Purpose-based User Modelling in a Multi-agent Portfolio Management System*

Xiaolin Niu, Gordon McCalla, Julita Vassileva

Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan, S7N 5A9 Canada
{xin978, mccalla, jiv}@mail.usask.ca

Abstract. This poster outlines a new approach for decentralized user modelling using a taxonomy of *purposes* that define a variety of context-dependent user modelling processes rather than creating and maintaining a single centralized user modelling server. This approach can be useful in distributed environments where autonomous agents develop user models independently and do not necessarily adhere to a common representation scheme.

1 Introduction

Traditionally user modelling has focused on creating and maintaining a single global description of the user used internally in an application for some purpose defined at design time [1]. Knowledge representation is a key issue in this kind of traditional user modelling. With the emergence of networked applications, user modelling servers have been proposed [3]. User modelling servers provide a centralized solution: even if the user data comes from and serves various applications, the representation of the user model follows a particular centralized schema, which is known in advance to the applications.

However, in a multi-agent based software environment, a single user model is typically replaced by user model fragments, developed for particular purposes and contexts by the various autonomous software agents populating the environment [7, 9]. These fragments cannot be expected to share the same representation scheme (the same problem arises in distributed databases, see [4]). Therefore the focus of user modelling shifts from the collection at one place of as many data about a user as possible to collecting on demand whatever user information is available at this moment from various agents and interpreting it for a particular purpose. This is called *active* user modelling [7].

Our goal is to develop a methodology for active user modeling and a library of purpose clichés for active user modelling. The representation of each purpose is procedural and contains a description of the context in which the procedure can be applied to achieve the purpose. The purposes are retrieved and executed by distributed

* To appear in Proceedings UM'2003, Johnstown, PA, June 22-26,2003.

autonomous agents to compute user models "just in time" [6] as they are needed. Similarly to developing a full ontology of a domain, envisaging all possible purposes for user modelling in all possible contexts is an impossible task. Therefore, the effort of the designer should focus on creating a library of important, reusable purpose clichés.

2 Purposes in a Multi-agent Portfolio Management System

A multi-agent portfolio management system [8] serves as a domain for our investigations into purpose clichés. There are two kinds of agents: personal agents (PA) represent investors who need advice, and expert agents (EA) who provide that advice. A main purpose for a PA is to find an appropriate EA for a given investor, and for this models of both the investor and the EA are needed. There are many other typical purposes in this domain.

Each purpose consists of inputs, functions and outputs. The inputs denote the type of raw data that is relevant to the given purpose. The functions are algorithms used to compute the desired outputs using the inputs within context-specific resource constraints. The outputs are the result of this computation, and can be considered to be context-specific partial user/agent models. These partial models can also form input to other purposes. The purposes can be organized into *hierarchies* with respect to *generalization* and *aggregation* (similar to plans produced using hierarchical planning [2]).

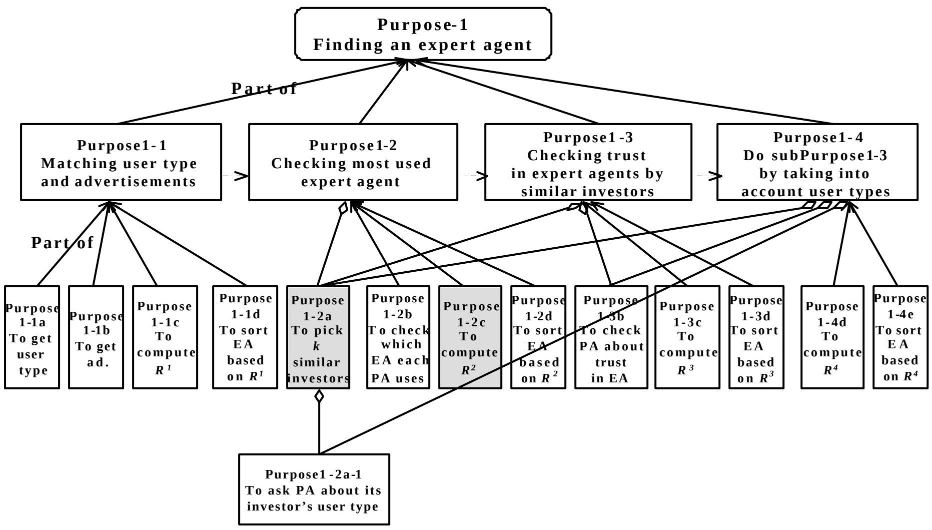

Fig. 1. The aggregation hierarchy for *purpose-1*, finding an expert agent

In a generalization hierarchy, the specific purposes inherit information and procedures from more general purposes in the hierarchy. For example, one specific purpose in the stock investment domain is *Purpose-1*, which is to select an appropriate expert agent for an investor. The higher-level purpose might be to select an agent (not necessarily an expert agent) to match the needs of a person (not necessarily an investor). Two more specific purposes might be to find an expert agent for an investor who is already retired and to find an expert agent for an investor who is a student.

Purposes can also aggregate sub-purposes, resulting in aggregation hierarchies where some sub-purposes can be part of several super purposes. The way sub-purposes are aggregated can be defined by the functions of the super purpose. For example, *Purpose-1* matches the model of the investor and the EA by gradually integrating user/agent model fragments by consecutively aggregating the results of 4 sub-purposes (as shown in Fig.1). The 4 sub-purposes of *Purpose-1* can be called in one by one depending on resource and time availability. The algorithm in *Purpose-1* can be stopped at any time and will generate the best answer so far (depending on which of the sub-purposes were executed). However, the sub-purposes of *Purpose1-1* (and the other second level purposes) do not have this anytime aspect, since they require fully completing each sub-purpose.

Consider one of the sub-purposes of *Purpose1-2*, ***Purpose1-2c***: to calculate the rating R^2 of each expert agent. This purpose calculates the rating of each expert agent based on what proportion of a given group of personal agents uses the expert agent. Let's denote with $ß_E \in [0,1]$ the evidence of usage. For example, if EA_1 (Expert Agent 1) is used by 75% of the personal agents, then $ß_1$ is 0.75. Therefore, the rating R^2 for each expert agent so far can be derived using the rating computed by *Purpose1-1* (*Purpose1-1* computes the rating of each expert agent based on the difference between the advertisement of the expert agent and user type of the investor), i.e. $R^1{}_E$ and the new evidence $ß_E$. The simple reinforcement learning formula is:

$$R^2{}_E = \varepsilon R^1{}_E + (1 - \varepsilon)\, ß_E$$

where $0 = \varepsilon = 1$ is a coefficient which denotes how much the agent values the new evidence $ß_E$. The output from *Purpose1-2c* is a vector $[R^2{}_1 \ldots R^2{}_n]$, where $R^2{}_i$ denotes the rating of the i^{th} expert agent.

3 Purpose Re-use

A library of purposes forms a repository of clichés, which can be plugged into any application where there is a correspondence between the application's needs and the reuse purposes. There are several ways to reuse the purposes:

- *Generalization*: A purpose can be generalized into a higher-level purpose, which can be used in different domains.
- *Specialization*: A purpose can be specialized into a more specific purpose by specifying more constraints in additional sources of information that can be used in a specific context.

- *Modification*: A purpose can be modified in order to adapt to a new domain where the available input data are of different types. For example, *Purpose-1* could be adapted for the domain of peer help [5] to choose a helper for a student who needs some help.
- *Sharing a purpose*: A sub-purpose can be shared in aggregation by several super purposes. For example, *Purpose1-2a* is to pick the personal agents of investors who are of similar user type. This purpose can be re-used by three super purposes. Once this purpose is created, designer effort is saved when other super purposes re-use this sub-purpose.

Purpose re-use is valuable from a software engineering point of view and critical to the active user modelling approach. The basic motivation of re-using purposes is to save time and effort. If an existing solution can be reused, this saves time that would otherwise be spent on the creation of similar or identical software components. Another motivation is flexibility in response to new requirements. Purposes can be selected and tailored to the designer's needs by changing some parameters, such as inputs and context information, etc. A library of purposes could thus be designed to provide the clichés that makes the engineering of a new system much easier.

4 Possible multi-agent system architectures

There are several options to place the purpose hierarchies within a multi-agent system architecture:

- *On board of each agent.* This, however, can result in complex and "fat" agents leading to scalability and performance issues.
- *On a centralized "server".* When an agent needs a particular purpose, it fetches it from the server. This contrasts with centralized user modelling where the server collects and computes the user modelling data and the applications are clients, which receive and use the data. In this case, the data is kept by the personal agents. Each agent retrieves from the server the purpose that is relevant to its user modelling need at the moment and uses it to compute the new user model data. The computation and data are distributed, only a library of purposes is centralized.
- *On specialized user modeling (UM) agents associated with each purpose.* Distributed application agents subcontract UM tasks to be done by specifically-tasked UM purpose agents. Each of these specialized UM purpose agents is like a UM server, but strongly specialized. Such agents don't store any data, but perform computation upon request.

Which of these options is chosen depends on the application and the designer. Important criteria are how complex the purpose hierarchies are, how often computation is needed, how much communication is involved (influencing network traffic, performance, response time), and how important is privacy.

5 Conclusions and Future Work

We are currently implementing a comprehensive purpose hierarchy to support active user modelling in the portfolio management system. The quality of decisions made by the agents will be evaluated with simulated users. However, demonstrating that user modelling helps achieve better decisions, is only a "proof of solution existence", and doesn't show the advantages of active user modelling versus centralized modeling. The strongest argument for the active approach is that it implies less constraint on the agents (with respect to shared representation scheme, reliance on a connection to a server etc.) and is more robust (no central point of failure). We feel that the weakest point for the active approach is the practicality of developing comprehensive reusable purpose hierarchies. Our hope is that in the future, much as ontology research is leading to comprehensive shared vocabularies for many domains, over time a set of overlapping user modelling purpose cliché hierarchies will be devised for many domains and will be used to carry out active user modelling by heterogeneous software agents.

References

1. Browne, D., Totterdell, P., Norman, M.. Adaptive user interfaces. Academic Press Ltd., London, UK, 1990.
2. Corkill, D. *Hierarchical Planning in a Distributed Environment*. In Proceedings of the Sixth International Joint Conference on Artificial Intelligence, pp. 168-175, 1979.
3. Fink, J., and Kobsa, A. 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) (2000), 209-249
4. Giunchiglia, F., Zaihrayeu, I. Making peer databases interact - a vision for an architecture supporting data coordination // Technical Report # DIT-02-0012. Also to appear in Proc. Cooperative Information Agents (CIA 2002), Madrid, September 2002
5. Greer, J., McCalla, G., Cooke, J., Collins, J., Kumar, V., Bishop, A. and Vassileva, J. The Intelligent HelpDesk: Supporting Peer Help in a University Course. In Proc. ITS'98, San Antonio, Texas, LNCS No1452, Springer (1998), 494-503.
6. Kay, J. A scrutable user modeling shell for user-adapted interaction. Ph.D. Thesis. Baser Department of Computer Science, University of Sydney, Sydney, Australia. 1999.
7. McCalla, G., Vassileva, J., Greer, J. and Bull, S. Active Learner Modeling. Proc. ITS2000, Springer LNCS 1839, (2000) 53-62.
8. Tang, T., Winoto, P. & Niu, X. Who Can I Trust? Investigating Trust between Users and Agents in a Multi-agent Portfolio Management System. In AAAI-2002 Workshop on Autonomy, Delegation, and Control: From Inter-agent to Groups. Edmonton, Canada, July 28, 2002.
9. Vassileva, J., McCalla, G., Greer, J. (to appear in 2003) Multi-Agent Multi-User Modelling, to appear in User Modelling and User-Adapted Interaction, 28 pp manuscript