Active user interface in a knowledge discovery and retrieval system

Hien Nguyen, G. Mitchell Saba, Eugene Santos Jr.
Intelligent Distributed Information System lab
Department of Computer Science and Engineering
University of Connecticut

Scott M. Brown
Air Force Research Laboratory
Crew System Interface Div.
WPAFB OH 45433-7022

Abstract

We discuss the construction of an active user interface agent in our on-going research toward the development of a system to help users retrieve information and discover knowledge for a medical domain application. The primary goal of the interface agent is to adaptively react to the dynamic changes in the user's interests and preferences in searching for information within the context of the on-going information retrieval task. The agent dynamically constructs and maintains a decision-theoretic user model that is updated and corrected over time. It is designed to support the complex notion of shifting user interests and preferences by combining his long-term and short-term interests intuitively. The context in which the user seeks information is modeled by the agent through analyzing the user's interactions with the system to dynamically construct an ontology of concepts representing the user's information seeking context. User feedback is used to verify concepts prior to their commitment to a context knowledge model and to make requests to the target system to update the domain knowledge when needed.

Keywords: Assistant agent, active user interface.

1. Introduction

Information seeking tasks occur every day in many different domains. Whether we are looking for the latest information on the weather, searching the news for interesting tidbits, or surfing the Web, we are seekers of information. With regards to information seeking, an *interest* (or *interests*) defines what we are seeking. During the information-seeking task, we use certain methods, for example, going to one web search engine versus another, using a local news station versus the national cable weather station, and prefer one method to another. As we seek information, the interrelated conditions in which parts of the discourse that surrounds a concept (e.g: word, passage) throws light on its meaning. This context/user interest is crucial in defining why a user seeks for the information.

Information seeking can be a time intensive and cognitively demanding task. The need to help people reduce their work and information overload has led to a flood of research into the construction of personal assistant agent [7], [8],[9],[14]. The major task of any personal assistant agent, also called an interface agent, is to provide intelligent assistance to the user. The assistance becomes more effective if it learns the user's interests and preferences [8]. We are interested in building an interface agent that proactively and adaptively assists the users in a knowledge discovery and retrieval system focusing on the medical domain. The guiding principle behind the design of the interface agent is the efficient construction of the model of the user's long and short-term interests and preferences; dynamic reasoning from the user's information seeking context; and, applying decision theoretic principles and probabilistic reasoning techniques wherever they are appropriate. This leads to a number of new and distinct features in our interface agent. In particular, we clearly separate the concepts of interest and preferences in a dynamic fashion. In this paper, we present our preliminary designs and results based on our current on-going work. Our approach is derived from our earlier attempt with a predecessor system, Clavin[1],[13].

The rest of the paper is organized as follows. The next section discusses the background related to the desired features of our interface agent. Then we describe the architecture of the interface agent in our medical information seeking system. Next, we go over some related work. Lastly, the on-going work and research issues are presented.

2. Background

The interface agent must autonomously react to changes in *user intent* as well as changes to information sources -- for example, a hospital's online patient records database together with a medical knowledge-base/electronic handbook -- by proactively and dynamically constructing the appropriate queries for the various (possibly heterogeneous) information sources given the changes identified. The term "user intent" denotes the action a user intends to perform in pursuit of his goal [2]. For example, a medical student wants to find out more about acute leukemia. He may ask questions about its definition, its causes, and possible cures. If an interface agent is able to assist the user in pursuing his goal, it must be capable of ascribing user intent to offer timely, beneficial assistance. In the above example, the interface agent may ascribe the student's intention of understanding more about leukemia to suggest related important information about the existing experiments, advance research work, contact addresses of people and institutions who are doing research on this disease. An accurate user model is considered necessary for effective ascription of user intent. We use utility theory and Bayesian network techniques to construct the user model "on-the-fly" from the user's interactions with the system [4],[13]. By maintaining the dynamic user model, it allows the system to start with no knowledge of the user and incrementally build the user model as the user interacts with the system.

Our interface agent is designed to support the complex notion of dynamic interests and preferences given the information-seeking context. The term *interests* denotes the topics or subjects which the user is focusing on in the information-seeking task. The term *preferences* denotes how the user would go about acquiring and viewing the desired information. For example, the user prefers to access general information first before going into detail and/or prefers a graphical view to a textual view. While people can be characterized to have general interests and preferences that are persistent over time, what they are currently looking for may deviate from these interests and preferences. The interface agent must capture the user's long-term and short-term interests and preferences and combine them in an intuitive manner. By long and short-term interests and preferences, we refer to ones that persist over time and ones that he/she is currently investigating.

One design goal of our interface agent is to discover new *knowledge* from the subjects that the user is studying and add it to the existing knowledge base. The term "knowledge" denotes the interrelation of the user's interests and its relationships with the existing knowledge from the knowledge base. Take for example, from a series of user interactions in seeking information for migraine headaches, we learned that a rapid widening and narrowing of blood vessel walls in the brain and head are its direct causes while emotional stress and alcoholic beverages are indirectly related to this disease. We can define a "cause-by" relationship between migraine and its direct causes and a "related-to" relationship between migraine and its indirect but relevant causes. We may add these relationships to the existing knowledge base if they do not exist yet. The challenge then is to avoid adding all possible facts which we may obtain from the user interactions and only maintain those which are pertinent to the current user interests. Failure to do so will result in the immediate practical problem of computational tractability, as the user model becomes highly unwieldy in size. In addition to discovery new knowledge from a sequence of a user's interactions, we use the existing knowledge base to help the user in his information seeking tasks. Take for example, a user is looking for information about migraine. In the existing knowledge base, we know about its definition, its causes, its related diseases and its prevention. We therefore can use existing information in the knowledge base to construct the proactive query on the user's behalf not only to retrieve all the relevant information for a particular query, but also to infer the user's overall goal in searching.

3. Framework

The interface agent is based on the Core Interface Agent (CIA) architecture [4]. The purpose of this architecture is to provide assistance to the user by maintaining an accurate model of the user's interaction with the target system environment. A user interacts with a target system (e.g. a medical database querying system), typically via direct manipulation interface such as through menu selections, mouse clicks, and button pressed. The user's interaction with the target system is reported to CIA architecture as observations. The interface agent uses these observations to infer what a user is doing within the environment. Based on the knowledge of the environment that the active user interface has and the user's current interactions with the system, the interface agent determines the user's goal with the highest expected utility and offers a suggestion to the user via the target system.

The user model consists of three components: a user profile, a Bayesian network model, and a utility model. The user profile is used to store the static knowledge about the user, including his demographic data, and skills.

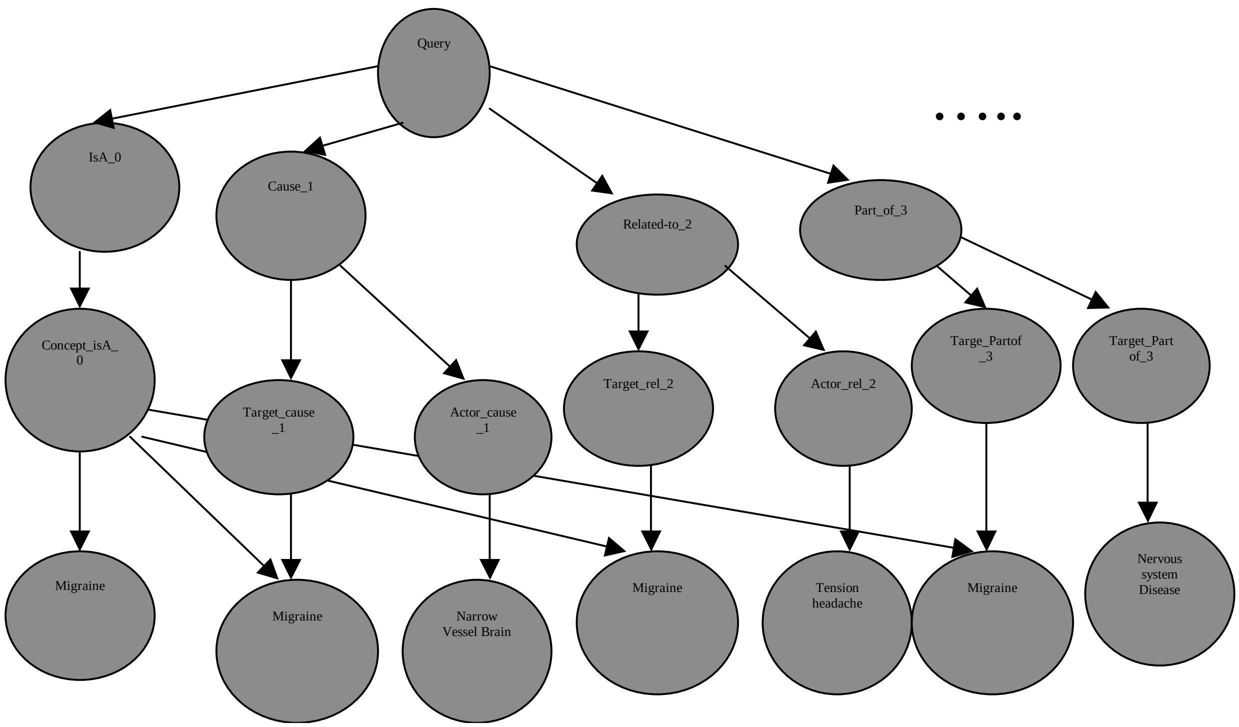

Figure 1: A portion of the action network**.**

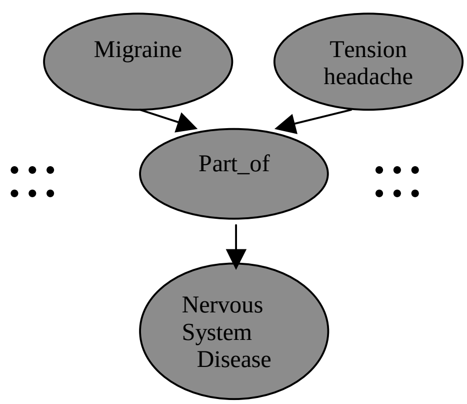

Figure 2: A portion of the ontology network obtained from the above action network.

Our approach is based on the belief that what a user intends to do in an environment is the result of environmental events and stimuli occurring in the environment and by the goals he is trying to achieve [5]. Goals can be composed of multiple actions with many pre- and post-conditions. Therefore, a Bayesian network model is appropriate here to capture the uncertain, causal relationship between the pre-conditions, goals and actions. In our particular information retrieval and knowledge discovery application, our model consists of an *action* and an *ontology* network. The action network is built from the user's natural language queries and relevant feedback on the results given by the system. It is maintained on regular basis by a fading technique. The CIA architecture supports the fading function and the ability to correct the user's model once it is found to be inaccurate. The fading function is used to "forget" past actions. It helps to focus on the recent and relevant actions and reduces the complexity of reasoning process over a long period of time. As the user interacts with the system, the interface agent receives a set of observations. They are stored in a history stack and are used as nodes and evidences in the action network. We use the fading function to "fade" these evidences over time and therefore "clean" the action network. There are several types of fading functions, including a time-based, and a length-based function. The action network captures the user's interests which are controlled in the ontology network. When the action network is faded, the user's actions which persist over time are inferred and copied to the ontology network. The ontology network captures the information regarding the user's long-term interests as well as captures the information regarding the interrelation among the subjects that the user is looking for.

While the Bayesian network model can capture the uncertain factors of the user's actions and goals and the ontology of the user's interests, the utility model is needed to capture the user's utility for having the active user interface performing an action on his behalf to achieve a goal. It is used to support the shifting in his long-term preferences and interests. The utility model contains the utility function over a set of requirements for our active user interface *($U_{aui\_requirement}$)* and the utility function over a set of requirements for capturing the user long-term interests and preferences ($U_{longterm\_requirement}$). The $U_{aui\_requirement}$ is a multi-attribute utility function over a set of metrics that measure the adaptivity, autonomy, collaboration, and robustness of the active user interface. It is used to determine if our active user interface is meeting the requirements. $U_{long\text{-}term\_requirement}$ utility function is defined over a set of metrics that measure the scope of knowledge, the rate of changes, the generality and specificity of the user's studying style. In the current implementation of the user model, we assume that both of these utility functions have additive forms.

We define three thresholds, one for offering assistance, one for autonomously performing action on the user's behalf and one for the long-term interests. These thresholds are used to determine how/if the interface agent will offer assistance and if the observation is included in the user's long-term interests, respectively. This approach is the same as the one presented in [8], except that the thresholds in [8] are based on statistical probabilities and the one in our approach is based on the expected utility function.

When the interface agent fails to meet its requirement (i.e. $U_{aui\_requirement}$ is falling below the threshold of utility requirement), we need to identify which requirements are not being met and attempt to correct the problem by updating the user model [4]. We take the approach of having the active user interface request "help" from a set of correction adaptation agents. Each correction adaptation agent offers a "bid" to the active user interface. The one that most likely improves the interface agent's requirement utility will win the bid and gets the chance to correct the user model. Some examples of the correction adaptation agent includes an agent to change the number of time slices that the interface agent needs to look backwards in history to compute the utility function or an agent that extends the user's original queries by taking only the positive or both positive and negative relevant feedback.

Figure 1 shows a portion of an action network of a user who is looking for information about migraines. Assuming that the user is new to the system and starts using the system by asking " What is migraine?", the query is converted to the first order logic form such as: isA(concept: migraine, definition: x). The interface agent then determines whether the user model is still correct or not by computing $U_{aui\_requirement}$ to see if it is less than the threshold of utility requirements. As mentioned above, $U_{aui\_requirement}$ is computed based on the four requirements that we feel best capture the essence of the interface agent. They are adaptivity, autonomy, collaboration, and robustness. Each requirement is associated with one or many sets of metrics. For example, the metrics for autonomy requirement is defined as follows:

$$M_{external\_autonomy} = \frac{n_a}{N}$$

in which n_a is the number of autonomous suggestions and N is the total number of suggestions suggested by the interface agent. For more detail information on metrics and requirements, please see our previous paper [5]. If the user model is not correct, i.e, the $U_{aui\_requirement}$ is less than the threshold of utility requirement, the interface agent asks the correction adaptation agents (CA) to help correct the existing user model. For example, one CA agent may suggest that we need to extend the action network by adding new nodes representing the newly asked query and construct the proactive query on the user's behalf. Another CA may suggest reducing the number of observations that we already observed in the past to focus on the recent observations or vice versa. Assuming that the CA suggested extending the action network wins the bid. The action network of the user model is then added some new nodes representing the newly asked query including "isA_0", "concept_isA_0", "migraine" nodes, as seen in Figure 1. A proactive query in Prolog-like format is constructed, in this case, simply being isA(migraine,x)" because the user is new to the system and has not issued any other queries yet.

The interface agent then sends the proactive query constructed by the winning CA to the database and search for relevant information. The search may return "Migraine is recurrent, throbbing headache generally left on one side of the head". In the action network shown in Figure 1, we made an assumption that the user did not give any feedback this time. The user continues to use the system and the similar process goes on with next several questions, such as, "What causes migraines", "What kind of headaches relates to migraine?" and "Are migraine and tension headaches part of nervous system disorders?". Assuming that after the questions about the causes of migraine, the system returns "Migraines are caused by a rapid widening and narrowing of blood vessel walls in the brain and head" and the user has confirmed that the answer is relevant to what he is looking for. The action network then is updated to reflect the user's feedback. The node actor_cause_1(in Figure 1) now links to the node "Narrow Vessel Brain" representing the user's feedback. After a series of four questions from the user, the action network contains information about concepts such as "migraine", "blood vessel", "tension headache", "nervous system disease", and the relationships between them such as "isa", "cause", "related to", and "part of". If the users keeps referring to the migraine and tension headache as a kind of nervous system disorder and this concept did not exist in our knowledge base yet, we add a new knowledge and its relationship with the existing knowledge of migraine and tension headache to our ontology network (as shown in Figure 2). Now, assuming that the user switches his topic to leukemia by asking "What causes leukemia?" The proactive query constructed then should include the definition of the leukemia, its causes, its related factors, and superior diseases. If the user keeps searching on information directly or indirectly related to migraine or leukemia over a long period of time, then these concepts are stored in the ontology network.

4. Related work

Capturing the changes in the user's interests and preferences is a challenging problem in information filtering, information retrieval and recommender system [10]. However, little work has been done in this regard. Webmate [6] and Alipes [15] represent long and short-term categories of interests by a set of vectors that contains a list of keywords and their corresponding weights indicating their important degrees. While in [15], authors claim that by taking into account both the negative and positive feedback results in a better result in terms of learning compared to using positive feedback alone (as in Webmate), both of the systems combine the long and short-term interests in an ad-hoc and unjustified fashion. Amalthaea [11] uses genetic algorithm to learn the changes in user's interests in an information filtering system. It however, requires a large number of generations before the system is back to an equilibrium status. Our work is different with the existing work in that it differentiates the user's interests and preferences in an information-seeking task. We also discover the new knowledge from the user's interests and use the existing knowledge base to guide the user interests.

5. Discussion

This paper has described our on-going work to construct an active user interface that provides intelligent assistance to the user in a knowledge discovery and retrieval system. We started implementing our prototype and chose the Unified Medical Language System (UMLS) knowledge base as our initial testbed. There are a number of issues that have arisen during the process of designing the early prototype of the system. In our active user interface, we construct the user model "on-the-fly". This approach allows the agent starting with no domain knowledge and incrementally construct the user model as the user interacts with the target system. We would like to evaluate the effectiveness of this approach and compare this with the existing approaches in building the user model statically and semi-dynamically. The active user interface autonomously builds the ontology between concepts and request to update the knowledge base to include this ontology. The question is how much autonomous we need in doing the job and how we can validate the new knowledge before committing it to the existing knowledge without causing any conflicting.

6. Acknowledgement

This work was supported in part by AFOSR Grant No. F49620-99-1-0244 and the AFRL Human Effectiveness Directorate through Sytronics Incorporated.

7. References

[1] Brown, S. M. and Santos, E. Jr. 1999. Active User Interfaces for Building Decision-Theoretic Systems. In *Proceedings of the 1st Asia-Pacific Conference on Intelligent Agent Technology*, Hong Kong.

[2] Brown, S. M. and Santos, E. Jr. 1999b. Active User Interfaces. In *IDIS Technical Report No. 101*, Intelligent Distributed Information Systems Laboratory, University of Connecticut.

[3] Brown, S. M.; Santos, E. Jr.; and Banks, S, B. 1997. A dynamic Bayesian Intelligent Interface Agent. In *Proceedings of the sixth International Interfaces Conference (Interfaces 97),* 118-120, Montpellier, France.

[4] Brown, S. M., Santos, E. Jr., and Banks, S. B. 1998. Utility theory-based user models for intelligent interface agents. In *Lecture Notes in Artificial Intelligence 1418: Advances in Artificial Intelligence -- AI '98*, 378-392, Springer-Verlag.

[5] Brown, S. M., Santos, E. Jr., Banks, S. B., and Oxley M.E. 1998. Using explicit requirements and metrics for interface agent user model construction. In *Proceedings of the Second International Conference on Autonomous Agents*, 1-7, Minneapolis, MN.

[6] Chen, L. and Sycara, K. 1998. WebMate: A Personal Agent for Browsing and Searching. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multi Agent Systems*, Minneapolis, MN.

[7] Chin, D. 1991. Intelligent Interfaces as Agents. Sulluvan J. and Tyler, S (eds.). *Intelligent User Interfaces.*177-206.

[8] Maes, P. 1994. Agents that reduce work and information overload. *Communications of the ACM*. 37(7),31-40,146.

[9] Horvitz, E., Breeze, J., Heckerman, D., Hovel, D., and Rommelse, K. 1998. The Lumie're project: Bayesian user modeling for inferring goals and needs of software users. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI 98).*

[10] Konstan, J. A.; Rield, J.; Borchers, A.; Herlocker, J. L. 1998. Recommender systems: A GroupLens perspective. In *Working notes of the 1998 AAAI workshop on Recommender Systems*, 60-64.

[11] Moukas, A. and Zacharia. 1997. Evolving a Multi agent information filtering solution in Amalthaea. In *Proceedings of Autonomous Agents 1997*.

[12] Salton, G. and McGill, M. J. 1983. Introduction to Modern Information retrieval. McGraw-Hill.

[13] Santos, E. Jr., Brown, S. M., Lejter, M., Ngai, G., Banks, S. B., and Stytz, M. R. 1999. Dynamic User Model Construction with Bayesian Networks for Intelligent Information Queries. In *Proceedings of the 12th International FLAIRS Conference*, 3-7, Orlando, FL.

[14] Sycara, K., Decker, K., Pannu, A., Williamson, M., and Zeng, D. 1996. Distributed intelligent agents. *IEEE Expert 11(6): 36-46.*

[15] Widyantoro, D. H., Ioerger, T., and Yen, J. 1999. Adaptive Agent for Learning Changes in User Interests. In *Proceedings of the International Conference on Information and Knowledge Management CIKM'99*, Kansas City, Kansas.