Distributed User Modelling for Universal Information Access

Julita Vassileva

Computer Science Department, University of Saskatchewan,
1C101 Engineering Bldg., 57 Campus Drive, Saskatoon, S7N 5A9, Canada

Abstract

In a distributed multi-agent based software environment, the traditional monolithic user model ceases to exist and is replaced by user model fragments, developed by the various software agents populating the environment. These model fragments have been developed in a variety of specific contexts to help achieve various goals. User models are thus fragmented, relativized, local, and often quite shallow. They are inherently inconsistent with one another and reflect not only characteristics of the users, but also certain social relationships among them. With the arising proliferation of models and data, the user modelling problem transforms into retrieval and integration of the available user model fragments "just in time" by a particular computational agent to the breadth and depth needed for a specific purpose. In this paper we explore the implications of distributed user models, drawing examples from I-Help, a collaborative system for peer help.

1. INTRODUCTION

The next generation of mobile, distributed, autonomous computer applications will allow future computing environments to be accessible from everywhere: not only on desk-top and lap-tops, but also on palm-tops, cell phones; these environments will be worn (data glasses, watches, etc.) and be embedded in everyday devices, home appliances, in the environment. People with various cultural background, goals, knowledge, and disabilities will be interacting with these mobile and ubiquitous computing environments in all imaginable contexts: in vehicles, in meetings, on public transport, while shopping, relaxing, eating, cooking etc… Meeting user needs adequately is crucial for future software applications, since the variety of contexts and devices in which software will be used implies a huge diversity of user needs to be met. The only way of achieving adaptation in distributed software environments is by designing systems that are no longer static stand-alone applications, but dynamic integrative environments that configure themselves according to the individual needs of the user, the context of use, and the platform requirements. Such systems should be able to operate independently of a person's location. In order to achieve such environments, the application functions have to be packaged into small atomic units or agents, which can re-assemble themselves dynamically depending on situational variables. Thus the atomic units will be reused in an appropriate way depending on the current hosting environment, e.g. uniformity and compatibility is achieved. These two qualities are important for two reasons: first, to allow the benefits for interaction and interoperability of software in the networked world, and second, to reduce the load on users learning how to operate a variety of software on different devices.

User modelling is crucial in order to track the context of the user. Due to the variety of contexts ensured by the mobility and ubiquity of the computing devices, a lot of information is needed to capture the aspects that are important for the user's needs, for example, about her exact location, state etc. The more information is available, the more adequate the user model and therefore, the better the adaptation of the functionality towards the individual user will be. Tracking the user's movement, location, state is especially important since the users are mobile.

We believe that conventional methods for software design even though appropriate for distributed applications design are not sufficient for applications supporting universal user access. These methods assume providing a fixed functionality defined at their design time independently of the conditions of their use, the experience or typical tasks of the user. There are techniques ensuring interoperability in distributed software, for example, CORBA, Jini and DCOM. However, with distributed objects, even though objects may run on different platforms, applications generally form a single monolithic entity of tightly bound objects, with hand-coded calls to known methods of pre-existing objects. The problem that the applications may require different functionality in a wide variety of contexts (platforms, locations, user needs etc.) cannot be addressed. There is research on methods allowing software applications to, for example, dynamically constrain the offered functionality when the parameters of the environment do not allow the provision of full functionality (Noble & Satayanaranyanan, 1999). For example, if a web browser is invoked on a laptop using a low-bandwidth network connection (modem or wireless), the browser should automatically adapt to ignore images that slow down the transfer of information. However, this adaptation is only with respect to the hardware and not with respect to the needs of the individual user. For example, for a particular task of the user it might be crucial to get a specific image, even though this image might not be important for 99% of the users. Providing applications with the capability to dynamically configure themselves, as proposed by Martin, Cheyer & Morgan (1999), but so that they meet best the needs of their users, according to the platform and context of use (e.g. location, task etc.) would lead to flexible self-adaptive software applications for universal access. The emerging environments that will dominate the software landscape of the future will be distributed, multi-user, ubiquitous, and mobile. There will be very few monolithic applications – the new applications will be inherently distributed and agent based. Negotiation between agents will become the way software applications function and interact with users; there will be no pre-defined at design time "method calls", but dynamically established links as result of negotiation among software agents. The functionality of the applications will emerge as a result of collaboration between software agents and users.

User modelling (UM) plays a crucial role in this type of environments. Continuous contact between users and ubiquitous information / communication technology allows for very fine-grained tracking of users' activities under different circumstances and by different modelling agents. 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 totally different information. Therefore, in a distributed multi-agent based software environment, the traditional user model is replaced by user model fragments, developed by the various software agents, populating the environment in a variety of specific contexts and with various goals.

2. AN EXAMPLE: THE I-HELP SYSEM

An example of such an environment is I-Help (Greer et al., 1998, Vassileva et al., 1999), a multi-agent system that allows users to request, receive and give peer help synchronously and asynchronously. I-Help is fully implemented and has been deployed in different versions with more than 1000 users. Ffor more information about evaluation of I-Help in these deployments, see Greer et al. (2001).

I-Help 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). Users are represented in I-Help by their personal agents; electronic resources or software applications are represented by application agents. A user request for help is sent by the user's personal agent to an agent broker who locates an appropriate other agent (either personal or application agent) using a database of models of the agent's resources. These resources can be the competence of user in the domain or the topic of material covered by an electronic material. I-Help introduces negotiation between agents and payment for help in terms of ICUs (I-Help Currency Units). On an agent-level the I-Help economy helps in regulating the supply and demand of human help resources. On a user-level, the market mechanism helps prevent overloading competent users with help-requests and motivates users to provide high quality help. In this function, it is similar to reputation / ranking based mechanisms used by Web sites like *slashdot.com*, *plastic.com* and *www.thewines.com*. In large multi-user environments, it cannot be expected that users will be intrinsically motivated to help other users. 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).

2.1 Adaptation in I-Help

There are various types of adaptations taking place in I-Help:

- Adaptation to the user's level of competence and specific help request.
- Adaptation of the number of help-requests directed to the user to his/her level of busy-ness, priorities at the moment, general preferences, and interpersonal relationships.
- Adaptation of the type of help-resource and the interface to the physical location of the user and the type of device being used, whether a browser on a PC, a hand held device, or a cell phone.
- Adaptation of the agent's appearance and basic behaviour to the user's wishes.
- Adaptation of the negotiation strategies deployed by the personal agents to the preferences of the user.
- Adaptation of the negotiation strategies deployed by the personal agents to the tactics of other users' agents.

Adaptation in multi-agent multi-user systems, like I-Help, happens all the time. Since there is no predefined behaviour to be "adapted", the behaviour of the system emerges depending on (1) the circumstances, (2) complex interrelationships between users and agents, (3) "just-in-time" generated models, and (4) decision-making techniques to judge the situation at hand and to calculate the utilities of the possible actions. Decision-theoretic approaches have become increasingly popular in the UM community recently (Bohnenberger et al., 2001; Jameson et al., 2000; Mudgal & Vassileva, 2000; Suryadi & Gmytrasiewicz, 1999). They require, however, a numeric or probabilistic representation of the situation (i.e. the user modelling features and other variables describing the context) and numeric evaluations of the payoffs of each possible action. These numbers are hard to invent; they need to be determined experimentally.

Adaptation within the multi-agent I-Help system is based on 1) models of users and other agents maintained by the personal agents, and 2) models of the involved software applications maintained by the application agents. The latter type of models (i.e. models of the resources provided by software applications: web-pages, discussion forum threads, etc.) is an important part of the adaptation of the system, since they are traded on the market by the application agents to satisfy users' needs. However, in this paper we focus on the user and agent models maintained by personal and diagnostic agents.

2.2. Agents Developing Models of Users and Other Agents

A multitude of user models is created by various agents and with various purposes. *Application agents* (agents of discussion forum threads, web pages, search engines) build their own *user profiles* representing features relevant to the context of the application, *based* on their interaction with the users and using "traditional" UM techniques.

Diagnostic agents (agents representing web-based test items, questionnaires etc.) represent a special type of application agents for a specific goal to create *user models in a particular area* of activity / knowledge and with a particular structure.

Personal agents (agents representing users), maintain user models containing private user characteristics. Examples of such characteristics are *lists with the person's friends and enemies*, *the user's preferences about negotiation* (how greedy/generous the personal agent should be in negotiation, the subjective importance for the user of certain resources like time or money), and the user's current goal. In addition, personal agents manage the set of competence models/profiles of the user for certain domains. Diagnostic agents create these domain specific user models on user's or personal agent's request. During negotiation with other agents, the personal agent acts as a representative of the user. The negotiation preferences and characteristics that control the agent's behaviour with other agents (for example, egoism, greediness, or generosity) are selected by the user. They reflect the way the user wishes to be perceived by the "world", therefore, indirectly, they represent also a kind of model of the user. During negotiation, the agents try to optimize their actions and to predict the opponent's actions. For this purpose, they create *models of the other agent's "character"* (priorities). Thus each personal agent models the character that the other user wants his/her agent to represent in the agent community.

Matchmaker / broker agents manage *databases of user models* (profiles) for a certain population of users; each broker is specialized to deal with models of certain user characteristics and to perform matchmaking for a specific purpose.

2.3. User Modelling Process: Making Sense of Fragments

During negotiation, the agents take into account the relationships between the users (if they exist), for example, by changing the negotiation strategy for friends or enemies (offering a discount or an extra high price). After repetitive 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.

The users are notified by their 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, it won't be able to earn virtual money. After a failed deal due to an undelivered resource (i.e. the user refused to help), the personal agent of the other user notifies the matchmaker. The matchmaker serves also as a "better business bureau": the personal agent, which breaks a deal, will gain bad reputation, and other agents will start avoiding it.

After a help session, the personal agent of each user presents to the user a brief questionnaire about the usefulness of the help session, and the perceived knowledge of the other party. This information is used to update the models of the users involved in the help session.

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 one monolithic 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). User and agent models are thus fragmented, relativized, local, and quite shallow. They are inherently inconsistent with one another and reflect not only characteristics of the users, but also certain social relationships among them. In addition, depending on who is modelling and who is being modelled in a distributed multi-agent environment, there can be agents modelling users, users modelling agents and agents modelling other agents. With the arising wealth of models and data, the user-modelling problem transforms into retrieval and integration of the available user model fragments "just in time" by a particular computational agent to the breadth and depth needed for a specific purpose. Thus, the need for integrating user model fragments grows in importance, and the ideal of maintaining a single monolithic user model is less desirable (and likely intractable).

3. DISCUSSION: DISTRIBUTED USER MODELLING ISSUES

With distributed applications/environments user modelling becomes the process of assembling and summarizing fragmented user information from potentially diverse sources. The key to making sense of this widely distributed information is the ability to interpret multi-modal information from multiple heterogeneous relevant sources and to integrate this information into a user model of appropriate granularity. The focus is shifted from the model itself to the process of modelling. The model is 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). This introduces many new requirements for the user modelling process. The main questions boil down to how to manage all this information:

- How does one locate the agent who has a relevant model given the context and the purpose for which the model is needed?
- How does one make sense of possibly inconsistent and even contradictory data?
- In general, how does one interpret models created by other agents?
- How does one ensure persistency and integrity of user models in such an environment? Can it be expected at all?
- How to ensure privacy?

New techniques will be required in order to carry out distributed, just in time integration of user models, relying heavily on modelling interpersonal relationships, decision making. Changes in the traditional methods for system evaluation will be necessary, since currently there is a lack of methodologies for evaluation of systems with emerging behaviour.

Many privacy and security issues arise in such environments that must be tackled. The idea of delegating the responsibility of user modelling to autonomous (and even worse, economically motivated) agents can be worrisome. Are agents guaranteed to serve their users’ best interests? Of course, if agents maintain only distributed context-loaded profiles, a full integration of the information by the “Big Brother” would be so expensive that it would be practically impossible. However, integration of specific data, with a specific purpose would be possible. Even if we trust the personal agents not to reveal some particular data, how do we know which piece of data may be critically integrated with other data and used against us?. The police warning “Anything you say can be used against you” will mean, “any information revealed by you or your agent can potentially be used against you”. In this case, the role of personal agents should become similar to the role of lawyers protecting the interests of their users. However, this requires a level of intelligent reasoning of the agents that is unlikely to be feasible in the near future. And even if users instruct their agents not to reveal *any* information about them (at the cost of much functionality and many potential benefits of multi-agent multi-user modelling), they can’t prevent other agents from modelling their behaviour, just as people cannot prevent other people from observing their actions, making conclusions and thinking certain things about them. Any person and agent who has encountered the user or his/her agent will be able to develop a model of him / her, which it may give to a third agent or user on request. So “universal access” to information becomes “universal transparency”, which makes every user potentially vulnerable.

Another problem is how to avoid spreading negative rumours and ill-informed gossip among agents, with the attendant risk of isolating agents or users from certain services (Bicknell, 2000). Reputation networks and rumours are necessary protection mechanisms in a multi-agent community, but how is it possible to avoid their misuse? Active research in I-Help into these issues is now underway (Winter, 1999).

So the privacy issues of this new type of multi agent / user modelling are many and deep. On the other hand, we are witnessing the appearance of inherently distributed applications, like Gnutella (Gnutella, 2000), and Mojo (Mojonation, 2000), allowing storage of data or computation to happen on demand on any machine in a distributed environment without an all-knowing centralized component, depending completely on the patterns of computer usage of the participants. It seems that the idea of revealing the possibly private information of computer usage and even providing their “sacred” computers with all their data to unknown third parties is not so undesirable for users, when it is not done by a centralized institutions and when it is based on an economic rewarding mechanism. We believe that, as this example shows, people will readily give some of their privacy in exchange for the convenience and benefits provided by the new technology and highly adaptive environment. Yet, these issues have to be tackled, since this technology, as any powerful technology, can be easily and badly misused.

4. CONCLUSION

These revised ideas about user modelling will shift the user modelling research agenda. Processes such as retrieval, aggregation, and interpretation will be much more important than they have been. Many very interesting research issues surrounding these techniques will have to be explored. 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 I-Help experiments have already shown, it will not be necessary to resolve all of these issues in order to usefully user model.

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

Bicknell, C. (2000) Anti-Fraud That's Anti-Consumer, Wired News, July 24, 2000, available on line at: http://www.wired.com/news.print/0,1294,37642,00.htm

Bohnenberger, T., & Jameson, A. (2001) When Policies Are Better Than Plans: Decision-Theoretic Planning of Recommendation Sequences In J. Lester (Ed.), *IUI2001: International Conference on Intelligent User Interfaces. New York: ACM.*

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. In Goetl B., Half H., Redfield C., Shute V. (Eds.) *Intelligent Tutoring Systems: Proceedings ITS'98*. LNCS No1452. (pp.494-503). Springer Verlag: Berlin.

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, to appear in Moore J (Ed.) *AI and Education: Proceedings of AIED'2001*, San Antonio, Texas, May 19-24, 2001.

Gnutella, 2000; available on line at: http://gnutella.wego.com/

Jameson, A., Großmann-Hutter, B., March, L., Rummer, R., Bohnenberger, T., & Wittig, F. (2000) When Actions Have Consequences: Empirically Based Decision Making for Intelligent User Interfaces. *Knowledge-Based Systems*, 13. In press.

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.

Martin D., Cheyer A., Moran D. (1999). The Open Agent Architecture: A Framework for Building Distributed Software Systems. *Applied Artificial Intelligence.* 13, 91-128.

Mojonation (2000): available on line at: http://www.mojonation.com/

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

Noble B.D., Satayanaranyanan M. (1999) Experience with adaptive mobile applicatons in Odyssey, *Mobile Networks and Applications*, 4, 4, 245-255.

Suryadi, D., Gmytasiewicz, P. (1999) Learning Models of Other Agents using Influence Diagrams, in Kay J. (Ed.) *User Modelling: Proceedings of the 7th International Conference on User Modelling.* (pp. 223-232). Springer: Wien-New York.

Vassileva J., Greer J., McCalla G., Deters R., Zapata D., Mudgal C., Grant S. (1999) A Multi-Agent Approach to the Design of Peer-Help Environments. In Lajoie S. and Vivet M. (Eds.) *Artificial Intelligence and Education: Proceedings of AIED'99.* (pp.38-45). IOS Press: Amsterdam.

Winter, M. (1999) The Role of Trust and Security Mechanisms in an Agent-Based Peer-Help Environment, , *Proceedings of the Workshop on Deception, Trust, and Fraud in Agent Societies* (pp. 139-149), associated with Autonomous Agents '99, Seattle WA,.