

Purpose-based Expert Finding in a Portfolio Management System

Xiaolin Niu, Gordon McCalla and Julita Vassileva¹

Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan, S7N 5A9 Canada
Telephone: +1 306 966-4886
Facsimile: +1 306 966-4884
{xin978, mccalla, jiv}@cs.usask.ca

Abstract. Most of the research in the area of expert finding focuses on creating and maintaining centralized directories of experts' profiles, which users can search on demand. However, in a distributed multi-agent based software environment, the autonomous agents are free to develop expert models or model fragments for their own purposes and from their viewpoints. Therefore the focus of expert finding is shifting from the collection at one place as much data about an expert as possible to accessing on demand from various agents whatever user information is available at the moment and interpreting it for a particular purpose. This paper outlines purpose-based expert modelling as an approach for finding an expert in a multi-agent portfolio management system in which autonomous agents develop expert agent models independently and do not adhere to a common representation scheme. This approach aims to develop taxonomy of purposes that define a variety of context-dependent user modelling processes which are used by the users' personal agents to find appropriate expert agents to advise users on investing strategies.

Keywords:

Purposes, expert finding, expert modelling, just-in-time user modelling, anytime, decentralized.

1 Introduction

Expert finding has been investigated extensively by the software agent community. Most of the research concerns building profiles of experts, which users can search on demand (Ackerman et al., 1999; Foner, 1997; McDonald & Ackerman, 2000; Vivacqua, 1999). The effort spent on constructing and maintaining profile information

¹ This work is supported by the Natural Sciences and Engineering Research Council of Canada.

Xiaolin Niu, Gordon McCalla and Julita Vassileva

of experts is significant and expensive (Fink & Kobsa, 2000) since the existing centralized methods rely on often-controversial information, coming from clients who may be satisfied or dissatisfied with the services of an expert because they have different preferences and criteria. Moreover, in the real world, the experts may advertise exaggerated capabilities in order to attract more clients. We have chosen to tackle these problems. To introduce our approach, let us first look at an example scenario.

An example scenario

Suppose Bank A offers customers a 24-hour on-line portfolio management service (investment in stocks, bonds, mutual funds, etc.) where a software agent represents each customer to invest/manage his/her portfolio. It is in the best interest of each broker agent to exaggerate its expertise when it advertises its abilities, in order to attract as many clients as possible. In order to decide which broker agent to choose, the investor might want to investigate the expertise of a broker agent by asking (through their agents) other users who have had contact with the broker agent. In this way, customers can get more realistic information about the real capabilities of the broker agents.

This example highlights an important issue related to expert finding: constructing expertise models based on the distributed information kept by a variety of peers in the environment. In a decentralized environment autonomous agents keep expert model fragments, which are private users' evaluations of the expert's performance and can be used by others only if the agents are willing to share the information. These fragments cannot be expected to adhere to the same representation scheme since there is no centralized authority to impose such a scheme. The same problem arises in distributed databases; see (Giunchiglia & Zaihrayeu 2002). Therefore, a centralized expert model (Browne et al, 1990), (Kobsa & Fink, 2003) is not always feasible. Even if the agents are willing to entrust their private models to a centralized database, the expert model fragments come from a range of sources (e.g. raw data, other agents) and are dependent on the context in which they were created (for what purpose, when and who created it), so it would be very hard to ensure consistency in a centralized expert model representation based on these fragments as input. However, it would be possible to collect and integrate specific data, which is relevant and useful to a specific purpose, i.e. to compute an expert model at the moment and for the specific purpose it is needed. Thus, the main focus of expert modelling shifts from traditional representation issues to issues such as determining what knowledge to retrieve for a given purpose and making sense of this knowledge in context. Thus expert modelling in a decentralized system consisting of autonomous agents is "active" and "distributed" just like the active distributed user modelling approach, proposed in (McCalla et al., 2000; Vassileva et al., 1999; Vassileva et al., 2003). This process happens "just in time" (Kay, 1999) and is invoked as a part of achieving a particular purpose and using information relevant to that purpose. An example of how the active user modelling approach is implemented in the domain of peer-help for university students can be found in (Bull et al., 2001 and Vassileva et al., 2003).

This paper describes an approach for expert finding which is based on defining taxonomy of expert modelling purposes. This will allow retrieving expert information relevant to a particular purpose in order to assemble and integrate fragmented expert

model information. The representation of each purpose is procedural and contains a description of the context (e.g. which agents are available at the moment to provide information, what kind of user models they can provide and how much time is available for computation) in which the procedure can be applied to achieve the purpose. The purposes are retrieved and executed by distributed autonomous agents to compute user models “just in time” as they are needed. Depending on the availability of resources (e.g. other agents able to provide an expert model or user model fragments, time for computation etc.) more purposes may apply to the situation and the quality of the results can be improved. Similarly to developing a full ontology that can represent all aspects of a domain, envisaging all possible purposes for user modelling in all possible contexts is a daunting task. Therefore, the effort of the designer in our approach should focus on creating a library of important, reusable purpose clichés.

The rest of this paper reports our work on developing purposes in the portfolio management expert finding domain and our proof of concept evaluation of our approach. We begin with a brief overview of the system architecture in section 2. Then section 3 describes the nature of a purpose, the purpose hierarchies and algorithms. Our arguments for purpose reuse are outlined in section 4. In section 5 we present the system architecture and the results of some experiments. Finally, we conclude this paper by pointing out future research directions.

2 Multi-agent Portfolio Management System

The portfolio management system has a multi-agent architecture as shown in Fig. 1 based on the system presented in (Tang et al. 2002). There are two kinds of agents: personal agents (PAs) and expert agents (EAs). Each investor has his/her own personal agent. The PA collects an investor’s risk-return preferences and other characteristics through questionnaires and game playing (Tang et al. 2002). The PA also needs to hire the most suitable expert agent for its investor to manage assets on the investor’s behalf.

The expert agent can choose a sequence of portfolios over time to achieve a measure of performance that is appropriate to the risk-return preferences of an investor. Each EA has a knowledge base that stores different strategies for different risk-return user types. Each EA also has a risk-return preference, thus determining the kind of user appropriate for the EA. The EA can broadcast its advertisement, for example, “I am a good expert for a risk-seeking person!” or “I am an expert for a risk-averse person!”

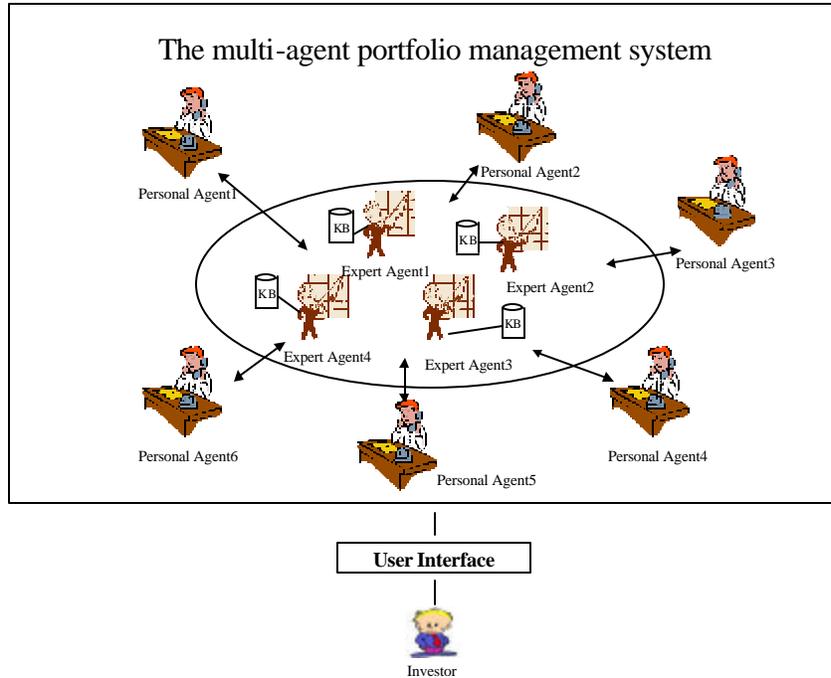


Fig.1. The architecture of the multi-agent portfolio management system

An example scenario is described below. An investor wants to have a diversified portfolio but lacks time for active investment. He/she believes the on-line brokers will possess superior information and uses a PA to manage his/her assets. First the PA will collect personal information about the investor through questionnaires and game playing and decide the risk return preference of the investor. Then the PA will choose the most appropriate EA for this investor based upon his/her risk-return preference, the advertisement of the EAs and the evaluation about the EAs from other PAs who have been hired by the investors who are friends of the investor or have similar risk-return preferences. After a period of investment, an evaluation form needs to be filled out by the investor to evaluate the performance of the EA. The result will be stored in the PA for further evaluation needs. From this example, therefore, the user models, containing characteristics of an investor such as risk-return preference of an investor and investor's trust in an expert agent, are created and maintained by each autonomous personal agent, i.e. there is no single user model associated with the individual expert agent. Rather there are many different "snapshots" of one expert agent taken by different agents in different contexts. The user and agent models are not only physically distributed throughout the system, but also logically decentralized since there is no standard representation scheme for these models (personal agents may be developed by different providers and are heterogeneous). Agents

communicate through messages using a shared ontology and protocol, but have no access to each other's individual representations (even if they had access, they wouldn't be able to interpret them). It is necessary to allow agents to request from each other and integrate decentralized model fragments stored privately by each of them and to generate a desired expertise model. The next section, describes the purpose-based user/agent modelling approach.

3 Purpose-based User / Agent Modelling Approach

Purpose-based user/agent modelling encompasses processes of information retrieval, integration and interpretation of model fragments. Purpose-based means that the distributed and fragmented models can be computed just-in-time, for a particular *purpose*, using only the data required for that purpose. The purposes are organized at different levels of granularity and are associated with routines for retrieval, integration and usage of the user/agent modelling data.

A purpose contains three kinds of information: inputs, functions and outputs. The inputs denote the type of raw data (describing context and domain variables), which 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 results of computation and can be considered to be context-specific partial user/agent models. These partial models can also form input to other purposes.

An important feature of purposes is that they can be organized into *hierarchies*. The library of purposes can be viewed at many levels, for example, from very general to very specific. There are two ways to organize purposes: *generalization* and *aggregation* (similar to plans produced using hierarchical planning (Corkill, 1979)).

In a generalization hierarchy (e.g. as in Fig. 2), the specific purposes inherit information and procedures from more general purposes in the hierarchy. Identifying higher-level purposes and lower-level purposes is of value in a domain because their presence allows us to understand concepts in more refined or more general terms. Moreover, it improves comprehension and leads to reduction in repeated information. For example, one specific purpose in the stock investment domain is *Purpose-1*, which is to select an appropriate expert agent for an investor. A higher-level purpose is to select an agent (not necessarily an expert agent) to match the needs of a person (not necessarily an investor). Two more specific purposes are to find an expert agent for a risk-seeking investor and to find an expert agent for a risk-reverse investor (Fig. 2).

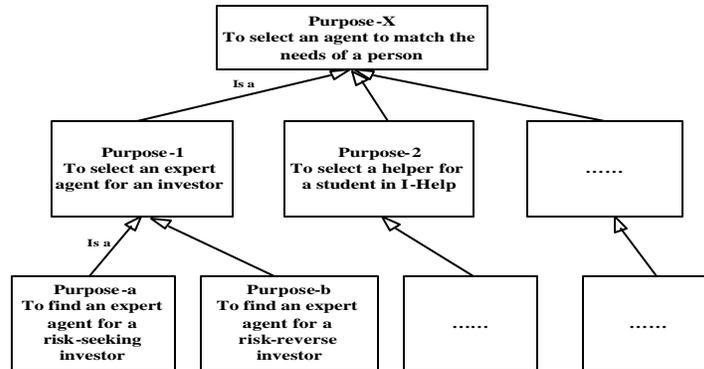


Fig. 2. An example of generalization hierarchy

Purposes can also aggregate sub-purposes; resulting in aggregation hierarchies (e.g. Fig. 3) where some sub-purposes can be part of several super purposes (the combination of abstraction and aggregation is similar to the granularity hierarchies of McCalla and Greer, 1994).

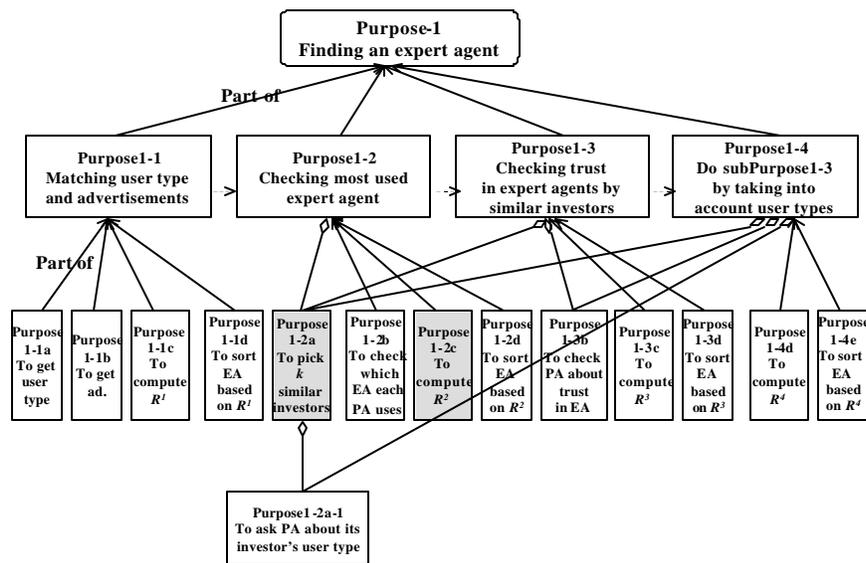


Fig. 3. The aggregation hierarchy for Purpose-1, finding an expert agent

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 consecutively integrating user/agent model fragments computed as results of 4 sub-purposes (as shown in Fig.3). The four 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 produce the best answer computed so far (depending on which of the sub-purposes were executed). This is similar to an *anytime* algorithm (Boddy & Dean, 1989). However, the sub-purposes of *Purpose-1-1* (and the other second level purposes) do not have this anytime aspect; rather the algorithms in them are *all-or-nothing* since they require fully completing each sub-purpose and solutions cannot be returned unless the agents have completely finished computing. Consider one of the sub-purposes of *Purpose-1-2*, ***Purpose-1-2c***: 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 $\beta_E \in [0,1]$ the evidence of usage. For example, if EA₁ (Expert Agent 1) is used by 75% of the personal agents, then β_1 is 0.75. Therefore, the rating R^2 for each expert agent so far can be derived using the rating computed by *Purpose-1-1* (which 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 β_E . A simple reinforcement learning formula can be used to compute the new rating in each sub-purpose, using the rating computed by the previously executed sub-purpose and the new evidence (or resource) available.

$$R^2_E = \varepsilon R^1_E + (1 - \varepsilon) \beta_E$$

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

In this way sub-purposes can be super-imposed, each starting with the rating produced by the previously executed sub-purpose and the new evidence specific for the sub-purpose. The order of execution of purposes can make a difference and therefore it is important to start with sub-purposes that consume less resources and bring significant improvement in the computed rating and to leave sub-purposes that bring smaller improvements or require more time or input data to be executed later, only if there are available resources (evidence and computational time).

4 Purpose Re-use

Purposes defined in this way form a repository of clichés, which can be reused to expand the range of purposes for the application or for different applications. 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 (as in Fig. 2).

Xiaolin Niu, Gordon McCalla and Julita Vassileva

- *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 (as in Fig. 2).
- *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, *Purpose1* could be adapted for the domain of peer help (Greer et al., 1998; Vassileva et al., 1999) 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 (e.g. risk-taking, moderate or risk averse). This purpose can be re-used by three super purposes (as in Fig. 3). 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 to save time and effort (Sutcliffe, 2002). 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 by the designer and tailored to the specific needs of the application 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.

5 Multi-agent system architecture and experiments

This section describes the implementation of the proposed purpose-based user modelling approach that can serve as a “proof of solution existence” and some of the experiments that we carried out to show that this user modelling approach can support adaptation in terms of selecting appropriate expert for the user.

5.1 Multi-agent system architecture

This multi-agent system is built in JADE (Java Agent Development Framework) – a software framework aimed at developing multi-agent systems and applications. The purpose hierarchies within this system architecture are maintained by a set of specialized user/agent modelling agents associated with each purpose. These agents are organized according to the purpose hierarchies. Each user/agent modelling agent asks the next (according to the aggregation or to the generalization dimension in the purpose hierarchy) agent to continue the computation needed to achieve the appropriate sub-purpose. Fig. 4 shows the system architecture for Purpose1. Personal or application agents subcontract user/agent modelling tasks to these specialized user/agent modelling agents, which perform computations upon request and return the results to the requesting agent without storing any data.

In this way, the computation of user models and the storage of user data in this architecture are fully distributed. Specialized purpose agents can be reused easily.

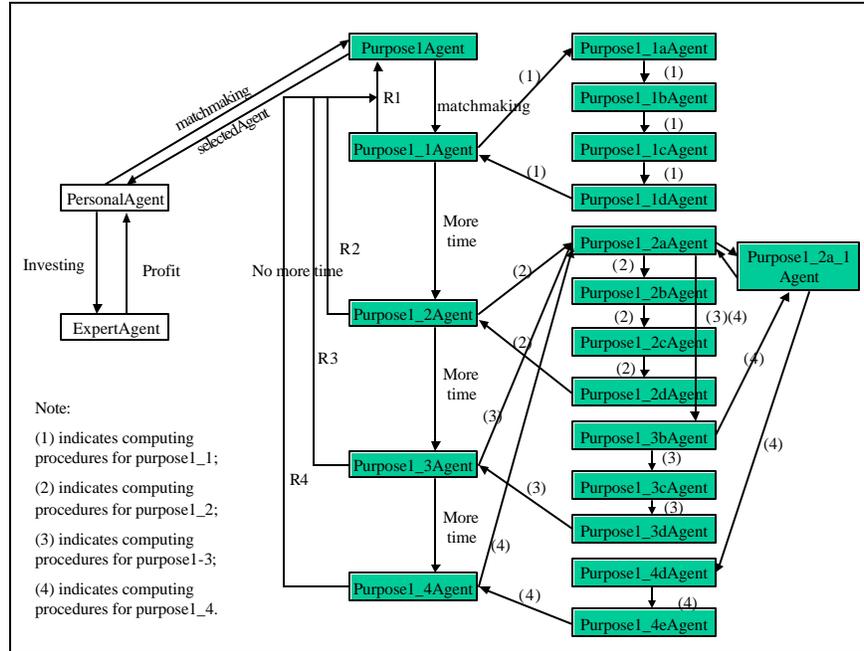


Fig. 4. The multi-agent architecture

5.2 Experiments and Results

In this section we will discuss experiments we have run to evaluate our purpose-based active modelling system. The system is hard to evaluate for several reasons. First, it is hard to evaluate the advantage of active/decentralized expert finding versus centralized expert finding with respect to software engineering, since it will require building the same system using a centralized user data repository and expert profile server. Such a system may turn out to be easier to build and will probably be more efficient. However, this is an unfair comparison since a centralized representation will require that all agents reveal their expert models, trust values and etc. to the central component, which leads to problems with privacy. This will be also hard to impose in the real world, especially, if the agents are heterogeneous and use different formats of representing user information. We leave the formal evaluation of the software engineering aspects of our approach to future research after we have gained more insight into the approach, and rely on arguments about the advantages of designing and building systems with a distributed, localized architecture (see the conclusion). However, we do want to provide a “proof of concept” demonstration that a system with a reasonably wide functionality can be built using an existing set of purpose hierarchies, and that these purposes can be reused for many goals.

Xiaolin Niu, Gordon McCalla and Julita Vassileva

It is also premature to evaluate our approach with real users until we are able to establish a performance baseline. Such a baseline will shed light on issues such as the effect of various distributions of expert agents and user types, the impact of various kinds of misrepresentation where advertisements don't match actual capabilities, and the capabilities for enhanced decision making of the anytime algorithms at the heart of the purpose-based approach. Once this baseline is established, variability and idiosyncrasies in user performances can be understood against the backdrop of the baseline. The goal of our experiments, then, is to establish such a performance baseline, and this can most easily be accomplished with simulated users. The system is thus evaluated with simulated users to determine how well the agents can make decisions using the purposes and the multiple sources of user data. We ran forty (40) personal agents and four (4) expert agents for each experiment. The criterion for success is based on a comparison of the features of the simulated user with the capabilities of the chosen expert agent (which may be different from the advertisement). This criterion is motivated by the observation that if, as a result of the system's expert finding, real users hire experts whose capabilities are a better match to their features, this means that the purpose-based expert finding would bring improved investment decisions and lead to less deception.

Experiment 1: How the quality of decisions improves with increasing the number of available agents

In the first experiment, each expert agent is assigned a different risk-return preference (from risk-seeking to risk-averse) and a neutral advertisement. The variable for this experiment is the number of personal agents who are currently available. Four sets of data were produced by running 10 personal agents, 20 personal agents, 30 personal agents and 40 personal agents respectively (each set is shown with different line in Fig. 5). The vertical axis represents the average difference between the risk return preference of the simulated user and the selected expert agent. This difference varies from 0 to 1, since both the users' and experts' risk return preferences are in the range [0, 1]. A large difference means a bigger average mismatch between the users' risk return preference (i.e. user profile) and the selected expert's ability. The horizontal axis represents the four sub purposes of Purpose-1 (see Fig. 3). The single line, e.g. for 10 personal agents, shows that the distance between the characteristics of the simulated user and the selected expert agent is getting smaller when more sub-purposes are executed, i.e. when more evidence is available. One can also see that the quality of the matchmaking improves when more personal agents are available (the drop of the line representing the setup with 40 agents is bigger, and the average distance between the user profile and the real expert agent capability is with about 0.06 smaller).

This result is not very surprising. Developers of real-world recommender systems have also found that the quality of recommendations improves with the number of recommendation in the systems, i.e. with the number of users who enter recommendations (Schafer, Konstan & Riedl, 2000).

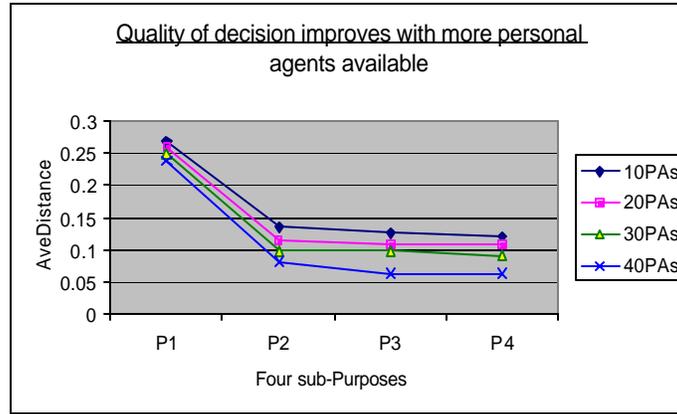


Fig. 5. The quality of expert selection improves with more agents available and resources available (when more purposes are executed).

Experiment 2: Exploring the performance in case of experts' deception

The second experiment explores aspects of the experts' characteristics-advertisement distribution. Here the total number of personal agents is fixed to be 40. By varying the value of experts' real risk attitudes and advertisements, the experiment shows that the purpose-based modelling computations bring improvement over a broad range of characteristics and advertisements. The baseline experimental condition is when the advertisement perfectly matches the real risk attitude of the expert, i.e. there is no expert deception (see Fig. 6). Fig.7 shows the results for this condition. We can see that the result from Purpose1-1 is better than the result after executing Purpose1-2. This is because Purpose 1-1 determines the match based on the expert's advertisement, which in the baseline condition means a perfect match. Further computation (such as in Purpose 1-2 which takes into account other users' attitudes) can only make worse the perfect match. This baseline condition, however, is not realistic, since expert agents have an incentive to misrepresent their capabilities to attract more customers, as already discussed in the introduction.

The opposite extreme condition that was tested in Experiment 2 is a total mismatch in the advertisement and real risk-attitude of the expert agent, or total deception (see Fig. 8). The results of this experiment (Fig. 9) show that the result of executing Purpose1-1 is extremely poor match (due to the deception). Executing Purpose1-2 brings small improvement, reflecting the probability that the most used expert fits the profile of the average user. The most significant improvement happens by executing Purpose1-3, which takes into account the trust values which similar users have in each expert in computing the matching expert for each user. Purpose1-4 which is a refinement over Purpose1-3 taking into account the actual degree of similarity between users (which of course, influences the trust in their trust values) brings a small, nearly negligible improvement.

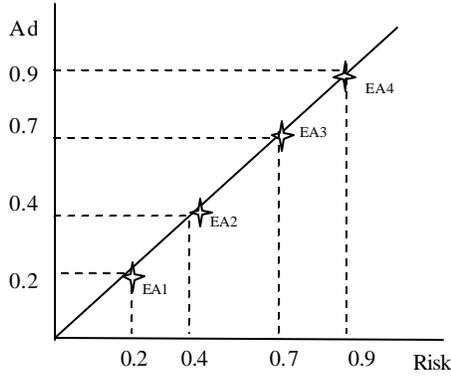


Fig. 6. The distribution of advertisement and risk in experiment 2 (no expert deception)

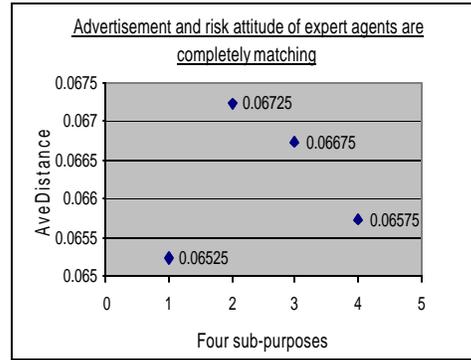


Fig. 7. Expert recommendation improvement with executing more purposes (fixed number of agents, no expert deception).

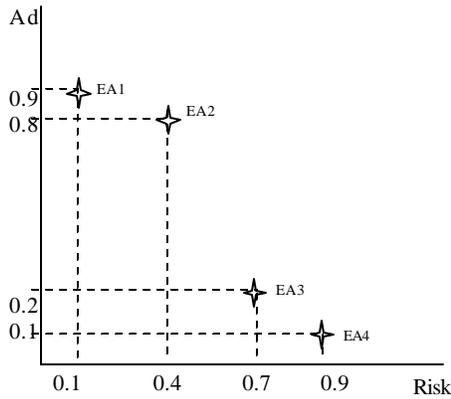


Fig. 8. The distribution of advertisement and risk in experiment 2 (total expert deception)

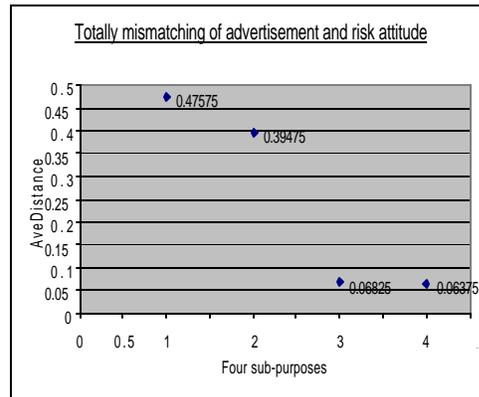


Fig. 9. Expert recommendation improvement with executing more purposes (fixed number of agents, total expert deception).

We also vary the value of the experts' real risk attitudes and advertisements in other different ways so that the study can get conclusion over a broad range of characteristics and advertisements. All the experiment results from those cases show that the expert recommendation improves with executing more purposes .

Experiment 3: Exploring aspects of the users' risk attitude distribution

In the third experiment, we fix the experts' advertisements and risk attitudes and vary the users' risk attitude distribution. The experts' risk attitudes and advertisements are fixed so that two of the four experts lie about their abilities and one is not precise in his advertisement, advertising 0.3 instead of 0.4 risk attitude. Figure 10 gives an intuitive idea of the distributions studied. Many cases were investigated corresponding to different regions of the curves in Fig. 10. For example, in one case the bulk of the users have average risk preference (modelled by a Gaussian distribution), in the other cases the distribution of the users is tending towards one of the extremes, e.g. most of the users have either risk-averse or risk taking attitude, and a "constant" case where there is one (or equal number) of users for each user type.

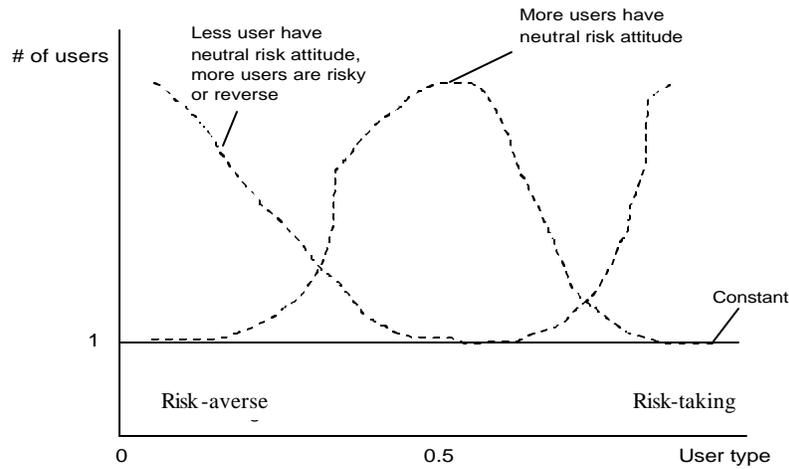


Fig. 10. Points on the lines show examples of cases of user types distribution

Fig. 11 shows the results using the Gaussian distribution of user types (more users have average risk-attitude). These results show that in this setup executing Purpose1-2 (selecting the most used expert agent) brings the most significant improvement in the average match between the user and selected expert profile. In the case of the Gaussian distribution where the majority of users have the same (average) risk attitude, this seems natural. In the setup where less users have average risk attitude, the results are slightly different from those shown in Fig. 11. In this case both purpose1-2 and purpose 1-3 contribute to the significant improvement of match. The distance becomes 0.1235 for Purpose1-2 and reaching values 0.059 for Purpose 1-3 and 0.0585 for Purpose 1-4. In each of the cases when more users are of risk taking type or risk-averse type, Purpose 1-2 seems to contribute most to the quality of match.

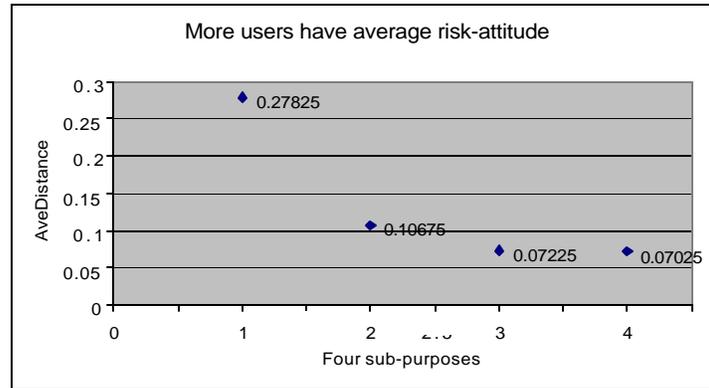


Fig. 11. Results from the third experiment

In summary, our experiments show that with the availability of more resources (more agents in the system and more time to execute more purposes), the quality of matching improves. Especially in situations where deception is likely, spending more resources (executing more and more resource-expensive purposes, like Purpose1-3) is well justified, since it leads to substantial improvement. In case that there is not much deception and the user population is fairly homogeneous, good results can be obtained also with less resources (only Purpose1-1 and Purpose1-2 lead to sufficient improvement of recommendations). These results help to establish a performance baseline for the system and to discover the influence of environment factors on the system and possible measures (e.g. reordering the purposes to take into account such factors as the expected level of deception). Future work will focus on experiments with real users and comparison of the performance of a real system using purpose-based expert modelling with the simulation baseline. If the performance with real users is similar to the simulation-generated baseline performance, it will allow optimizing the composition of purpose hierarchies and sequences using the results of simulations that take into account specific factors in the environment.

6 Conclusions

We have demonstrated through our experiments that the active approach is feasible and can be used to develop decentralized purpose-based user modelling computations to support decision making in a non-trivial domain. Arranging these into purpose hierarchies facilitates re-use. Having them deployed in an “anytime” fashion allows incremental addition of user modelling information (if needed to make better decisions and/or if time and resources allow). In short, the experiments provide a proof of concept of our approach and initial insight, at least, into performance issues.

However, the experiments do not directly show the advantages of the purpose-based user/agent modelling approach versus centralized matching of profiles of experts and customers. The strongest argument for the purpose-based modelling approach is that it distributes the load and costs required to maintain accurate and consistent expert profiles centrally and provides community-based evidence to the customers. Compared to centralized user/expert modelling, this approach implies fewer constraints on the agents (with respect to the need for a shared representation scheme, reliance on a connection to a server etc.) and is more robust (no central point of failure). It also allows for better protection of privacy, since the user models as well as users' private expert models are kept locally by the personal agents and can be shared according to pre-set policies of the users. We feel that the weakest aspect of 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, a set of overlapping expert modelling purpose cliché hierarchies will be devised for many domains and will be used to carry out active expert finding by heterogeneous software agents.

References

- Ackerman, M. S., et al. 1999. Recommenders for Expertise Management. In *Workshop on Recommender Systems, Algorithms and Evaluation* at the ACM SIGIR'99, Berkeley, CA.
- Boddy M. and Dean T. 1989. Solving time-dependent planning problems. *Proceedings International Conference on Machine Learning*. Morgan Kaufmann, Madison, Wisc.
- Browne, D., Totterdell, P., Norman, M. 1990. *Adaptive user interfaces*. Academic Press Ltd., London, UK.
- Bull, S., Greer, J.E., McCalla, G., Kettel, L., Bowes, J. 2001. User modelling in I-Help: what, why, when and how. *Proceedings of 8th International Conference on User Modelling*, UM 2001, Sonthofen, Germany, July 2001, Pages 117-126.
- Corkill, D. 1979. Hierarchical Planning in a Distributed Environment. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, 168-175.
- Fink, J., and Kobsa, A. 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), 209-249.
- Foner, L. N. 1997. Yenta: A Multi-agent Referral-Based Matchmaking System. *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, Marina del Rey, CA, 301-307.
- Giunchiglia, F., Zaihrayeu, I. 2002. 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.
- 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, Texas, LNCS No1452, Springer, 494-503.
- Kay, J. 1999. A scrutable user modeling shell for user-adapted interaction. *Ph.D. Thesis*. Department of Computer Science, University of Sydney, Sydney, Australia.
- Kobsa, A. and J. Fink (2003): Performance Evaluation of User Modeling Servers Under Real-World Workload Conditions. *Proceedings of the 9th International Conference on User Modeling*, Johnstown, PA, 143-153.

Xiaolin Niu, Gordon McCalla and Julita Vassileva

- McCalla G.I. and Greer, J.E. 1994. Granularity-based reasoning and belief revision in student models. In *Student Models: The key to Individualized Educational Systems* J. Greer and G. McCalla (eds), New York: Springer Verlag, 39-62.
- McCalla, G., Vassileva, J., Greer, J. and Bull, S. 2000. Active Learner Modeling. Proc. *Intelligent Tutoring Systems, ITS2000*, Springer LNCS 1839, 53-62.
- McDonald, D. W. & Ackerman, M. S. 2000. Expertise Recommender: a Flexible Recommendation System and Architecture. CSCW 2000, *Proceeding of the ACM 2000 Conference on Computer Supported Cooperative Work*, Philadelphia, PA, December 26, 2000, 231-240.
- Schafer, J.B., Konstan, J., and Riedl, J., Electronic Commerce Recommender Applications. *Journal of Data Mining and Knowledge Discovery*, vol. 5 nos. 1/2, pp. 115-152.
- Sutcliffe, A. 2002. *The Domain Theory: Patterns for Knowledge and Software Reuse*. Lawrence Erlbaum Associates, Publishers Mahwah, New Jersey, ISBN: 0805839518.
- Tang, T., Winoto, P. & Niu, X. 2002. 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.
- 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 *Proceedings of AIED'99*, Le Mans, France, July, 1999, 38-45 IOS Press: Amsterdam.
- Vassileva, J., McCalla, G., Greer, J. 2003. Multi-Agent Multi-User Modelling. *User Modelling and User-Adapted Interaction*, **13**:(1), 179-210.
- Vassileva, J.I., Greer, J.E., McCalla, G.I. 1999. Openness and disclosure in multi-agent learner models. In Morales R. and Kay J. (Eds.) *Proceedings of the Workshop on Open, Interactive, and Other Overt Approaches to Learner Modelling*, International Conference on AI in Education, Lemans, France.
- Vivacqua, A. S. 1999. Agents for Expertise Location. In the Proceedings of the AAI Spring *Symposium on Intelligent Agents in Cyberspace*, Stanford, CA, March 1999.
- Yimam, D. and Kobsa, A. 2000. Expert Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach. In Ackerman, M., Cohen, A., Pipek, V. and Wulf, V. eds.: *Beyond Knowledge Management: Sharing Expertise*. Boston, MA: MIT Press.