Dynamic User Model Construction with Bayesian Networks for Intelligent Information Queries

Eugene Santos Jr.
Dept. of Comp. Sci. & Eng.
University of Connecticut
Storrs, CT 06269-3155
eugene@cse.uconn.edu

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

Moises Lejter **Grace Ngai**
Dept. of Comp. Sci. & Eng.
University of Connecticut
Storrs, CT 06269-3155

Sheila B. Banks **Martin R. Stytz**
Calculated Insight
Orlando, FL 32878-0214

Abstract

The complexity of current software applications is overwhelming users. The need exists for intelligent interface agents to address the problems of increasing taskload that is overwhelming the human user. Interface agents could help alleviate user taskload by extracting and analyzing relevant information, and providing information abstractions of that information, and providing timely, beneficial assistance to users. Central to providing assistance to a user is the issue of correctly determining the user's intent. The Clavin project is to build an intelligent natural language query information management system. Clavin must maintain a dynamic user model of the relevant concepts in the user inquiries as they relate to the information sources. The primary goal of Clavin is to autonomously react to changes in user intent as well as the information sources, by dynamically constructing the appropriate queries relative to the changes identified. In this paper, we discuss the problems and issues that arise in achieving user-intent ascription through dynamic user model construction with Bayesian networks.

Introduction

The goal of *interface* or "personal assistant" agents is to reduce information overload by collaborating with the user and performing tasks on the users' behalf (Maes 1994). Examples of interface agents include office assistance agents, such as e-mail, scheduling, and financial portfolio management agents (Maes 1994; Sycara *et al.* 1996; Boone 1998); tutor and coach agents (Chin 1991; Conati *et al.* 1997); and character-based assistants for word processors, spreadsheets, and presentation software (Horvitz 1997; Horvitz *et al.* 1998). Unfortunately, most of these agents have either been pedagogical or narrowly focused.

Reducing user task load involves providing intelligent assistance to the user. Providing intelligent assistance and performing tasks on the user's behalf requires an understanding of the goals the user is performing, the motivation for pursuing those goals, and the actions that can be taken to achieve those goals. The term *user intent* denotes the actions a user intends to perform in pursuit of his/her goal(s). The term *user intent ascription* is the attribution of actions to the goal(s) a user will pursue. That is, user intent ascription is the process of determining which actions are attributable to a specific goal or goals. Therefore, for an interface agent to be able to assist the user in pursuing those goals, the agent must be capable of ascribing user intent to offer timely, beneficial assistance. An accurate user model is considered necessary for effective ascription of user intent.

User modeling is concerned with how to represent the user's knowledge and interaction within a system to adapt the system to the needs of the user. The benefit of utilizing a dynamic user model within a system is to allow that system to adapt over time to a specific user's preferences, work flow, goals, disabilities, etc. To realize this benefit, the user model must effectively represent the user's knowledge and intent within the system to accurately predict how to adapt the system.

The Clavin project is to build an intelligent natural language query information management system. Clavin must maintain a dynamic user model of the relevant concepts in the user inquiries as they relate to the information sources. The primary goal of Clavin is to autonomously react to changes in user intent as well as the information sources, by dynamically constructing the appropriate queries relative to the changes identified. In this paper, we discuss the problems and issues that arise in achieving user-intent ascription through dynamic user model construction with Bayesian networks.

User Intent Ascription — From Goals to Actions

To ascribe user intent, interface agent designers must identify the salient characteristics of a domain environment and specifically determine goals a user is trying to achieve, the reason and/or cause for pursuing those goals, and the actions to achieve those goals (Brown *et al.* 1998a). This approach is based on the belief that what a user intends to do in an environment is the result of environmental events and/or stimuli occurring in the environment and by the goals they are trying to obtain as a reaction to the events and stimuli. That is, the reason *why* users perform actions is to achieve

goals they pursue as a result of environmental stimuli (Pestello & Pestello 1991).

The terms *task* or *intentional level* describe the component of a user model containing knowledge about the user's goals (Benyon & Murray 1993). The task level knowledge is used to infer the goals the user is pursuing. The "failure to recognize the intentions underlying some user action will result in less satisfactory interaction" is the result of failing to recognize the pursuit of one goal versus another.

To achieve a goal a user must perform certain actions. Goals can be composed of multiple actions with many pre- and post-conditions. Pre-conditions are directly observable events in the environment. These preconditions cause a user to pursue a goal and/or affect the goal a user will pursue. Additionally, other factors affect the goals a user pursues as well as the actions the user will take to achieve the goal. In particular, human factors (e.g., skill, work load, expertise, etc.) all affect the user's decision to pursue goals and perform actions in pursuit of goals. Typically, these factors are not directly observable but they are measurable, either *a priori*, such as skill or expertise, or dynamically as the user interacts with the environment, e.g., work load.

A directed acyclic AND/OR graph shows causality between the pre-conditions, goals, and actions. For AND goals, all the actions must be performed to achieve the goal. For OR goals, only one action is needed to achieve the goal. Similarly, pre-conditions for a goal may all have to be present (AND), or one or more may need to be present (OR). Other possible relationships can exist. For example, an XOR relationship would represent the case where only one pre-condition or action can be "active" for a goal.

There are several advantages to representing users' intentions in a goal hierarchy, as represented by a directed acyclic AND/OR graph, such as the following:

- Goal abstraction allows one to design and detect higher level goals, in pursuit of lower level goals.
- Keyhole plan recognition is made easier by explicitly enumerating pre- and post-conditions and atomic actions composing goals (Albrecht *et al.* 1997; Waern 1996).
- Natural language explanations of actions based on prediction of goals can be easily generated from the structure.

Core Interface Agent Architecture

The Core Interface Agent (CIA) architecture is a multi-agent system composed of an interface agent and a collection of correction adaptation agents. The purpose of the architecture is to provide assistance to the user which is accomplished by maintaining an accurate model of the user's interaction with the target system environment. The user model is used to ascribe the user's intent. The task of ascribing user intent is delegated to the interface agent component of the architecture, while continual adaptation of the interface agent's user model is a task shared by the interface agent and the collection of correction adaptation agents.

A user interacts with a target system, typically a direct manipulation interface. This interaction — the menus chosen, the buttons pressed, the text typed — as well as other target system environment stimuli (e.g., a spelling error, the arrival of a new e-mail message) are communicated to the CIA architecture as observations. These observations are used by the interface agent to infer what a user is doing within the environment and to ascribe the user's intent. A keyhole plan recognition approach is used, where the human is unaware of or indifferent to the user intent ascription process. Based on the knowledge of the environment that the interface agent possesses and the user's interaction with the environment, the interface agent determines the goal with the highest expected utility and offers a suggestion to the user via the target system. If the interface agent determines its user model is inaccurate, it begins a bidding process with the correction adaptation agents. The correction adaptation agents offer "bids" to modify the interface agent's user model. The interface agent allows the correction adaptation agent offering the best bid to modify the user model.

Each target system observation (environmental stimuli or user action) is communicated to the agents via the KQML message passing API (Mayfield, Labrou, & Finin 1996). Every observation is stored by the agents' evaluator in a history stack (i.e., most recent observation is on the top of the stack). These observations are used by the agents as evidence in the Bayesian network-based user model. Evidence may "fade" over time, essentially allowing the interface agent to "forget" past observations. The architecture supports (possibly) unique fading functions for each observation. The types of fading functions supported include a time-based fading function (evidence is relevant for a specified time) and a queue-based fading function (only the N newest observation are relevant).

The user model is composed of three components: the Bayesian network user model, a utility model, and a user profile. The Bayesian network user model captures the uncertain, causal relationship between the pre-conditions, goals, and actions. The utility model contains the utility functions for the attributes (i.e., human factors) and the additive multi-attribute utility function combining the other utility functions. The utility functions capture a user's utility for having the interface agent perform an action on the user's behalf to achieve a goal. The user profile captures knowledge about the user including background, interests, and general knowledge about the user that is typically static. Two user defined thresholds, one for offering (collaborative) assistance and one for autonomously performing actions on the user's behalf to obtain a user's goal, determine how/if the interface agent will offer assistance. This approach is the same as the one presented by (Maes 1994), except she based her thresholds on statistical probabilities and the ones in

this research are based on the expected utility function. The user profile also contains the values of any *static* human factors (e.g., skill, spatial memory). The normalized values of these human factors can be determined off-line and do not change as the user interacts with the target system. Detailed description of the CIA can be found in (Brown *et al.* 1998b; 1998a; Brown, Santos Jr., & Banks 1998).

Clavin

Clavin responds to users' natural language inquiries by forming intelligent queries to a database of potentially dynamic, heterogeneous information sources. Clavin maintains a dynamic user model of the relevant concepts in the user inquiries as they relate to the information sources. The user model allows Clavin to determine the relevancy of the various concepts in the domain, in order to properly address user inquiries. Clavin should be capable of proactively retrieving relevant information for the user based on the current inquiries and the user's previous history of inquiries.

The Clavin System architecture is shown in Figure 0.1. The system consists of four main components. The human language interface (HLI) component is composed of a commercial off-the-shelf voice recognition system and a natural language (English) parser. The voice recognition system is responsible for transforming a spoken utterance by the user into a natural language sentence. The natural language sentence is then parsed into an S-expression predicate logic representation (well-formed formulas) of the user queries by the natural language parser (NLP) component. The well-formed formula (wff) query is then passed to the interface agent. (An example of an uttered query, its wff representation, and the resulting representation in the user model is shown in Figure 0.2.) Once the interface agent processes the query (see the discussion below), the query is passed to the retrieval engine. The engine parses the query wff and transforms it into a database query. Heterogeneous, possibly dynamic information sources are then queried by the retrieval engine. These information sources are presented to the user as one homogeneous information source. Results are then passed back to the interface agent for data visualization of the resulting query. If no results are returned, the interface agent can request the HLI to produce another possible parse (i.e., re-tag the parts of speech and generate alternative wffs) for the uttered query.

The CIA architecture is responsible for two key functions within the Clavin system: information filtering and proactive querying. The first occurs as a result of adding context to the spoken queries. The second occurs as a result of combining various relevant pieces of previous queries. For example, if the user asks "What causes Lyme disease?"[1] the system returns information about deer tick bites; then the user's next query asks "What treats Lyme disease?", to which the system replies with information concerning a new Lyme disease vaccine. At this point, the user model contains information about the concepts "Lyme disease," "tick bite," "Lyme disease vaccine," "causes," and "treats." If the user then makes a inquiry about "what causes cancer?" the interface agent component can use the CIA architecture to not only retrieve information about the causation of cancer, but also proactively query the database about possible treatments for cancer. The proactive construction of queries is discussed in the next section.

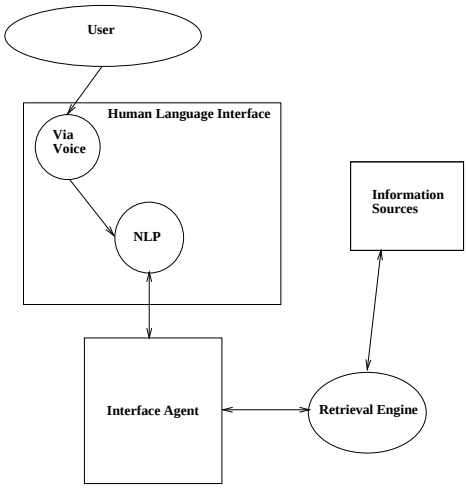

FIG. 0.1. The Clavin System Architecture.

Clavin User Model Construction

To ascribe user intent, interface agent designers must identify the salient characteristics of a domain environment and specifically determine goals a user is trying to achieve, the reason and/or cause for pursuing those goals, and the actions to achieve those goals. As Figure 0.2 shows, the user's inquiry, as represented by the wff, has a hierarchical structure representation within the Bayesian user model. The leaves of the graph are actions and the rest of the nodes are sub-goals, except for the query node, which is a goal node.

To construct proactive queries based on relevant concepts within the user model, the interface agent uses the rank ordering feature of the decision-theoretic CIA architecture. Internally, at each time slice (i.e., observation), the CIA architecture ranks every node in the user model. When the user utters a new inquiry, the interface agent and correction adaptation agents receive notification that a new query has been added to the user model by the wrapper agent. For the Clavin system, a domain-dependent correction adaptation agent was designed to proactively generate new queries based on the user's current inquiry and past inquiries and concepts seen. Taking the rank ordering of the nodes

[1] As one of our knowledge sources, we currently use the Unified Medical Language System (UMLS), a library of widely available medical-related knowledge sources actively compiled by the National Institute of Health.

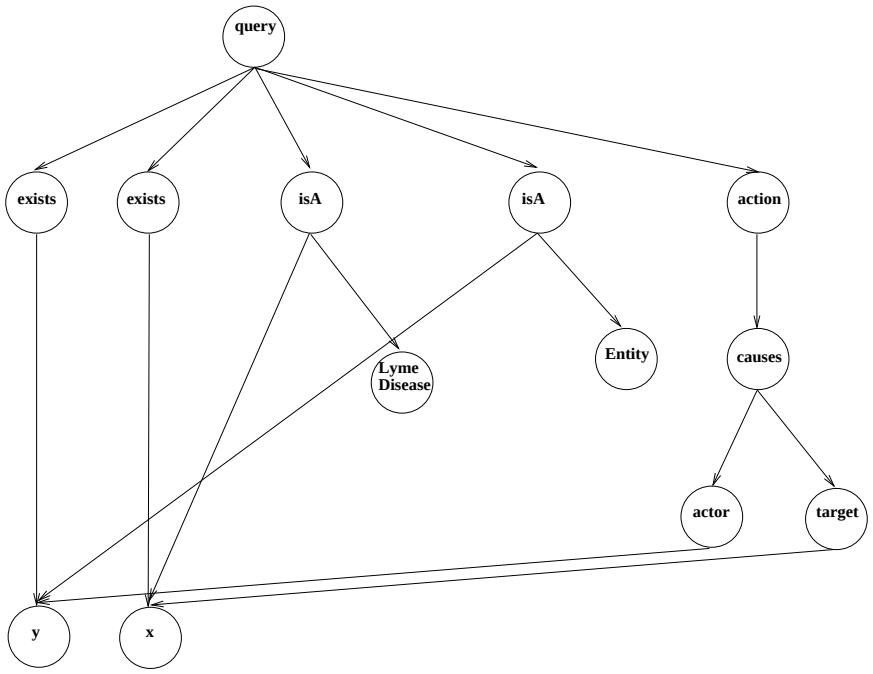

FIG. 0.2. A User Model Representation of the Spoken Query "What causes Lyme Disease?". The utterance is transformed to the wff (exists x)(exists y)(isA x Lyme Disease)(isA y Entity)((action causes)(actor y) (target x)).

for the most recent time slice, this correction adaptation agent performs a traversal over previous queries to determine which might be candidates for a proactive query. The agent only selects queries with an expected utility greater than the user's autonomous threshold, and which guarantees that resulting proactive query will have an expected utility greater than the user's autonomous threshold. This approach insures the interface agent will make this query autonomously. After candidate queries have been selected, the children of these queries are considered for inclusion in the proactive query. The chosen children are then further processed to determine which of their children are processed. There are several considerations when generating the proactive query. If a parent was selected for the proactive query, at least one child will be, since the expected utilities are added and normalized for AND nodes and the maximum expected utility child is selected for OR nodes.

Discussion

In our development and early prototyping of Clavin, a number of issues have arisen while some predicted issues never occurred. Here are some of the most relevant issues to Clavin and interface agents in general:

1. **"On-the-Fly" User Model Construction:** The CIA architecture's decision-theoretic approach offers a framework with which to determine which subqueries are relevant and can be combined to make proactive queries on behalf of the user. By using the expected utilities of the various sub-query concepts (e.g., isA(x, dog), color(y, red), action(hit(john, ball))), the interface agent can combine those subqueries with high expected utility. This approach allows the agent to combine concepts in a methodical fashion. Especially in situations where information is highly specific (few variables), this procedural-based user model construction method allows the CIA architecture to begin with no domain knowledge and incrementally construct the user model as the user interacts with Clavin. This unanticipated side benefit allows designers to mitigate the classic knowledge acquisition bottleneck problem.
2. **Feedback:** While feedback from the user must be an integral part of the system for user intent modeling, the current approach of positing alternative queries and results to the user seems cumbersome. Is there a theory or model to better capture the users desires without a potentially lengthy trial and error approach?
3. **Agent Autonomy:** Since Clavin interacts with many heterogeneous information sources, some of them dynamic (e.g., news streams), the interface agent component of Clavin should be free to autonomously fetch new information, reason over it, and present updated results to the user. The issue becomes the following: how much autonomy should the agent have?
4. **Dynamic Information Sources** Unlike the current approaches, Clavin is presented with dynamic, possibly massive information sources. Therefore a declar-

ative approach to user model construction is not feasible. The sub-query relevance approach taken in Clavin offers a useful alternative to the declarative approach as well as data-centric, statistical learning approaches.

In conclusion, the Clavin system has attempted to construct an intelligent natural language query information management system. It must deal with the issues of dynamic user modeling and accurate user intent ascription. Our approach has been to look at the larger overall information process in order to better appreciate the bigger role interface agents will play in future information intensive systems.

References

Albrecht, D. W.; Zukerman, I.; Nicholson, A. E.; and Bud, A. 1997. Towards a Bayesian model for keyhole plan recognition in large domains. In Jameson, A.; Paris, C.; and Tasso, C., eds., *User Modeling: Proceedings of the Sixth International Conference, UM97.* Vienna, New York: Springer Wien New York. 365–376. Available from http://um.org.

Benyon, D., and Murray, D. 1993. Adaptive systems: from intelligent tutoring to autonomous agents. *Knowledge-Based Systems* 6(4):197–219.

Boone, G. 1998. Concept features in Re:Agent, an intelligent email agent. In Sycara, K. P., and Woolridge, M., eds., *Proceedings of the Second International Conference on Autonomous Agents (Agents '98)*, 141–148. ACM Press.

Brown, S. M.; Santos Jr., E.; Banks, S. B.; and Oxley, M. E. 1998a. Using explicit requirements and metrics for interface agent user model correction. In *Proceedings of the Second International Conference on Autonomous Agents (Agents '98)*, 1–7.

Brown, S. M.; Santos Jr., E.; Banks, S. B.; and Stytz, M. R. 1998b. Intelligent interface agents for intelligent environments. In *Proceedings of the 1998 AAAI Spring Symposium on Intelligent Environments.*

Brown, S. M.; Santos Jr., E.; and Banks, S. B. 1998. Utility theory-based user models for intelligent interface agents. In *Proceedings of the Twelfth Canadian Conference on Artificial Intelligence (AI '98)*, 378–392.

Chin, D. N. 1991. Intelligent interfaces as agents. In Sullivan, J. W., and Tyler, S. W., eds., *Intelligent User Interfaces.* ACM, New York.

Conati, C.; Gertner, A. S.; VanLehn, K.; and Druzdzel, M. J. 1997. On-line student modeling for coached problem solving using Bayesian networks. In Jameson, A.; Paris, C.; and Tasso, C., eds., *User Modeling: Proceedings of the Sixth International Conference, UM97.* Vienna, New York: Springer Wien New York. 231–242. Available from http://um.org.

Horvitz, E.; Breese, J.; Heckerman, D.; Hovel, D.; and Rommelse, K. 1998. The Lumiére 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).* San Francisco, CA: Morgan Kaufmann Publishers.

Horvitz, E. 1997. Agents with beliefs: Reflections on Bayesian methods for user modeling. In Jameson, A.; Paris, C.; and Tasso, C., eds., *User Modeling: Proceedings of the Sixth International Conference, UM97.* Vienna, New York: Springer Wien New York. 441–442. Available from http://um.org.

Maes, P. 1994. Agents that reduce work and information overload. *Communications of the ACM* 37(7):811–821.

Mayfield, J.; Labrou, Y.; and Finin, T. 1996. Evaluation of KQML as an agent communication language. In Woolridge, M. J.; Müller, J. P.; and Tambe, M., eds., *Intelligent Agents II: Agent Theories, Architectures, and Languages*, 347–360. Berlin: Springer.

Pestello, H. F. G., and Pestello, F. P. 1991. Ignored, neglected, and abused: The behavior variable in attitude-behavior research. *Symbolic Interaction* 14(3):341–351.

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

Waern, A. 1996. *Recognising Human Plans: Issues for Plan Recognition in Human-Computer Interaction.* Ph.D. Dissertation, Royal Institute of Technology.