will use an example scenario. Imagine that a student working on a programming assignment in a Computer Science course has a question. She delegates the task of finding help to her personal agent. The personal agent tries to find another agent (either application agent or another personal agent) that offers information resources related to the help request. These resources can be electronic resources, for example web-pages created by the instructor or other students (and represented by their application agents in the system), or threads / postings in the I-Help discussion forum (represented by discussion forum application agents). For example, Figure 1 shows the screen where the personal agent has found a relevant thread in the discussion forum and highlighted it for the learner.

The agent can also find "human help resources", i.e. students who are currently on line and competent in the topic of the question. The agents share a common taxonomy for indexing the information resources, based on the topics/concepts taught in the class, and particular events, like labs, assignments, and exams (see Figure 2). Usually the course instructor creates the taxonomy from the course outline when the system is configured for a given course. The taxonomy can be expanded later, if necessary. Locating the agents that possess information resources or represent users knowledgeable on certain topics is facilitated by matchmaker agents that maintain profiles of the knowledge and some other characteristics of users and applications.

Format of program	Functions	Debugging
Data types	definition of	Searching & sorting algorithms
Math operators, Expressions	parameters	Computational complexity
Type casting	void/return values	Objects and Classes
Function calls	scope of variables	Numerical Analysis Assignment 1
Input/Output	recursive functions	Assignment 2
Selection	Arrays	Assignment 3
Loops	single & multi-dimension arrays	Assignment 4
File Input/Output	parallel arrays	Mid-term exam questions
	records / structs in arrays	Final exam questions

Figure 2: An example of a topic/concept taxonomy (used in one of the 2000/2001 Term 1 deployments of I-Help in a course on introductory programming in C++ with 251 students)

Back to the scenario: if there is no appropriate electronic resource for the student's question, the matchmaker creates a ranked list of the users who are on line and who know something about the topic. The matchmaker sends this list to the personal agent of the student who asked for help. The personal agent starts negotiation with each of the personal agents of the potential helpers on 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. If not, the personal agent has to negotiate with other agents from the list of suggested helpers. If the student 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 terminates the chat, an evaluation form (specific for the matchmaker that recommended the helper) pops up allowing each student to evaluate the other student. This information is stored by the personal agents (i.e. each personal agent contains a model of the other user); it is also forwarded back to the matchmaker to update profiles of users in its database.

I-Help introduces negotiation and payment for help in order to: (1) prevent competent users from being overloaded with help-requests, and (2) motivate users to help, since especially in large multi-user environments, it cannot be expected that users will necessarily be intrinsically motivated to help other users. Thus, on an agent-level the I-Help economy helps in regulating the supply and demand of human help resources. An economic regulating mechanism is important in multi-user environments since otherwise they tend to get invaded by "harvesters" which cause degradation in the performance (Adar & Huberman, 2000). On a human level an economy is optional. It could help by providing a source of motivation for human users to participate and offer their time and advice to other users. For this it is necessary to introduce a real world equivalent for the virtual currency (e.g. bonus marks in the class). However, as some authors (Raymond, 1999) point out, such a "real-world" economy may not always be necessary, since users can also be motivated by reputation. While the questions about the economic, ethical, and security issues related to introducing a real–world economy in an educational system like I-Help are important and interesting (Kostuik & Vassileva, 1999; Winter, 1999), we will not focus on them further in this paper, since they are not directly related to user modelling.

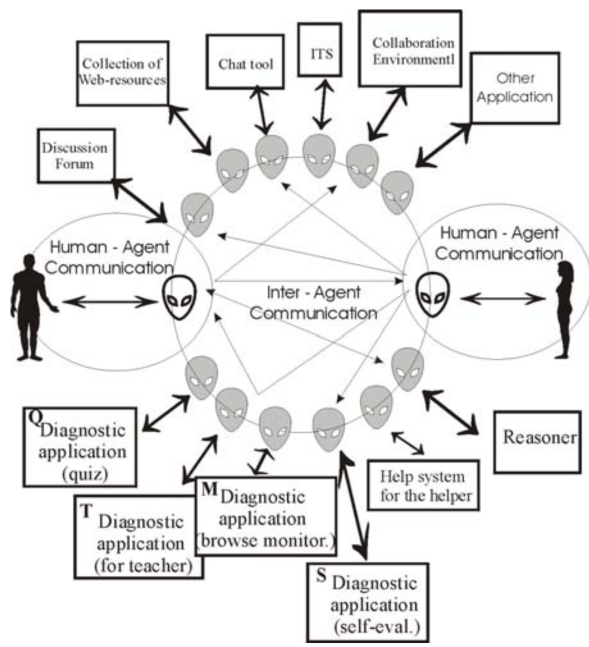

Figure 3: The multi-agent architecture of I-Help. The grey faces represent application agents, the white ones – personal agents. The applications are shown as rectangular boxes. The four boxes with capital letters in the bottom of the picture are different diagnostic applications whose purpose is to create and update user models in particular domain. Matchmaker agents are not shown on the picture, but several of them may exist – they are a special kind of application agents.

2.2 MULTI-AGENT ARCHITECTURE

The architecture of I-Help (Vassileva et al., 1999) is based on two types of agents (see Figure 3): personal agents (of human users) and application agents (of software applications). These agents use a common ontology and communication language. Each agent manages specific resources of the user or the application it represents, for example the knowledge resources of the user about certain tasks or concepts, or the instructional materials belonging to an application. Agents trade these resources when they need resources that they do not possess. For this they negotiate and establish long-term inter-agent relationships, some of which reflect relationships between human users. In this way, we achieve a complex multi-user, multi-application, adaptive, self-organising system that supports users in locating and using resources (other users, applications, and information).

Adaptation in I-Help is based on information held in the different agents distributed through the system. Some of the most important agents are briefly discussed below.

Personal agents (agents representing users), maintain partial user models containing certain basic, and in some cases very private, user characteristics. Examples of such characteristics are lists of the user's friends and foes, preferences about how the agent should negotiate on the user's behalf, taking into account the subjective importance to the user of certain resources like time (business) or money (greediness), the user's egoism or altruism. These characteristics are set explicitly by the user (see Figure 4). They reflect the way the user wishes to be perceived by the "world" through his/her personal agent; therefore, indirectly, they also represent a kind of model of the user. During negotiation with other agents, the personal agent acts as a representative of the user. The agents try to optimize their actions and to predict the "opponent's"[1] actions. For this purpose, they create models of the other agent's "character" and priorities. Thus, each personal agent models the character that the other user wants his/her agent to represent in the agent community.

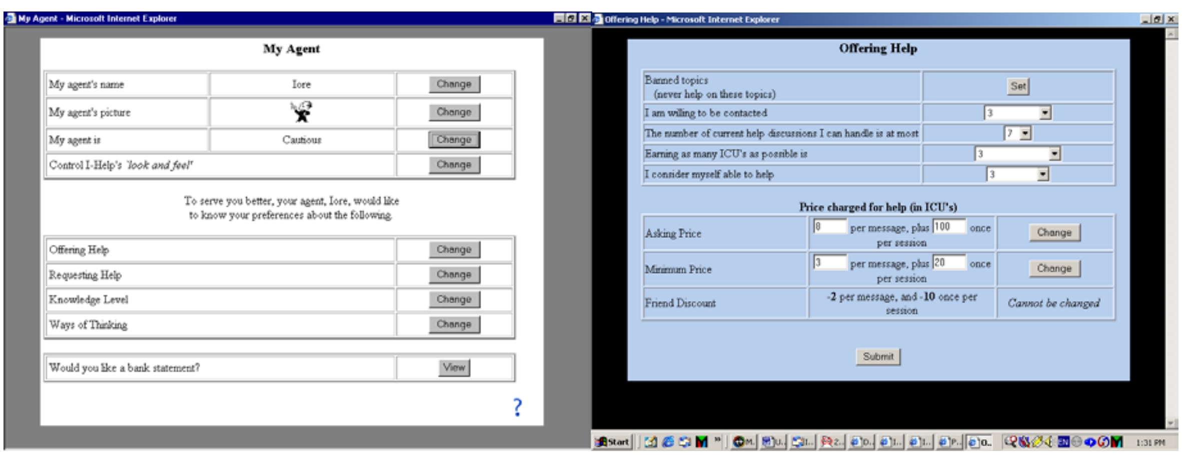

Figure 4: The user can change the characteristics and "character" of the personal agent.

During negotiation, the agents take into account relationships that may have previously formed between the users, for example, by changing the negotiation strategy (offering a discount for friends or an extra high price for enemies). After repeated successful negotiations followed by successful help sessions between the users, the agents offer to add a new relationship between the users in their models, thus increasing the number of "friends" of their users. In addition, personal

[1] Actually the agent of the other user is not a real opponent. However, this term is used to follow the terminology of the literature on agent negotiation and game theory.

agents collect references to other agents who keep information about the user, for example, diagnostic agents that have developed models of the user's knowledge in various domains.

Application agents (agents of discussion forum threads, web pages, search engines) build their own user profiles representing user features relevant to the context of the application, based on their interaction with the users and using "traditional" user modelling techniques. For example, the agent representing a web-based course on C++ stores data (quite coarse grained) about the progress of students who have used the course, which is matched to the taxonomy of topics/concepts taught in the class. It also stores user preferences with respect to the interface of the web-based course.

Diagnostic agents (agents representing web-based test items, questionnaires etc.) represent a special type of application agent that creates user models in a particular area of activity / knowledge, for a particular purpose, and with a particular structure. For example, a diagnostic application ("M") monitors user activities (browsing, reading and posting in the discussion forum), checks time-stamps and updates the level of user eagerness (one of the user's social characteristics). There are many different agents that model users' domain knowledge, and new ones can be added at any time. One diagnostic agent (labelled "S" in Figure 3) allows the user to fill a self-evaluation form (the form is the diagnostic application) to initialize another knowledge model of the user. There are agents representing on-line quizzes (application) in the system (labeled "Q" in Figure 3), developed specifically with the purpose of updating the student model on particular sets of topics/ concepts. When a particular time in the term comes when the students have learned and practiced these topics/concepts, the instructor sends a message to all personal agents of students that they have to contact the quiz agent. The quiz agent presents the quiz to the student, takes the student's answers, and updates the student's knowledge levels on the concepts/ topics examined in the quiz. Another kind of diagnostic agent translates student assignment grades into probabilities about the student's level of knowledge about course concepts. There is no integration of the different models of the student's knowledge at a central place (even if virtually central), which is a crucial difference with centralized user modelling approaches. Data is retrieved and integrated "on the fly" from the various agents only when it is needed.

Matchmaker (broker) agents manage collections of user models (profiles) for a certain population of users. Broker agents are specialized to deal with models of certain user characteristics (for example, for particular domains) and to perform matchmaking for specific purposes (e.g. locating the most knowledgeable helper, or locating a helper who has a compatible cognitive style). The brokers initially request and receive models/profiles from the personal agents and from diagnostic agents that create user models. Feedback from the users after help sessions is also integrated into these models with a higher priority, since it reflects the qualities of the user as a helper, which is directly related to the purpose of matchmaking. Users are notified by their personal agents only when a deal has been arranged, and they can agree to participate in the help session or they can discard the message. If the user always discards the notifications of his/her agent, he/she won't be able to earn virtual currency (ICUs). The matchmakers serve also as a "better business bureau", keeping track of personal agents whose users break deals. Matchmaker agents do centralize information in order to make initial decisions about appropriate peers to deal with a help request, but it is important to realize that they do not really keep global models of users.

Instead they keep only the information they need, and even this is integrated according to context-specific rules, reflecting the purpose of matchmaking -- by giving more weight to some of the data (e.g. the more recent, more objective etc.). In fact, there are many types of matchmaker agent, specialized according to various purposes (see section 3.2.1).

As can be seen from the discussion above most of the communication and reasoning about users' knowledge is distributed among the agents in the system. User modelling and adaptation is thus fundamentally fragmented and localized.

2.3 WHY A MULTI-AGENT ARCHITECTURE?

PHelpS (Collins et al., 1997), the predecessor of I-Help, used a client-server architecture. All user data was stored in a database on a server and was managed by one central matchmaker program, which was accessed by the clients (the user interface) when the user requested help. The clients provided context information (the current task) and the matchmaker selected users that had completed this task and in addition had some other desirable characteristics, for example, have already successfully helped somebody else with the task, spoke the same language as the person requesting help, etc. In this way, the matchmaker also played the role of a user modelling server, providing information (in this case the phone numbers of people suited to help) which was adapted to the context of the user (current task).

This centralized approach worked well in this relatively simple distributed application. However, when we decided to apply it to a University environment, it was no longer sufficient. The multitude of heterogeneous help resources and the dynamics of their availability could no longer be accounted for by a (logically) centralized component. It did not make sense to have one matchmaker, since matchmaking with various criteria, using various sets of users needed to be done. Other type of matchmaking were needed to match people with electronic help resources, like threads in the discussion forum and web-based resources. We could have added new components to the basic PHelpS architecture, for example, an information retrieval component for web resources, a component retrieving relevant discussion threads, etc. However, we were not excited by such a conservative approach, since it would have led to an eclectic and complex architecture that would become increasingly difficult to modify and extend. Also, it would have been hard to combine user data from the usage of the various resources since such resources can be added at any time by anyone. For example, it would have been difficult to retrieve for peer-matching purposes user modelling information about people sending good postings to the discussion forum because communicating information across several centralized architectures (of peer-help matchmaking, information retrieval for help request and discussion forum search) is generally harder than in a flat, unified architecture.

Another reason for deciding to "de-centralize" the PHelpS architecture is related to the implementation. In PHelpS the number of simultaneous interactions was small. When we tried in fall of 1997 to apply the same centralized architecture to support peer help in a university class with about 40 users, interacting simultaneously, it lead to a collapse of the service. The reason was the enormous demand on the database connection with the application. This reflected the fact that centralized systems are inherently vulnerable. These problems have been recognized (Fink, 1999, Fink & Kobsa, 2000) and approaches to replicate the user-modelling server while remaining "virtually central" have been proposed (Fink, 2001). A good analysis of the benefits and tradeoffs of centralized and decentralized architectures for user modelling in expert finding is provided by Seid & Kobsa (2000).

There are also other reasons for adopting a multi-agent approach to developing complex heterogeneous environments. First, a metaphor is needed that reduces complexity for users and allows them to concentrate on their primary goals or tasks, while supporting their learning and collaboration. A personal agent /assistant provides a metaphor that enables the user to delegate tasks / goals to the system without having to specify all details and be always in control. Second, an agent-based approach is inherently modular and offers extendibility. For example, new matchmakers can be added as needed, as long as they speak the same language and share the same ontology as at least some of the agents. It also allows better handling of load distribution and task balancing. Third, it becomes important to provide a regulating mechanism in such a complex environment, ensuring equal chances for all to participate, for example, not just the best helpers that a central system may recommend. Multi-agent system research emphasizes agent economies and societies, and investigates possible ways to bring them to stable equilibrium states. Finally, the field of multi-agent systems opens a number of extremely interesting and potentially useful research avenues concerning inter-agent negotiation, persuasion, competition and coalition formation in agent societies. We can learn from the adaptability, robustness, scalability and reflexivity of social systems to come up with more powerful multi-agent technologies for decentralized applications.

I-Help provides an excellent environment for studying these issues. A distributed, agent-based architecture reflects naturally the distributed web-based environment in I-Help. If one views "live" sessions with peer helpers and electronic peer help (discussion forum postings; on-line materials) as help resources provided by agents (human in the former case, software in the latter case), a multi-agent architecture provides for a natural collaboration between humans and software agents.

2.4 TECHNICAL REALIZATION

Before creating our own agent architecture, we explored off-the-shelf agent frameworks, but they didn't scale well. Especially the goal of handling large numbers of complex agents was difficult due to the inefficient use of computational resources of many agent frameworks. This forced us to build our own multi-agent infrastructure (Deters, 2000, 2001) that has proven to be critical to our success in scaling up I-Help over the last two years. CORBA is used as an object sharing protocol, since it is a standard and simple way to ensure a scalable system.

The most recent version of the agent infrastructure 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-utilized processors. If one processor fails, the entire set of agents that it supported migrates onto a new processor without interruption. This ensures scalability to well over 1000 simultaneously active agents. We used simultaneously up to 8 different matchmaker agents (see section 3.2.1). This implementation offers also a rule-based expert system shell on board each agent, permitting the agents to be "programmed" in flexible ways. As we incorporate more and more functionality into the I-Help multi-agent architecture, it becomes easier to modify a particular agent's capability and watch its effects on the system.

2.5 DEPLOYMENT OF I-HELP

In this section we briefly overview various deployments of I-Help. We hope to demonstrate in particular the scale of I-Help. For more details see (Greer et al., 2001). There have (to date) been four deployments of I-Help in classes at the University of Saskatchewan: (1) Sept.-Dec. 1999; (2) Jan.-Apr. 2000; (3) Sept.-Dec. 2000; and (4) Jan.-Apr. 2001. (A fifth deployment was underway at the time this paper went to press.) In the first two deployments, there were two separate but communicating systems: I-Help Pub (handling access to electronic resources) and I-Help 1-on-1 (handling requests for human help). They were integrated in deployment 3 and deployment 4. Deployments 1 and 2 of I-Help Pub allowed students to post questions and answers in threaded forums, having a structured, organized environment as a main benefit to the learner. Deployment 1 was available to 600 users. Deployment 2 was available to around 1000 users, but was actually used by about 750. Deployment 3 was available to 1600 students taking undergraduate courses in the Department of Computer Science at the University of Saskatchewan, and to 100 students enrolled in two courses in Law at the same University, with some 900 students actually using it. The fourth deployment of I-Help Pub was in Computer Science and Law courses at the University of Saskatchewan and was deployed in a languages course in the UK and France, with a similar total number of potential student users as in deployment 3. We have not yet fully analysed the data for deployment 4, but estimate a similar level of involvement as for deployment 3. It is planned to continue to use I-Help in all Computer Science courses at the University of Saskatchewan and in other courses both in Saskatchewan and elsewhere, including in Columbia (in collaboration with the Connexiones project at EAFIT University in Colombia).

We now attempt to give some feeling for the scale of user modelling in the third deployment of I-Help, the most recent thoroughly analysed version. The number of agents in the third deployment was over 1000 (personal agents of active users and various kinds of application agents). The number of user models was much higher than the number of agents, because information about each user is kept by many agents throughout the system. Each user had up to 20 fragmented models of himself/herself, held by their own personal agent, several matchmaker agents, and other users' personal agents. Overall, there were over 10,000 such fragmented models distributed through the I-Help system. The information kept in each model contained preferences, rankings, ratings, and numeric overlays on course topics depending on which agent creates the model and for what purpose.

In summary, I-Help is an example of a system with many users interacting at any point of time with a varying pool of agents. In such a setting, there is no centralized user model associated with each user. Rather the knowledge about the user is distributed among the various agents who interact with the user (both human and software agents). Thus, the need for integrating user model fragments depending on the context and purpose grows in importance.

In the next section paper we present a framework, enabling a more structured discussion of the new features of distributed and fragmented user and agent modelling, drawing on examples of what has been done in I-Help, what is being done in our ongoing development and what will be done. Section 4 compares our approach with centralized user modelling approaches and discusses some open issues and problems that need to be resolved.

3. The "Central User Model" Fading Away

The key to making sense of the distributed user models is the ability to interpret multi-modal information from multiple heterogeneous relevant sources and to integrate this information as needed into a user model of appropriate granularity. The main questions boil down to how to manage all this information:

- How does one locate the agent who has a model of relevant user characteristics, given the context and the purpose for which the model is needed?
- How does one make sense of possibly inconsistent and even contradictory user information?
- How does one interpret models created by other agents?

As we discussed in McCalla et al. (2000), user modelling in this approach is viewed as a process – a computation over a space of four major dimensions: subjects, objects, purposes and resources. This shifts the attention from issues of representation (consistency, representation schemes, indexing) to the process of collecting, interpreting and utilising user data for a particular purpose. As indicated in Table 1, we can represent this computation in a functional notation, which we call "FSOPR" (from the letters involved in the user modelling function shown in Table 1). The notions of "single user model" and of "user modelling server" can be described at a high level with this notation (see Table 2). We discuss other kinds of user modelling in this notation as we go through the remainder of this section. While this notation has no ambition of broad generality to cover all possible scenarios for user modelling, it does help to clarify the issues involved in this distributed approach to user modelling.

It is not by chance that most currently existing single system / single user models show a great variance only within *r2* (see Table 2). Classic user modelling approaches assume that the system does the modelling and the user is being modelled (i.e. *s* and *o* are fixed). They also assume that the model is created from analysing the user's interaction with the system only (*r2.info* is fixed), generally assuming unconstrained time and computational resources (*r2.time, r2.comput*). The main variety is in the reasoning mechanisms and knowledge representation (*r2.knowledge*) for user modelling. There are very few approaches that reuse models created by other systems, which are unaware that their models are going to be used later for different purposes by another system. While practically all of the commercial server systems (Fink & Kobsa, 2000) have the ability to import user data from external databases, there are restrictions when this can be done (in an initialization phase or at runtime) and how this data can be used (e.g., can new types of data be used by the inference rules?).

Our goal (and that of our notation) is to show that there are new dimensions open to multi-agent and multi-user modelling, related to the possible variety of purposes for which models can be built, resources on which they can be built, subjects and objects of the modelling process. We will show that this process is contextualized by the purpose and moment in time when modelling is needed. In the next subsections we will discuss the major components of the user modelling process in a multi-agent multi-user environment. We will focus mainly on the different possible subjects and objects of modelling and on a variety of purposes for user and agent modelling. In this discussion we will use examples from I-Help. We will show how the purpose of modelling and the object and subject of modelling determine the required resources.

Table 1. The FSOPR Notation.

where:			***user modelling → f (s, o, p, r ={r1, r2})***
subject ***s***			*(subject)* is the agent (software or human) *doing the modelling*
object ***o***			*(object)* is the human user or software agent that is *being modelled*
purpose ***p***			represents the purpose of the model, the adaptation or the activity for which the model is being created
resources ***r*** used in the modelling process	*referees* ***r1***		represent other agents (software or human), contributing to the modelling process
	resources ***r2*** *(computational and information resources)*	*time*	represents the time available for user modelling and adaptation for the particular purpose
		info	represents the information or data (from H-C interaction or from other user models) based on which the user model is generated or updated
		knowledge	the knowledge representation scheme and reasoning mechanism available to the subject to perform the modelling
		comput	the computational resources available for user modelling
function ***f***	process the information (*r2.info)* and data from other user models, obtained from referees (*r1)* using knowledge representation scheme, reasoning mechanism *(r2.knowledge)* and computational resources *(r2.comput)* in time period *(r2.time)*		

Table 2. Traditional User Modelling (also User Modelling Servers) in the FSOPR Notation.

		Traditional User Modelling
subject		System (or a User Modelling Server)
object		User
purpose		Usually one single purpose, which depends on the system and domain, like adapting the interface or a particular functionality of the application. In UM servers rich models are developed to serve a set of purposes (for adaptation in client applications).
referees ***r1***		Either none or the user herself (in inspectable / explicit user models)
resources ***r2***	*time*	Fairly fast, but usually no need to create / update the model and adapt in “real time”
	info	The "raw" data from the interaction between the user and the system that is used to create and update the user model
	knowledge	The "intelligence" involved in the modelling process on the side of the system: the knowledge representation scheme and inference mechanism used (predicate-value, BBN, logic-based formalism, machine learning/data-mining mechanism)
	comput	The computational resources available to the system for modelling.
function ***f***		adapt one or more features in the system's interface or functionality to the user's characteristics that are deemed relevant according to a predefined algorithm

3.1 SUBJECT AND OBJECT OF MODELLING

In this section we will discuss the *"s"* and *"o"* (the subject and the object of user modelling). In a multi-user multi-agent environment, in contrast to a single user single system there are many models developed by the various agents, as needed. Who models and who is being modelled is an essential question determining to a high degree the way modelling is done.

3.1.1 *Agents Modelling Users*

s - an agent; *o* - a user or users

This case comes closest to the traditional process of a system modelling the user. In I-Help the diagnostic agents use specific rules to infer values of particular user characteristics from raw data and from other agents. For example, the diagnostic agent computing the eagerness of a given student receives data about the number of times the student has logged into the system and the number of postings that the student has read and posted from the application agent of the discussion forum. Matchmakers collect user model information about the knowledge of all students that are in a given group (class) from diagnostic agents specific to certain topics of the class taxonomy and from the personal agents of the students, from which they receive both results of students' self evaluations and peer evaluations (after a help session). In integrating this information they use specific rules (e.g. give more weight to more recent information, to diagnostic agents over personal agents and to peer evaluation over self evaluation).

3.1.2 *Users Modelling Agents*

s - a user; *o* - an agent or agents

If the word "modelling" is used in the sense of creating, forming, shaping, we can say that the user models his/her personal agent's character and strategy, i.e. how the agent will appear to the other agents, how cooperative it will be, and how it will engage in negotiation. Since the agent represents the user in the system, in some sense, this is how the user is perceived by the other agents and (indirectly) by the other users. The "character" imprinted in the agent by the user is somewhat related to the notion of explicit user modelling (Rich, 1983) and open (inspectable, manipulable) user models (Bull & Pain, 1995; Paiva, et al., 1995). However, in the traditional notion of “inspectability” of user models it is assumed that the user can view the model that the system has created of him/her and the user corrects misrepresentations that the system may hold about him/her. In our case the user can create a "personality" of an agent (see Figure 4, the screen on the right), which deliberately differs from the user's. For example, the user may want her agent to be a tough negotiator, if the user is not.

3.1.3 *Agents Modelling Other Agents*

s - an agent; *o* - another agent or agents

To negotiate better, a personal agent needs to be able to predict the next move of the other agent. This move depends on the strategy and preferences of the opponent. Since the personal agents are self-interested (they work to satisfy best the needs of their users), it cannot be expected that the agents will reveal their priorities. Creating and maintaining a model of the opponent and sharing this model with other agents (gossiping) may help the agents overcome this problem. The I-Help personal agents use probabilistic influence diagrams to model the preferences of the opponent agent (Mudgal & Vassileva, 2000). More sophisticated techniques, like those proposed by (Gmytrasiewicz et al., 1998) and (Suryadi & Gmytrasiewicz, 1999) could be applied.

3.1.4 *Users Modelling Other Users*

s - a user; *o* - another user(s)

A user can instruct an agent about other users, either by providing evaluation in some form to be interpreted by the agent or by explicitly setting values for certain features in the models of other users (or their agents) maintained by the user's agents. For example, in I-Help users evaluate each other after a help-session. The personal agent of each user gives her a short evaluation form to fill; in effect a simple model of the other user's competence and helpfulness is constructed by the personal agent. Users also can instruct their agents about who their friends are and what their friends can be contacted about, what topics they are good at, topics for which they shouldn't be contacted. Thus, the user creates simple models of other users; the personal agent of the user utilizes these models to navigate better in the social space of the environment.

3.2 PURPOSES FOR USER MODELLING

Since modelling happens on demand, the purpose for modelling defines to a great extent all other factors. In the following discussion we describe (using FSOPR notation) various purposes for user modelling. The list of purposes discussed here is by no means complete; here we focus merely on the main purposes for which user modelling is done in I-Help.

3.2.1. *Locating Appropriate Resources: Information Retrieval and Matchmaking*

Information retrieval is a special case of resource brokering, where electronic resources or services are matched with a particular information need of the user, expressed in a request or inferred (see Table 3). In I-Help the information retrieval brokers are agents that find web pages or postings in discussion forums relevant to the help request of the user. All information resources are represented either individually, or in groups (clusters) by an application agent. The application agents maintain models of the resources they represent. For example, the application agent of a thread in a discussion forum maintains a list of the topics addressed by the postings in the thread. A broker agent maintains models of some set of relevant application agents, thus achieving an index of the resources represented by the agents.

In the case of an information request by the user or by the personal agent of the user (based on a perceived user need), the broker agent locates the agents representing the most relevant resources. It is desirable that the models of the resources maintained by the application agents follow the same ontology, so that the broker can maintain a consistent index. In I-Help the forum administrator (usually the class instructor) provides a topic taxonomy to serve as an ontology (see Figure 2). User information requests have to be related to some of these topics. However, in general, a strict adherence to taxonomy is not necessary. Other factors, besides the topic can be used in the search (e.g. postings by particular individuals or postings within a certain temporal interval, or responses to a particular posting, etc.). The redundancy of the system ensures that even if the agents don't strictly follow a specific taxonomy, a relevant resource can be retrieved if at least a subset of agents does follow the taxonomy.

Table 3. User Modelling for Locating of Resources.

		Information retrieval	**Matchmaking**
subject		broker agent	matchmaking agent
object		user	user
purpose		to find appropriate information resource(s) for particular goal	to find appropriate other user(s)
referees ***r1***		application agents representing on-line information resources or applications	personal agents of other users (potential peer helpers)
resources ***r2***	time	must complete in "real time"	must complete in "real time"
	info	user query, browsing history	help request from user, activity context
	knowledge	the indexing scheme, algorithms for calculating the appropriateness	the representation format of the peer helper models, algorithms for selection
	comput	a lot, since search it is done usually on a server	as in information retrieval
function		Match *r2-i* and models obtained from *r1*; select models/agents from *r1* that satisfy a particular criterion	Match *r2-i* and models obtained from *r1*; select models/agents that satisfy a particular criterion

Matchmaking, along with information retrieval is a special case of resource brokering. In I-Help, there are several broker agents that find ready, willing, and able peer helpers for a particular user and learning need. Each agent finds suitable peers according to different criteria (each corresponding to a slightly different purpose). One broker, for example, finds the most competent peers on the topic of the help request. Another broker finds peers that are currently available (on-line). A third broker finds peers that have particular social characteristics (e.g. eagerness, helpfulness, class ranking) that might be beneficial for helpers. A fourth broker finds peers that have a similar learning style to that of the user asking for help. A fifth broker finds peer-helpers only among the friends of a user. There is even a horoscope broker that matches users according to their star-signs! Thus, each matchmaker takes a different set of user characteristics and calculates a score using a simple ranking algorithm.

Typically several broker agents work together and pipeline their results to produce ranked short-lists of helpers that are optimal according to some combination of criteria. For example, in I-Help the personal agents are programmed to call the "competence" and "availability" matchmakers. Depending on the user preferences (i.e. how the user has instructed his/her personal agent), a cognitive style, friends-only, or a horoscope-based matchmaker can be called either as an alternative, or in combination with the previous two, to cut down further the list of possible helpers.

3.2.2 *Facilitating Interpersonal and Inter-agent Communication*

Modelling social networks has been an area of research since the 1930's in the social sciences, more specifically in sociometrics. However, user models so far have focussed only on inherent user features like knowledge, experience, and preferences. Vassileva (1998) argued that for distributed multi-agent multi-user environments it is crucial to model features describing the user on an interpersonal level, i.e. the relationships between two or more individuals. The importance of taking into account interpersonal relationships has been pointed out more recently also by Kelly & Jones (2001) and Cassel & Bickmore (2002). These relationships provide an important insight to the way the person is likely to pursue his/her goals, the scope of resources available to him/her, the people s/he is most likely to cooperate or collaborate with. Knowledge of these relationships can also help to better model some inherent characteristics of the user, for example, the person's interests ("tell me who are your friends and I will tell you who you are"). Table 4 shows two important types of user modelling that arise in inter-personal situations.

Table 4. User Modelling for Facilitating Interpersonal and Inter-Agent Communication

		Modelling Interpersonal Relationships	**Modelling Negotiation Strategy**
subject		personal agent	a personal agent
object		the user's relationships with other users	another personal agent or an application agent (opponent)
purpose		to facilitate the inter-user collaboration according to the social contacts of the user	to optimize the negotiation strategy, to benefit most from the interaction
referees ***r1***		the user (privacy concerns prevent obtaining the information from other users or agents)	other personal agents (who have encountered *o* before)
resources ***r2***	*time*	no need of fast response, usually off-line	in "real time"
	info	explicit data from user, data from success of negotiations	the offers/behaviour of the opponent agent, models created at previous encounters
	knowledge	the representation of the relationship as a scalar / vector of parameters or more complex; rules for goal adoption depending on relationship parameters	the representation chosen for the parameters relevant to the opponent agent's strategy, algorithms for inferring the parameters from the opponent's actions, and for calculating the anticipated next move of the opponent based on the opponent's model
	comput	a lot if done on server, limited if done locally	a few, usually done on board of the agent
function		infer and represent relationship parameters, use relationship as a filter when user sends and receives requests to other users, and in negotiation with other personal agents	integrate *info* and *r1*, calculate the probability of opponent's actions, optimize decision/strategy with respect to the prediction

In I-Help, the relationships of the user influence the negotiations happening between the personal agents. A simple representation of the user's relationships (a list of the names of the friends and enemies of the user) is included in the model maintained by the user's personal agent. For privacy reasons, neither the contents nor the result of any reasoning based on the user relationships is shared with other agents. Maintaining a list of friends and enemies allows the personal agent to filter out undesirable interactions for the user. This is achieved either by blocking help requests from enemies of the user (by setting an extra high price in negotiation with the helpee's agent) or by encouraging help-sessions with friends of the user (by providing a discount in the negotiation).

It is possible to develop a more sophisticated representation of interpersonal relationships between users taking into account parameters like importance, closeness, symmetry (dominant, peer, dominated), sign (collaborative, co-operative, competitive or adversarial), and roles (Vassileva, 1998). We believe that such a representation will help to achieve much more fine-grained reactions / adaptations in the agent's (and correspondingly, the system's behaviour) and there is currently a student project underway in this area in our research group.

Agents interact for various reasons: to buy and sell resources on a market, to exchange user model data, to share knowledge or experience (Maes, 1994) so that agents can learn to serve their users better. In some cases the agent interaction can be assumed to be cooperative (i.e. the agents will be willing to share their information or resources). In many cases, however, the agents are self-interested (non-cooperative). Personal agents in I-Help, for example, are interested in pursuing the goals of their users, but not necessarily the goals of other users or agents.

In an environment populated by self-interested agents, conflicts are bound to occur. Agents resolve these conflicts through negotiation. Various mechanisms for negotiation have been proposed (Durfee & Lesser, 1987), (Zlotkin & Rosenshein, 1991) ranging from multi-value auctions to bilateral negotiation (Zheng & Sycara, 1997). In order to perform well in negotiation, independently of the type of the particular mechanism and protocol, the agent benefits from knowing the state of the environment, including the other agent(s) participating in the negotiation and their priorities. Theoretical approaches to modelling the environment and the actions of other agents have been proposed and shown to bring benefit to the individual informed agent that utilizes them (Carmel & Markovitch, 1998), (Suryadi & Gmytrasiewicz, 1999).

Agents modelling each other in negotiation have been proposed and shown to be beneficial too (Sycara, 1988; Zheng & Sycara, 1997). In bilateral negotiation, the purpose of agent modelling and the type of adaptation is similar to adaptation carried out in non-cooperative dialogue systems where the participants infer the intentions and beliefs of the other participant (Jameson et al, 1994). In I-Help the agents model each other to better predict the reactions of the opponent in the negotiation, and thus to select a more efficient strategy. In the current system, the representation of the opponent's model is a probabilistic influence diagram. The agent recalculates the probabilities of the priorities (preferences) of the opponent at each step (each offer) during the negotiation. The modelling functions of the agents can be the same or different for the two negotiating agents. In I-Help right now all agents use the same modelling function for negotiation purposes. However, in more general and complex situations, when for example two agents are negotiating about the contents of another agent's or user's profile, the functions would have to be completely different.

We ran experiments comparing the performance of agents using the adaptive negotiation mechanism with modelling the opponent with agents who do not model the opponent and negotiate following a simple uninformed strategy (Mudgal & Vassileva, 2000). The experiments showed that modelling the opponent brings benefits to the agents in terms of better deals and less time spent in negotiation on average. This, however, comes at the price of a higher percentage of failures in the negotiations, which is explainable, since the personal agents try to bring the best deal for their users and reject deals that they don't consider profitable. The success of the strategy in ensuring deals depends also on the difference in the initial offers. If the difference is big, modelling the opponent makes the agent reject the deal sooner.

An interesting point here is that an agent, by trying to infer the preference and decision model of the opponent agent, actually creates indirectly a model of the user of the other agent. The "character" of an agent may not correspond to the user's character at all, but it expresses certain user preferences with respect to his/her agent, so indirectly, it is a model of the user too (at least of the features the users wants his or her agent to present to the other agents). The accuracy of this model determines to a high degree the behaviour of the personal agent during the negotiation and the outcome of the negotiation. This outcome is one of the major factors deciding whether or not the user is going to get help from the other user and also how high the costs will be. Therefore the functionality of the system as a peer-help finder is dependent on the quality of the models agents are creating.

3.2.3 *Users Learning about Themselves and Other Users*

In this section we look at four other types of user modelling involving users learning about themselves or other users: for reflection, for validation, for assessment, and for diagnosis. Table 5 outlines these in FSOPR notation.

Reflection. Making the contents of user models accessible to users can be helpful to promote reflection about their knowledge of a target domain (Bull & Pain, 1995, Paiva et al., 1995). With the broader information in fragmented models in multi-user systems, such reflection will concern not only domain content, but will be also focussed on social issues, e.g. "How do other users view me?"

I-Help enables several forms of reflection. The simplest one is for students to check who else is on line right now. On such a request, the personal agent consults the "who is on line" broker and obtains a list of personal agents whose users are active at the moment. Then it requests from each agent identification information about its user, if it has been allowed by its user to share this information, and presents parts of this information (e.g. the user's name and picture) in a table.

Reflection allows the user to inquire into how he/ she is being perceived by other users or agents with respect to his / her knowledge. In principle (but not in I-Help yet), students may also reflect on reactions of others who have viewed their work, which, as (Bull, 1998) suggested, might lead to a better understanding of their own difficulties. Users in I-Help can inquire about how they are perceived with respect to their helpfulness (how people to whom he/she has given help have evaluated him/her) and about their group / class ranking. This can be helpful perhaps to assist users in rethinking their attitude to the group, or for learners to compare their performance with that of their peers. They may wish to see how well they are doing with respect to the average student, or they may wish to view the possibilities attainable by high achievers (Kay, 1997). In general, seeing other agent's views of oneself can be very valuable for reflection.

Table 5. User Modelling for Reflection, Validation, Assessment and Diagnosis.

		Reflection	**Validation**	**Assessment**	**Diagnosis**
subject		personal agent	a personal agent	assessment agent	diagnostic agent
object		certain user characteristics	the user's belief about a fact or concept in the world	as for reflection	as for reflection
purpose		to find out how the user is viewed by other agents/users (with respect to *o*)	to improve *o* (to confirm a belief, to add new beliefs, to leverage others)	to generate an overview of certain user features for some evaluation purpose	to find out / update particular characteristics of the user
referees ***r1***		other personal agents (who have encountered *o* before)	other personal agents (who have encountered *the fact* before or have a relevant belief)	other agents that have created models of these character-istics for some purpose	∅ if only raw data from interaction with the modelled user is used as input, or data from models developed by other diagnostic agents
r2	*time*	fairly fast response needed, but not necessarily real time	as for reflection	as for reflection	as for reflection
	info	the user's request (the area / feature, etc. that the user wants to reflect on	the user's request (the belief or the fact, that the user wants to validate)	a set of character-ristics to be evaluated, a depth of search, a scale	observation of user actions, or direct interaction with user
	kno wle dge	the representations or *r1* containing the relevant user parameters, mechanisms for knowledge sharing / aggregation and interpre-tation utilized by the personal agent	as for reflection	the representations of *r1*, algorithms for calculating appropriateness score	depending on the kind of model being built, various traditional user modelling representation and reasoning techniques
	com put	a few if done on board of the personal agent, a lot if done off-line on server	as for reflection, but it is usually done on board of the personal agent	a lot, usually done on server	as in reflection
function		integrate *info* and *r1*, aggregate / interpret the relevant data and compile an appropriate presentation	as reflection	generate an assessment based on a certain set of features and certain scale	update the representation of the user model according to most recent data from user input *i*

We have also been exploring techniques for knowledge externalization and knowledge visualization, that is putting knowledge into a form that it can be easily understood by the user(s) as they reflect (Zapata-Rivera & Greer, 2000). Such techniques can be crucial to making reflection useful to users.

Validation. The user can make use of the multiple viewpoints contained in the various user models maintained by different agents in the system to validate (i.e. confirm or deny) particular opinions/knowledge that he/she has. This could be used not only to confirm domain knowledge, but also to find out other people's opinions about a person's social characteristics.

In I-Help the users can validate their knowledge of a given topic. The user can request from her personal agent to be notified when a given user posts a reply or a question on a given topic, thus seeing the other user's understanding on the topic. The user's personal agent can request to view details about other users' evaluations of certain postings, calculate a score for the resource and compare it with the score (vote) given by the user. Such a mechanism, together with reputation mechanisms (for example based on class ranking, produced by diagnostic agents) can serve a social navigation or recommendation tool to find good on-line resources, as is already done in hypertext systems (Terveen & Hill, 1998; Munro et al, 1999).

Assessment is a third type of user modelling that supports a user who wishes to learn about other users or about himself/herself. A user (e.g. an instructor, evaluator, boss) may want to get an overall evaluation of a user for some reason (for example, assigning grades to students). In a distributed multi-agent user model, potential assessment material may be of different kinds, from different sources, and exist in different locations. Thus these representations must be selected, transformed, aggregated, and then interpreted as an assessment.

It is possible in principle to create assessment agents, which generate an overall evaluation of certain features of any given user for the purposes of assigning grades, job performance review, before entering into negotiations, etc. The purpose of modelling in this case is more similar to traditional user modelling, focussed on a particular user characteristic (knowledge on one of more topics) and trying to get a holistic view. However, due to the diversity of the information sources, issues of consistency and interpretation become of central importance. Assessment is easier to realize when agents are validating models that have been created for an identical purpose, and with a similar modelling function. It is much harder to extend a model with information collected by another agent, for a different purpose with a different function. Several assessment agents are currently being developed in I-Help that focus on evaluating the knowledge or activity of a given student with respect to the whole class or with respect to a user group (e.g. project teams).

Diagnosis has been the focus of much work in the areas of user modelling and intelligent tutoring systems. Diagnosis is the act of a system inferring representation for a user model, according to the user's input and performance, or from their behaviour during problem solving and past history. There are a range of diagnostic techniques which may be used to construct a user model. However, we are here not so much concerned with the techniques themselves, but rather, with the more general issues of diagnosis as applicable to multi-agent user models.

As in an intelligent tutoring system, diagnosis may be performed by drawing on a learner's current problem solving activities, perhaps taking into account past learning behaviour, in order to determine what difficulties the learner may be having, and why. However, in cases where model contents are spread across a variety of locations, in a variety of forms, the diagnostic process is more complex. It must either extend back to the transformation of raw data from the various sources, or it will need to draw on some such process performed by another agent. Typically, diagnosis takes time, so it is unrealistic to expect real-time results. However, in order to ensure adequate adaptation, it is desirable that diagnosis be fairly fast (depending on the application and the type of adaptation required).

In I-Help diagnosis is performed by diagnostic agents, whose role is to create and update specific profiles / models of the users for use by other agents. The diagnostic agents can be invoked by some of the matchmakers, by the personal agent of the user, or by the user himself/herself. Currently, they are called by the personal agents to create "objective" models of the user's knowledge. Since the agents can replicate themselves on demand when they need to interact with a particular user, at any moment in time there can be many identical agents, interacting with different users and creating different user models.

3.3 The Process of User Modelling

From the discussion of the various purposes for user modelling above, it is clear that the processing function and resources involved depend strongly on the 3 main factors: who is doing the modelling, who is being modelled, and the purpose of modelling. Specific processes and mechanisms have to be developed for different possible combinations of these factors, and some of them might be domain-dependent. Developing user modelling processes should be done pragmatically, on demand, for specific domains and adaptation types that need to be supported. We are currently working on an ontology for user modelling purposes, involving different types of subjects and objects in I-Help. The next step will involve defining a library of standard processes for each purpose.

In conclusion, it is useful to summarise several features of multi-agent multi-user modelling. First, at any given moment of time there is no consistent model of a user; there are many "snapshots" taken by various agents, in different contexts, containing different information. These models are normally small, simple, and valid only in particular context. They can be stored anywhere – in a centralized or distributed database, or in files known to the agents that created them. Unlike Fink's (2001) physically distributed but virtually centralized user modelling system, even if stored in a centralized database (as is the case in I-Help, where all the agents store their models in an ORACLE database), the user data in our approach is virtually distributed since only the agents that have created the models know how to access them and how to interpret the information in them. Other agents can access this information only through requesting it from the agents who created the models. In order to find the agents that have the appropriate information, they use brokers or matchmakers that know about agents that keep models of a certain user feature. These centralized "points of reference" allow more efficient search in the distributed system. It is important to note that there are many such central points (as many as there are matchmakers) and they are not absolute, but relative with respect to particular matching goals. The agents can answer requests about particular user features if they possess information about these features. Thus, user modelling becomes a focussed process of collecting and integrating information about the user (and about agents) at particular times and with specific purposes.

4. Discussion

In distributed environments user modelling can be a process of assembling and summarizing fragmented user information from diverse sources at the time when it is needed for a particular purpose. In contrast to centralized user modelling approaches, like user modelling servers (Kobsa, 2001a), where the goal is to collect at one place as much information as possible about the user, so that it can be used for many purposes, we propose to shift the focus from the model itself to the process of modelling at the time of use. Thus the model is adaptive, computed "just in time" (Kay, 1999), and only makes sense in the context in which it is created (time, purpose, the agent creating it, the agent being modelled, the available sources of information). Under these circumstances it is close to impossible to expect consistent models within any particular subset of agents. However, even though they are computed within the resource constraints at the moment of usage, these user models can still be useful, reliable and appropriate for the purpose at hand.

User modelling servers are an alternative approach to support a variety of distributed applications. The advantages of user modelling servers have been summarised in (Kobsa, 2001a):

- Information maintained centrally or at a virtually central location can be used by many (distributed) applications; also information inferred by one application can be used by another application at the same time for a different purpose.

Multi-agent multi-user modelling offers the same advantage, without the need to maintain a physically central or even virtually central repository of user data. This implies that the requirement for equipping a user modelling server with strong inference capabilities will be less important and that the inherent vulnerability of a central location is avoided.

- Information is stored in a non-redundant manner in centralized user modelling, which more readily supports consistency and coherence of the user information.

This is not the case in our approach. There is no global coherence and consistency. In fact, for an environment as complex as I-Help with so many sources of user modelling information, consistency is in practice impossible. Every agent makes sense of the information available from its own point of view as it needs to, using its own heuristics about relevance. This puts the emphasis on interpreting the information for end use rather than on the information itself. It also leads to higher stability and scalability and can even serve as a kind of privacy protection, since there is no one ultimate reference point about each user to be hacked.

- User modelling servers allow the application of methods and tools for security, identification, authentication, and access control (Schreck, 2001).

This is not so much needed in our case, since there is no central model to be protected. Just accessing the information will not be sufficient to understand it: most of the information makes sense only in the context of interpretation. This is certainly true in I-Help where the most of the user information is about knowledge or learning preferences. There are types of user data that make sense in a wide range of contexts, e.g. names, addresses, credit card numbers. To obtain such sensitive data, of course, agent authentication, access control, secure protocols and encryption will be required in the agents' communication. Distributed user models give rise also to some other privacy concerns that are discussed later in this section.

- Information about user groups can be obtained more easily, as well as group user information can be added to the user models.

This is not so easy to achieve with multi-agent multi-user modelling. Extraction of information common for a user group can be done if needed by providing special diagnostic agents that compare user models of given user groups. Inserting information common to a group of users can be done by creating an agent who maintains models of the users from this group with the desired information. This agent will be contacted by other agents, if the user information that it maintains is relevant for some purpose. However, it will be just one more source of user information, not a "correction" in the ultimate user model.

Fink (1999) has defined a set of criteria for evaluating user modelling architectures. These criteria are: speed of adaptation, extensibility, load balancing, fail over strategies, transactional consistency. From the previous discussion, it seems clear that multi-agent multi-user modelling meets the first four criteria. The final criterion, transactional consistency, involves ensuring that the user model is not being changed at the moment when some application is using it, since it might lead to an inconsistent user interface. In our approach this can be achieved (if the purpose deems it to be necessary) by requiring agents to make local copies from the user models they receive from other agents and use these local copies instead of requesting new information in the course of adaptation.

As a final note on the issue of centralized vs. decentralized user models, it should be pointed out that our approach in practice may need one kind of centralization: a catalogue of purposes for user modelling, which define relevant parameter values (subject s, object o, resources $r1$ and $r2$) and mechanisms f (processes) for each purpose p. In this way the agents will not need to "know" how to perform user modelling in each possible context and for each possible purpose, but will be able to retrieve a mechanism that fits best the purpose and refine it to fit the context. The ontology of purposes will have to be developed manually and can be elaborated into a library of user modelling functions. Such libraries could be centralized, thus creating a new type of a user modelling server (not storing user data, but storing "know how" about how to integrate and interpret user data for various purposes) or distributed (offered as services by competing specialized user modeller agents). A graduate student project in our laboratory is currently exploring the creation of such a purpose ontology.

Fink and Kobsa (2000) have defined another set of dimensions for evaluating user modelling architectures, which generalize and extend those proposed by Fink (1999): performance, scalability (user modelling workload), integration of pre-existing user information and domain knowledge, and privacy protection. While performance and scalability seems to be an inherent advantage for multi-agent multi-user modelling, the issues of integration of pre-existing user information and domain knowledge and privacy need more discussion.

The first dimension (performance) is related to ensuring common agent communication languages and shared ontologies, which are in general very important for the functioning of a multi-agent system. There has been significant progress in the area of ontology design (Fridman-Noy & McGuinness, 2001). In a heterogeneous open environment, however, it might be hard to ensure adherence to standard ontology. Fortunately, the lack of a common representation language for user models will, we believe, not be a critical impediment to distributed user modelling, because of the redundancy in the system and the power of localized control. It is not likely that agents are going to request information that they will not understand and even more unlikely those agents that don't understand a request will reply. The redundancy of user information provided by the multi-agent environment compensates for the lack of "understanding" and the sometimes-insufficient interpretation abilities of the agents. However, there must be a base level of understanding for the agent society to function.

The scalability of a multi-agent multi-user approach can be ensured to a certain extent on an agent level through mobility of code/data and load balancing. I-Help scales up currently very well with over thousand agents. However, if there are millions of users in open environment, like for example, in some e-commerce applications, bottlenecks might appear. If each agent has to keep several models of every other agent and user it has ever encountered and it may lead to an explosion of models in the system. This will likely slow down the retrieval of relevant profiles and lower the performance of the system. One solution would be to allow the agents to be able to generalize their models over time (i.e. learn a more complex model from different experiences with a given user/agent) or across other users and agents with which it interacts (i.e. create group / stereotype models). This will require learning and consistency maintenance techniques that are currently deployed by user modelling servers. Another solution would be to break down the pool of users and agents into smaller subsets (coalitions) depending on the most frequent interaction patterns or on trust relationships (Breban & Vassileva, 2001).

The last dimension defined by Fink and Kobsa (2000), ensuring privacy, is very important. User modelling server providers have to take measures to conform to the privacy legislation of the countries where they reside, of the countries whose residents they model, and of the countries in which their servers operate (Kobsa, 2001b). Since most of the current applications that are currently served by user modelling servers are e-commerce applications operating on the web, it seems that the constraints on the collection and processing of user data are almost prohibitive.

Multi-agent multi-user modelling seems to open a "back door" for user modelling in a way similar to the one of distributed peer computing programs like Gnutella or Morpheus provided for sharing files. The reason that it was possible to shut down NAPSTER was that there was a centralized infrastructure, which had an obvious provider. If there is no central repository of user data and a centralized infrastructure, there is no one to sue. On one hand, the idea of delegating the responsibility of user modelling to autonomous (and even worse, economically motivated) agents can be worrisome. It would be hard to guarantee that agents will serve their users' best interests when they share their user information with other agents. In this case, the new role of personal agents may become similar to the role of lawyers protecting the interests of their users, i.e. instead of delivering information about their user's interests and preferences, being careful to reveal as little as necessary. However, even if users trust their agents to be intelligent about revealing personal information (at the cost of much functionality and many lost benefits of multi-agent multi-user modelling), they can't prevent other agents from modelling their behaviour, albeit possibly illicitly, just as people cannot prevent other people from observing their actions, making conclusions and thinking certain things about them. Any person or agent who has encountered the user or his/her agent will develop a model of him / her, which it may divulge to a third agent or user on request.

Protocols for interagent communication ensuring security (by use of encryption, authentication etc.) can be useful to prevent third party agents (or users) from observing or receiving user model information. However, an interested malevolent party may pretend to be a benevolent agent requiring information about a user or another agent for some legitimate (but false) purpose and receive the user information entirely "legally". By assembling information about the user / agent from many other agents, this malevolent agent may obtain a lot of information that can be used for some harmful purpose. However, this could be a problem for a user modelling server as well. In both cases, it boils down to proving credentials to some authority (the centralized user modelling server or to each agent from which user information is requested). In a multi-agent environment, however, there are also social means available to protect the society of agents from malicious agents, for example, creating networks of trust among the agents (Winter, 1999).

Evaluation is a new problem for multi-agent multi-user modelling. Even though it was not considered as a dimension for judging user modelling architectures by Fink and Kobsa (2000), we realize that this is one criterion where user modelling servers have an advantage, because of the large amount of usage data they collect. The main difficulty with our approach is what criteria to use for evaluation. There is nothing to compare with, no "baseline". There is no guarantee that the user model compiled by an agent in a particular situation and for particular purpose will even be similar to a model compiled at different time since the resources available or other contextual elements may change. In this way adaptation decisions and system behaviour at any moment can be different. We are not concerned that this will bring usability problems like inconsistency of the interface etc., since the applications where our approach is suitable are mainly resource location types of applications, where the adaptation takes form in the resources found (e.g. information retrieved, recommendations, advertisements displayed, etc.) rather than adapting interface layout (buttons, menus, windows) which may confuse the user by the adapted functionality. The problem is to show that the personalized offering by the system is better than the non-adapted one, i.e. that user modelling plays a positive role.

Traditional user performance measures do not seem appropriate. Peer helper finding in I-Help depends on the situation at the moment of the help request, on who is on line, on what the results of negotiation between the agents have been. There is no guarantee that the best match will be found for the user. Even if it is, the final success of the help session is entirely in the hands of the users – there is no guarantee that the best possible candidate helper would in fact act as a good helper in the particular case. We believe that there must be a change in the evaluation criteria for multi-agent multi-user systems. User satisfaction, along with robustness of the whole system, stability in its performance, productiveness of agent interactions, and economic well-being of the agent society will become some of the measures for evaluating such a system, instead of only trying to evaluate if the individual user's needs were served in the best theoretically possible way at the moment. If the behaviour of the system where agents use profiles is better than random or better than a simulated system behaviour without profiles, this might be a sign of a better system. Usage can become as respectable an evaluation criterion in research as it is in the market: a system that is good will be used; if it is not good, users will abandon it, its economy will collapse, the agents will cease their interactions. It seems that in such a system, there will never be a guarantee that user modelling will bring an advantage in each individual case.

5. Conclusions

This paper has argued for shifting the focus of user modelling from syntactic (representation-focussed) to pragmatic (purpose-focussed) issues, and from the data structure (representation) to the process (the computation). In new distributed computational architectures such a view will not only be useful, but necessary. User modelling and agent modelling will be a fragmented activity, performed on demand as a function of the purpose of the modelling, the people or agents being modelled, and the resources available. User modelling will be carried out for a wide variety of purposes, for many of which user modelling has to compute social aspects of the user. This should be easier than it has been in the past given the vast amount of information that will be available about the user interaction in the ubiquitous computing world that is emerging. Defining purposes, relevant sources of user modelling data and mechanisms for retrieving and integrating user model becomes a major research goal.

These revised ideas about user modelling will shift the user modelling research agenda. Processes such as retrieval, aggregation, and interpretation of user modelling information created by different agents will become the main focus of user modelling research. Many interesting research issues surrounding these techniques will have to be explored and in fact, we have a number of research projects underway that investigate these techniques. In a fragmented, distributed, and universally accessible technological environment, user modelling will increasingly be viewed as essential to building an effective system, but will also increasingly be seen to be tractable as new techniques emerge from these explorations. Nevertheless, as our experiments have already shown, it will not be necessary to resolve all of these issues in order to usefully user model.

Acknowledgements:
This work has been partially supported by the Telelearning Network of Centres of Excellence under project 6.28. The authors also wish to acknowledge the funding provided to aspects of this research by the Natural Sciences and Engineering Research Council of Canada, the Canadian Foundation for Innovation, the Saskatchewan Innovation Fund and the University of Saskatchewan. Many graduate students, research associates and colleagues have contributed to the ideas in this paper. We would particularly like to acknowledge Ralph Deters, Lori Kettel, and Susan Bull. Special thanks go to the three anonymous reviewers, whose constructive comments and recommendations helped us to improve and focus the paper.

References

Adar E., Huberman B. (2000) Free Riding on Gnutella, First Monday, available on line at: http://www.firstmonday.dk/issues/issue5_10/adar/index.html

Breban S., Vassileva J. (2001) Long-term Coalitions for the Electronic Marketplace, in B. Spencer (ed.) *Workshop Proceedings E-Commerce Applications*, Canadian AI Conference, Ottawa, June 7-10 2001.

Bull, S. & Pain, H. (1995) "Did I say what I think I said, and do you agree with me?": Inspecting and Questioning the Student Model, in J. Greer (ed), *Proceedings of World Conference on AI in Education AIED'95*, AACE: Charlottesville, 501-508.

Bull, S. (1998) 'Do It Yourself' Student Models for Collaborative Student Modelling and Peer Interaction, in B.P. Goettl, H.M. Halff, C.L. Redfield & V.J. Shute (eds.) *Intelligent Tutoring Systems ITS98*, LNCS No1452, Springer-Verlag, Berlin Heidelberg, 176-185.

Carmel, D., and Markovich, S. (1998) Learning Models of Intelligent Agents, in *Proceedings 3rd International Conference on Multiagent Systems. ICMAS'98*, 64-71.

Cassel J., Bickmore T. (2002) Negotiated Collusion: Modelling Social Language and its Relationship Effects in Intelligent Agents. This issue.

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 6th Conference on User Modelling, UM97*, Springer Wien New York, 327-337.

Deters R. (2000) Maintainability & Dependability of Mobile Autonomous Agents, *Proceedings Workshop on Infrastructure For Scalable Multi-Agent Systems, Autonomous Agents'2001*, Barcelona.

Deters, R. (2001) Towards Scalable MAS, *Proceedings International Conference on Enterprise Information Systems ICEIS 2001*, Setubal, July 7-11, 2001.

Durfee, E., and Lesser, V. (1987) Negotiating Task Decomposition and Allocation Using Partial Global Planning. Huhns, L., and Gasser, M. (eds.) *Distributed Artificial Intelligence*. Vol. 2, Morgan Kaufmann: San Mateo, California, 229-243.

Fink, J. (1999) Implications from Transaction Management for Adaptive and User Modelling Systems, in *Proceedings 7th User Modelling Conference UM'99*, Springer Wien New York, 191-200.

Fink, J. (2001) User Modeling Servers – Requirements, Design, and Implementation, Doctoral Dissertation, Dept. of Mathematics and Computer Science, University of Essen, Germany.

Fink, J. and A. Kobsa (2000): A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web. *User Modeling and User-Adapted Interaction* **10**(3-4), Special Issue on Deployed User Modeling, 209-249.

Fridman-Noy N. and McGuinness D. (2001) Ontology Development 101: A Guide to Creating Your First Ontology'. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

Gmytrasiewicz, P., Noh, S. Kellogg, T. (1998) Bayesian Update of Recursive User Models, *User Modeling and User Adapted Interaction*, **8** (1-2), 49-69.

Greer, J., McCalla, G., Cooke, J., Collins, J., Kumar, V., Bishop, A. and Vassileva, J. (1998) The Intelligent HelpDesk: Supporting Peer Help in a University Course, *Proceedings Intelligent Tutoring Systems ITS'98*, San Antonio, TX, LNCS No1452, Springer Verlag: Berlin pp.494-503.

Greer. J. McCalla, G, Vassileva, J., Deters, R., Bull S., Kettel, L. (2001) Lessons Learned in Deploying a Multi-Agent Learning Support System: The I-Help Experience. *Proceedings International AI and Education Conference AIED'2001*, San Antonio, IOS Press: Amsterdam, 410-421.

Jameson, A., Kipper, B., Ndiaye, A., Schäfer, R., Simons, J., Weis, T., & Zimmermann, D. (1994) Cooperating to Be Noncooperative: The Dialog System PRACMA In B. Nebel & L. Dreschler-Fischer (Eds.), *KI-94: Advances in artificial intelligence*. Springer Verlag: Berlin, 106-117.

Kay, J. (1997) Learner Know Thyself: Student Models to give Learner Control and Responsibility, in Z. Halim, T. Ottmann & Z. Razak (eds), *Proceedings International Conference on Computers in Education ICCE'97*, AACE: Charlottesville, 18-26.

Kay, J. (1999). A Scrutable User Modelling Shell for User-Adapted Interaction. Ph.D. Thesis, Basser Department of Computer Science, University of Sydney, Sydney, Australia.

Kelly, S., Jones M. (2001) Groupware and the Social Infrastructure of Communication, Communications of the ACM, **44** (12), 77-79.

Kobsa, A. (2001a) Generic User Modeling Systems, *User Modeling and User Adapted Interaction*, **11**, 49-63.

Kobsa, A. (2001b) Tailoring Privacy to Users' Needs (Invited Keynote). In M. Bauer, P. J. Gmytrasiewicz and J. Vassileva, (Eds.): *User Modeling 2001: 8th International Conference UM2001*. Berlin - Heidelberg: Springer Verlag, 303-313.

Kono, Y., Ikeda, M., Mizoguchi, R. (1994) THEMIS: a non-monotonic inductive student modeling system. *International Journal of AI and Education*, **5**(3), 371-413.

Kostuik K., Vassileva J. (1999) Free Market Control for a Multi-Agent Based Peer Help Environment, in Proceedings of the Workshop on Agents for Electronic Commerce and Managing the Internet-Enabled Supply Chain, held in association with the 3rd *International Conference on Autonomous Agents'99*, Seattle, May 1-5, 1999.

Maes, P. (1994) Agents that Reduce Work and Information Overload, *Communications of the ACM,* **37**(7), 31-40.

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

Mudgal, C., Vassileva, J. (2000) An Influence Diagram Model for Multi-Agent Negotiation, *Proceedings Cooperative Information Agents CIA'2000*, M. Klusch & L. Kerschberg (eds.) Springer-Verlag: Berlin Heidelberg, LNAI 1860, 107-118.

Munro, A., Hook, K., Benyon, D. (Eds.) (1999) Social Navigation of Information Space, Spirnger Verlag: Berlin-New York.

Paiva, A., Self. J. & Hartley, R. (1995) Externalising Learner Models, in J. Greer (ed), *Proceedings World Conference on AI in Education AIED'95*, AACE: Charlottesville, 509-516.

Raymond E. (1999) Homesteading the Noosphere, The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, O'Reilly & Associates. http://www.tuxedo.org/~esr/writings

Rich, E. (1983) Users are Individuals: Individualizing User Models, *International Journal of Man-Machine Studies*, **18**, 199-214.

Schreck J. (2001) Security and Privacy in User Modeling. Ph.D. Thesis, Dept. of Mathematics and Computer Science, University of Essen, Germany, 2001.

Seid Y., Kobsa A. (2000) Centralization vs. Decentralization Issues in Internet-Based Knowledge Management Systems: Experiences from Expert Recommender Systems. Workshop on Organizational and Technical Issues in the Tension Between Centralized and Decentralized Applications on the Internet (TWIST2000). University of California Software Institute, Irvine. http://www.ics.uci.edu/~kobsa/papers/2000-TWIST-kobsa.pdf

Suryadi, D., Gmytrasiewicz, P. (1999) Learning Models of Other Agents using Influence Diagrams, in *Proceedings 7th International Conference on User Modelling UM'99*, Springer: Wien-New York, 223-232.

Sycara, K. (1988) Resolving goal conflicts via Negotiation in *Proceedings 7th National Conference on Artificial Intelligence* (AAAI-88), St. Paul. MN, August 1988.

Terveen, L.G., and Hill, W.C. (1998) Evaluating Emergent Collaboration on the Web, in *Proceedings of the International Conference on Computer-Supported Collaborative Work CSCW'98* (Seattle WA, November 1998), ACM Press, 355-362.

Vassileva J. (1998) Goal-Based Autonomous Social Agents Supporting Adaptation and Teaching in a Distributed Environment, in Goetl B., Halff H., Redfield C., Shute V. (eds.) *Proceedings of Intelligent Tutoring Systems ITS'98*, San Antonio, LNCS No1452, Springer Verlag: Berlin pp.564-573

Vassileva J., J. Greer, G. McCalla, R. Deters, D. Zapata, C. Mudgal, S. Grant (1999) A Multi-Agent Approach to the Design of Peer-Help Environments, in S. Lajoie and M. Vivet (eds.) *Artificial Intelligence and Education*, IOS Press: Amsterdam, 38-45.

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*, Seattle WA, 139-149.

Zapata-Rivera, J.D. & Greer, J., (2000), Inspecting and Visualizing Distributed Bayesian Student Models, in G. Gauthier, C. Frasson & K. VanLehn (eds), *Intelligent Tutoring Systems, Proceedings ITS'2000*, Springer-Verlag, Berlin Heidelberg, LNSC 1839, 544-553.

Zheng, D., and Sycara, K. (1997) Benefits of Learning in Negotiation, in *Proceedings of Fifteenth National Conference on Artificial Intelligence*, 36-41.

Zlotkin, G., and Rosenschein, J. (1991). Cooperation and Conflict Resolution via Negotiation among Autonomous Agents in Non Cooperative Domains. *IEEE Transactions on Systems, Man and Cybernetics*, **21** (6): 1317-1333.

Authors' Biographies:

DR. JULITA VASSILEVA:

University of Saskatchewan, Dept. of Computer Science, 57 Campus Drive, Saskatoon, SK, CANADA S7N 5A9

Dr. J. Vassileva is Associate Professor of Computer Science at the University of Saskatchewan. She received her Ph.D. degree in Mathematics and Computer Science from the University of Sofia, Bulgaria in 1992. Dr. Vassileva has worked in the areas of artificial intelligence in education, adaptive hypertext and hypermedia. More recently, her research has focussed on ways to support virtual learning communities, on multi-agent systems, agent negotiation and coalition formation. She has authored over sixty technical papers and has co-edited two books. Dr. Vassileva was program co-chair of the 8th international conference on User Modelling in Sonthofen, Germany, 2001.

DR. GORD MCCALLA:

University of Saskatchewan, Dept. of Computer Science, 57 Campus Drive, Saskatoon , SK, CANADA S7N 5A9

Dr. G. McCalla is a Professor of Computer Science at the University of Saskatchwan. He received his M.Sc. from the University of Alberta and his Ph.D. from the University of British Columbia. Dr. McCalla's research interests are in applied artificial intelligence, in particular artificial intelligence in education, user and student modelling, agent-based computing environments, and other applications of AI. He is currently exploring "just-in-time" active learner modelling, intelligent group formation, agent modelling, culturally sensitive learning environments, and web mining for learner modelling.

DR. JIM GREER:
University of Saskatchewan, Department of Computer Science, 57 Campus Drive, Saskatoon, SK, CANADA S7N 5A9

Jim Greer is a Professor of Computer Science at the University of Saskatchewan and the Director of the ARIES Laboratory, a centre for research in intelligent educational systems. He received his Ph.D. from the University of Texas at Austin in 1987. His research has focused on artificial intelligence, intelligent tutoring systems, diagnostic reasoning, collaborative learning and peer help systems.