ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-606-5-1055

ANTIPA: an agent architecture for intelligent information assistance[1]

Jean Oh and **Felipe Meneguzzi** and **Katia Sycara** [2]

Abstract. Human users trying to plan and accomplish information-dependent goals in highly dynamic environments with prevalent uncertainty must consult various types of information sources in their decision-making processes while the information requirements change as they plan and re-plan. When the users must make time-critical decisions in information-intensive tasks they become cognitively overloaded not only by the planning activities but also by the information-gathering activities at various points in the planning process. We have developed the ANTicipatory Information and Planning Agent (*ANTIPA*) to manage information adaptively in order to mitigate user cognitive overload. To this end, the agent brings information to the user as a result of user requests but most crucially, it proactively predicts the user's prospective information needs by recognizing the user's plan; pre-fetches information that is likely to be used in the future; and offers the information when it is relevant to the current or future planning decisions. This paper introduces a fully implemented agent of the *ANTIPA* architecture using a decision-theoretic user model.

1 Information-dependent planning problem

We focus on a class of problems where a user (or a planner) must access various types of information sources to acquire current information that is required for executing certain actions. Here, in addition to domain-specific planning objectives, the user must also take the cost of getting information into consideration in selecting actions. Furthermore, the quality of information (that also depends on the source of information) affects the user's transition to another state after taking the action.

For instance, consider a student preparing for a final exam by reviewing selected topics covered in a semester. When a question is encountered, the student may search online for a quick answer by taking a risk that the answer found may be incorrect, or email the teacher and wait to get a generally more precise answer. The outcome of the student's *action* (to understand the concept) depends on the quality of information, which in turn depends on the source from which it has come. For example, by receiving high-quality information, the student's *state* regarding the understanding of a certain concept is more likely to *transition* from `not-learned` to `learned`, also increasing the chance of getting a better grade (*reward*).

[1] Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

[2] Robotics Institute, Carnegie Mellon University, USA, email: {jeanoh, meneguzz, katia}@cs.cmu.edu

Given that the user is trying to solve an information-dependent planning problem, we aim to develop an agent that can adaptively identify and manage the user's information needs to facilitate the user's actions.

1.1 MDP user model

We make two specific assumptions in modeling the user's decision-making process. First, we assume that the user's decision-making process respects the Markov property: the conditional probability of being in a certain state in the next time step depends only on the user's current state and not on any past states. Second, we assume that users will try to maximize the plan quality (while minimizing the action cost) by means of their local information and bounded reasoning capability.

Based on these assumptions, we use a *Markov Decision Process* (MDP) [4] to represent the user's planning process. An MDP is a specification of a sequential (discrete time) decision-making process for a fully observable environment with a stochastic transition model, *i.e.*, there is no uncertainty regarding the user's current state, but transitioning from one state to another is nondeterministic. The user planning objective modeled in an MDP is to create a plan that maximizes her long-term cumulative reward.

Note that we do not use the MDP-based sequential decision-making model for the *ANTIPA* agent's decision making (on assisting the user) as in related work on assistive agents [1, 2] where the agent uses an optimal Partially Observable (PO) MDP policy to decide the best assistive action for its belief state. In contrast, we use an MDP to estimate how the *user* plans the future actions (when the user can fully observe her current state), so that the agent can *plan* information-gathering actions for the predicted user plans (from the assistant's current belief state) in order to satisfy the user's *future* information needs in a timely manner. Because the agent's decision making is only loosely tied to the user model the *ANTIPA* agent is not mandated to take actions sequentially, enabling a wide range of assistance using various planning techniques with flexibility, *e.g.*, constraint-based and parallel planning techniques.

2 The *ANTIPA* agent architecture

As part of the process of deciding on collecting and presenting information, the agent's reasoning process tries to accomplish four main objectives. First, the agent must identify the current state of the user from a sequence of observations on user activities. Second, the agent needs to predict the user's information needs that are changing dynamically through time. In order to accomplish this, the agent needs to identify the user's high-level goals and a set of planned actions to achieve these goals. If the agent can recognize the user's goals and plans, then the agent can infer the information needs associated with

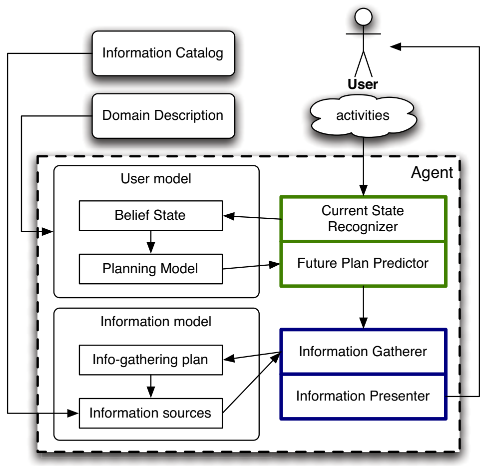

Figure 1: The *ANTIPA* agent architecture.

these planning activities. Third, the agent needs to construct a plan for collecting the information from various information sources. This plan must consider the tradeoff between obtaining the information of which the user is likely to make the most use and satisfying temporal deadlines that certain information must be obtained before a specific time point to be useful. Finally, the agent must decide when to offer certain information to the user based on its belief about the user's current state.

Figure 1 depicts the high-level architecture of an *ANTIPA* agent where the agent processes are contained within the dashed box; the cloud represents the agent's observations; rounded boxes represent data structures; and rectangle stacks represent reasoning tasks. The basic interactions between the user and the agent are: the agent observes some of the user's planning activities; the agent may present information to the user. At deployment time, the agent is supplied with two inputs: a `domain description` representing the user's planning problem (*e.g.*, state-based planning problems, plan libraries, or workflow); and an `information catalog` that describes a set of properties of information sources from which the agent can retrieve information. In the following subsections, the four main *ANTIPA* components are described.

Current State Recognizer: The agent may not be able to directly observe the user's true states nor the actions that the user has taken. In this context, the agent must infer the user's current state from a series of primitive sensory data known as *observations*, *e.g.*, in the student example, possible observations include the keywords that the user types into search engines or the documents that the user opens.

The agent models the user's current state (which the agent cannot observe directly) as a probability distribution over a set of possible states, known as a *belief state*. When the agent perceives a new observation from the user interacting with an environment, Current State Recognizer updates its belief state such that the updated belief state best explains the observations. Specifically, we estimate the probability $p(s|o_1, ..., o_t)$ of that the user is in state s given a sequence of t observations $o_1, ..., o_t$, using a dynamic programming technique known as the forward algorithm [3]. The updated belief state then triggers other components to adjust accordingly (*e.g.*, the agent can determine a set of information to present immediately according to the current belief state).

Future Plan Predictor: Given a belief state, Future Plan Predictor identifies most likely plans from the current belief state and constructs a tree of action-nodes, known here as a *plan-tree*, representing a set of planning paths highly likely to be taken by the user. The algorithm computes an optimal stochastic policy of the MDP user model using the value iteration algorithm [4] where the policy specifies, for each state and action pair, the probability of the user's taking the action from the state. From each state, the belief probability (of that the user is in that state) is propagated, for each action, to the next states (that the user will be after taking the action) using the stochastic policy and the state transition probabilities. In the resulting plan-tree, an action-node includes a *query* for the information that is required for the action (*e.g.*, a database query), the *priority*–the probability that the user will take the action, and a set of *constraints* (*e.g.*, a deadline constraint specifying the time by which the data must be retrieved). This plan-tree is then supplied to the Information Gatherer.

Information Gatherer: Given a plan-tree of predicted information-gathering tasks, Information Gatherer determines (or schedules) when and which information sources to use in order to satisfy the information needs of the user as well as coping with resource constraints (*e.g.*, network bandwidth) imposed by the problem domain; that is, the agent should not interfere with the user's planning activities by overconsuming computing resources. Initially, the information-gathering tasks are ordered by the priorities and the deadlines, ensuring not only the acquisition of the most useful information, but also a timely acquisition of data. In order to accommodate changing information requirements, Information Gatherer must optimize its current schedule incrementally to satisfy newer (thus more relevant) information-gathering constraints. The retrieved data is stored locally until used by Information Presenter.

Information Presenter: The agent directly interacts with the user through Information Presenter, which selects a subset of data from the locally cached data, and presents to the user at appropriate times. When to present which data is determined by the estimated user's future information needs. In order to avoid information overload, Information Presenter must only present data in temporal proximity to the actual need, with a sufficient time for the information to be useful for the action at hand. Additionally, Information Presenter must select an appropriate presentation format when offering information to the user. Finally, user feedback (*e.g.*, whether the presented information has been used) is collected and is provided for the agent as reinforcement in order to allow future improvements on the quality of supplied data.

3 Conclusion

We introduced an architecture for anticipatory agent, *ANTIPA*, that proactively assists cognitively overloaded users through intelligent information management. The *ANTIPA* architecture presents a flexible framework that could easily be customized for various types of intelligent assistant applications including adaptive learning support and assistive living technologies. Future work includes enhancing constraint management regarding the information sources, as well as adding a component for assessing user cognitive overload.

REFERENCES

[1] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie, and A. Mihailidis, 'A decision-theoretic approach to task assistance for persons with dementia', in *Proc. IJCAI*, pp. 1293–1299, (2005).

[2] A. Fern, S. Natarajan, K. Judah, and P. Tadepalli, 'A decision-theoretic model of assistance', in *Proc. of AAAI*, (2007).

[3] L. Rabiner, 'A tutorial on HMM and selected applications in speech recognition', *Proc. of IEEE*, **77**(2), 257–286, (February 1989).

[4] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, MIT Press, Cambridge, MA, 1998.