Active Acquisition for User Modeling in Dialog Systems

Dekai Wu*
Computer Science Division
University of California at Berkeley

Bettina Horster
FB Informatik
University of Dortmund, West Germany

ABSTRACT

A *user model* in a natural language dialog system contains knowledge about particular users' beliefs, goals, attitudes, or other characteristics. User modeling facilitates cooperative adaptation to a user's conversational behavior and goals.

This paper proposes *active* strategies for acquiring knowledge about users. Current systems employ *passive* acquisition strategies, which build a model of the user by making inferences based on passive observation of the dialog. Passive acquisition is generally preferable to active querying, to minimize unnecessary dialog. However, in some cases the system *should* actively initiate subdialogs with the purpose of acquiring information about the user. We propose a method for identifying these conditions based upon maximizing expected utility.

INTRODUCTION

Consider the following dialog with a route consultant system:

User:	Is there an inexpensive motel close to Fisherman's Wharf?
System:	Yes, but they are usually full at this time of the year.
	You might try the Ponderosa Inn, which is about a mile from Fisherman's Wharf.
	There is a cable car stop less than two blocks away.
U:	How do I get there?
S:	Go down the street, then turn right on Powell, at the third light.
	There is a sign on the left-hand-side after half a block.

The system's responses are cooperative, in that they help the user achieve his goals. Not only does the system directly answer the user's queries, but in addition, responds at a useful level of detail and expertise, and volunteers pertinent information that was not specifically requested. This example is representative of the kind of cooperative dialog behavior at which current research efforts are directed [Wilensky et al. 1988, Chin 1988, Mayfield 1989, Wahlster et al. 1983, Kobsa 1985].

Generating cooperative responses involves making many different types of inferences. Some of the inferences needed to produce the responses in the example above are:

- The user has the goal of knowing an inexpensive motel close to Fisherman's Wharf.
- The user assumes the system has this knowledge, and wants the system to communicate it to him.
- The user has the goal of staying in the motel today.
- The user wants to be as close as possible to Fisherman's Wharf.
- The user wants to be able to reach Fisherman's Wharf as conveniently as possible.
- The user's budget constraints outweigh his desire to stay close to Fisherman's Wharf.
- The user has the goal of knowing how to reach the motel.
- The user is driving a car (because he asked for a motel).
- The user does not live in the area, and is probably a tourist.
- The user does not know the area well, and requires detailed directions.

* This research was sponsored in part by the Defense Advanced Research Projects Agency (DoD), monitored by the Space and Naval Warfare Systems Command under Contract N00039-88-C-0292, the Office of Naval Research, under grant N00014-80-C-0732, and the Sloan Foundation, under grant 86-10-3.

Inferences such as these fall in the area of *user modeling*, because they involve inferring and utilizing knowledge about the user's beliefs, goals, attitudes, or objective characteristics. Wahlster and Kobsa [1986] have given the following definition:

> A user model is a knowledge source in a natural-language dialog system which contains explicit assumptions on all aspects of the user that may be relevant for the dialog behavior of the system.

To some extent, determining the user's goals and beliefs is an intrinsic part of understanding an utterance; for example, to understand "Is there an inexpensive motel close to Fisherman's Wharf?" the hearer must recognize the speaker's information acquisition goal, and must recognize the presupposition that the hearer knows the information. Other inferences are made only when needed; for instance, the planning stage requires the inference that the user's budget overrides other constraints.

User modeling affects most tasks performed in a dialog system. Some aspects of user modeling merely improve "user-friendliness", while others are absolutely crucial to communication. The user model helps the system understand the deictic reference in "How do I get there?" because knowing that the user has the goal of staying in the motel makes "Ponderosa Inn" a more plausible referent than "cable car stop". The system can only build an appropriate plan for the user if it knows the user's goals—to find accommodations convenient to Fisherman's Wharf—and constraints—a limited budget. It generates a response at the appropriate level of detail and expertise, assuming the user is a tourist. Thus, the user model improves the system's cooperativeness by influencing understanding, planning, and generation tasks.

Now consider the following interchange:

U1:	How do I get to the center of the bay?
S2:	Why do you want to go there?
U3:	I want to take a picture of the skyline.
S4:	Is it sufficient to drive to Treasure Island, or is it necessary to take a cruise?
U5:	No, I don't want to take the picture from Treasure Island.
S6:	Then you can take a bay cruise tour from Fisherman's Wharf.

This is an example of dialog behavior that is beyond most existing dialog systems. It involves *active* acquisition on the system's part, because the system seeks additional user model information by directly querying the user. With S2, the system initiates a subdialog to determine a more specific user goal, and with S4, the system initiates a subdialog to determine the user's preference between two plans. In contrast, all of the knowledge about the user in the earlier example was *passively* acquired by making inferences from observing the dialog, without intentionally altering the course of the dialog.

Usually, passive acquisition is preferable to active, because unnecessary querying wastes time. However, as the example shows, active acquisition is sometimes useful. A passive system would respond like this:

U1:	How do I get to the center of the bay?
S2:	You can drive to Treasure Island.
U3:	But I don't want to be on land.
S4:	Then you can charter a yacht.
U5:	But I don't have the money.

and so on. Alternatively, the system could suggest all known options:

S2/b:	You can drive to Treasure Island, charter a yacht, rent a windsurfer, swim in a wetsuit, take a bay cruise tour, buy a rowboat, scuba dive, hop on the ferry, or hire a helicopter.

Clearly, neither passive approach is satisfactory.

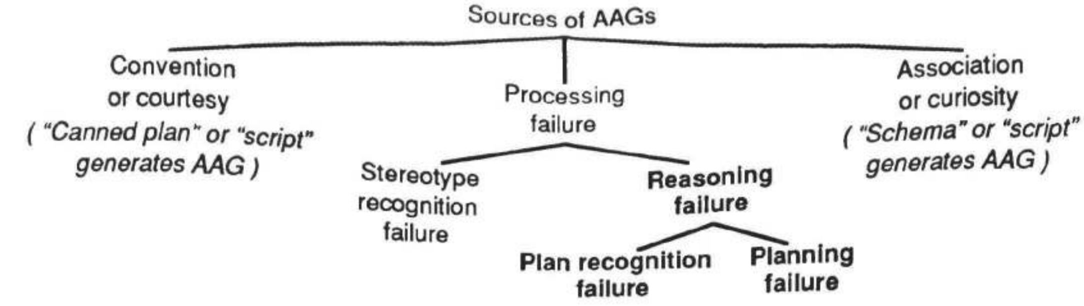

Figure 1. Sources of AAGs.

Dialog systems lack principled means for detecting situations that call for active acquisition during the course of a dialog. Early user modeling systems relied too heavily on active acquisition. GRUNDY [Rich 1979] asks the user a "canned" set of questions at the beginning of a session, and builds a user model from the answers. "Canned" active approaches are valid only for highly restricted domains with little conceptual variation from one dialog to another. Because of this limitation, later systems such as KNOME [Chin 1988], PAGAN [Mayfield 1989], and TRACK [Carberry 1988] concentrate on passively inferring models of the user, an approach that inherently ensures the user model is tied to the particular dialog context. However, although the ideal agent relies primarily upon passive acquisition to meet efficiency and appropriateness considerations, the same considerations dictate occasional *motivated* use of active acquisition.

This work is heavily related to plan recognition research [Allen and Perrault 1980, Litman and Allen 1984], but differs in emphasis. Plan recognition techniques only *recognize* the correlations between discourse structure and conversants' plans; they do not *predict* the most plausible continuation of a dialog given the conversants' plans.

ACTIVE ACQUISITION GOALS

As a useful conceptual notion, we define an *active acquisition goal* (AAG) as a goal held by a dialog agent to actively acquire knowledge about the dialog partner. We assume that in normal operation a dialog system builds a model of the user through passive acquisition, but occasionally adopts an AAG, and therefore initiates an information-seeking subdialog. Thus, the issues to be addressed are: (1) under what conditions should AAGs be generated, and (2) once generated, what are the criteria for adopting or rejecting AAGs?

A broad classification of reasons for generating AAGs is given in Figure 1. Our analysis is limited to the categories in boldface, but there are other potential sources of AAGs. In particular, systems employing stereotypes may generate AAGs in the event that no satisfactory stereotype is recognizable.[1]

Plans for achieving AAGs involve *speech acts* in Austin's [1962] sense that utterances are produced by actions. The most straightforward type of plan initiates a *clarification subdialog* by verbalizing a direct request for information. Subtler plans for actively acquiring information use *indirect speech acts* [Searle 1969] or ask a question whose answer can then be used to infer the desired information.

There are two ways to interpret AAGs. In one view, there are only ordinary acquisition goals—in themselves neither passive nor active—for which some plans involve active speech acts and others involve passive means. According to this view, there are active acquisition *plans*, but AAGs are merely a conceptual shorthand that help us focus on a particular phenomenon. On the other hand, AAGs can be seen as a real class of goals that circumscribe a narrower class of plans than ordinary acquisition goals; that is, the existence of a separate class of AAGs constitutes compiled indexing information for retrieving plans from memory. We take no position on this distinction, in the absence of empirical psychological data.

[1] Some cases where this might happen are: (1) if the only matching stereotypes are over-abstract (i.e., superordinate rather than basic-level categories [Rosch et al. 1976]), (2) if mutually exclusive stereotypes are equally plausible, or (3) if a stereotype is a very close but imperfect match.

Plan recognition failure
Plan recognizer produces no plan of acceptable utility explaining user's speech act
1a → *Generate AAG to identify unknown user goal*
1b → *Generate AAG to identify (and/or correct) unknown user misconception*
Plan recognizer produces more than one "maximum utility" plan explaining user's speech act
-or- Plan recognizer lacking information needed to compare plan utility
2 → *Generate AAG to disambiguate user's intention*
Domain planning failure
Domain planner lacking information needed to compare plan utility
3a-b → *Generate AAG to ascertain needed information*
Domain planner produces more than one "maximum utility" plan
4 → *Generate AAG to determine the user's preference*
Dialog planning failure
Dialog planner lacking information needed to compare plan utility
5 → *Generate AAG to ascertain needed information*

Figure 2. Taxonomy of failure conditions under which to generate AAGs.

REASONING FAILURES AS A SOURCE OF AAGS

Our analysis concentrates on AAGs generated in response to failures or exceptions in a dialog system's normal reasoning processes. This is a highly productive source of AAGs and accounts for dialogs like the example given earlier. We assume three major types of normal reasoning processes in a dialog system: plan recognition, domain planning, and dialog planning. The following paragraphs briefly sketch how failures in any of these processes can lead to AAGs. We then discuss how the failures can be detected, and give rules for generating AAGs in response to specific failure types. Figure 2 summarizes the general failure types.

Plan recognition: The system performs plan recognition to explain the user's speech acts by inferring his underlying plans and goals; as such, plan recognition is an important form of passive acquisition. If, however, the system is unable to construct any plan to explain the user's actions, the system may hypothesize one of two possible causes: (1) an unknown user goal, or (2) an unknown user misconception. In either case, it may then generate an AAG to identify the unknown. On the other hand, if the system can produce two or more plausible explanations for the user's speech act but cannot decide between them, it may generate an AAG to determine some fact that would eliminate all but one explanation.

Domain planning: The system performs domain planning in its area of expertise to produce a solution for the user's goals (e.g., a route plan). If the system is unable to evaluate the utility of the plans it produces because it lacks information about the user, it may generate an AAG to obtain the information. If it produces multiple plans of high utility, an AAG to determine the user's preference may be generated.

Dialog planning: Dialog planning is performed to determine the system's own actions, particularly its speech acts. In some cases where the system is unable to evaluate the expected utility of alternative plans, an AAG to obtain relevant information may be generated.

Detecting Reasoning Failure Conditions

The foregoing failure conditions all involve evaluating or comparing the utility of plans. The dialog-planning cases involve the system's own potential plans, whereas the plan-recognition cases involve plans which the system hypothesizes the user holds, and the domain-planning cases involve hypothetical plans for the user. Thus, to define failure criteria requires a metric for plan utility.

For current purposes, we use a 3-tuple measure, where plan utility is broken down into the following independent attributes:

(1) *Difficulty*: The difficulty of a plan has to do with the difficulty (effort or time required) of the individual steps of the plan, as well as the difficulty of achieving the preconditions (if they are not already satisfied). Estimating the difficulty of a plan, without actually executing the plan, requires some sort of statistical

expectation based on experience.

(2) *Degree of goal conflict*: Any plan satisfies the goal that was planned for, but may also have unintended consequences as side-effects. The side-effects may conflict with other goals. Some goal conflicts may be acceptable, while others may not.

(3) *Over-specificness of the plan's consequences*: Even if side-effect consequences do not conflict with any other goals, a plan should have as few extra consequences as possible, because extra consequences unnecessarily constrain future actions.

These attributes are meaningful only in relation to a particular agent, in the context of his goals and constraints. The utility of a plan cannot be measured in absolute terms.

Given these definitions, the failure condition "plan recognizer produces no plan of acceptable utility explaining the user's speech act" can be detected by establishing minimum standards for each attribute, so that a failure occurs if no plan meets the standards in all three dimensions.

The failure conditions "plan recognizer produces more than one 'maximum utility' plan" and "domain planner produces more than one 'maximum utility' plan" are detected when two plans' utility measures are *non-comparable*. This can occur because a 3-tuple measure of utility establishes only a partial ordering of plans, since two utilities can only be ranked if one is better than or equal to the other in all three attributes. A "maximum utility" plan must be at least equally good as every other plan in all three attributes, and better than every other plan in at least one attribute (not necessarily the same attribute for each plan). If no plan meets the "maximum utility" criteria, a failure condition holds.

Finally, the system may lack information needed to determine values for utility attributes. For example, if the achievability of some precondition is unknown, the difficulty of a plan cannot be computed. This accounts for the remaining failure conditions above. (Note that it is sometimes possible to compare plans even when it is not possible to determine individual utility attribute values. If, for instance, a precondition status is unknown, but both of the plans being compared require the precondition, then the status of the precondition does not affect the comparison.)

Generating AAGs

We now enumerate the rules for generating AAGs in response to reasoning failures.

- *Rule 1a*: If it is not possible to produce a plan of acceptable utility explaining the user's speech act, then generate an AAG to identify an unknown user goal.

- *Rule 1b*: If it is not possible to produce a plan of acceptable utility explaining the user's speech act, then generate an AAG to identify (and/or correct) an unknown user misconception.

These rules are responsible for S8/1a or S8/1b in the following example. The system cannot produce a plan of acceptable utility to explain U7, and adopts an AAG assuming either an unknown user goal or misconception.[1]

U1: How do I get to the center of the bay?
S2: Why do you want to go there?
U3: I want to take a picture of the skyline.
S4: Is it sufficient to drive to Treasure Island, or is it necessary to take a cruise?
U5: No, a cruise isn't necessary.
S6: Then you should drive to the Bay Bridge and take the Treasure Island exit.
U7: What about Angel Island?

[1] Rule 1b is only one case of suspecting a user misconception; detecting and correcting misconceptions is an orthogonal issue which many researchers have addressed [e.g., McCoy 1988, Chin 1988].

S8/1a: Why do you ask?
U9/1a: I also want to visit Angel Island.

S8/1b: There is no bridge to Angel Island, you must take a ferry.

- *Rule 2*: If the highest-utility plans explaining the user's speech act are non-comparable, or the plan recognizer lacks information needed to compare plan utility, then generate an AAG to disambiguate the user's intention.

Rule 2 is responsible for S4 in the following dialog. The system cannot determine whether the user utters U3 to request directions or to convey a wish to eat in Chinatown, because it lacks the information needed to determine the speech act's appropriateness for the user's overall goals.

U1: Where can we eat a good Chinese meal tonight?
S2: There is a good restaurant called Mandarin House on Clement.
U3: Is that in Chinatown?
S4: No, do you want to eat in Chinatown?
U5/a: Yes.
S6/a: The Red Chamber on Jackson is quite good.

U5/b: No, that doesn't matter.
S6/b: To get to Clement, you can take Geary down to 30th and then go north one block.

The following rules are subcases of the general situation where the domain planner lacks information necessary for comparing the plans with the highest expected utilities. The subcases involve different plan utility attributes.[1]

- *Rule 3a*: If, while comparing the over-specificness of the highest-utility domain plans, no plan closely matches the specificness of the known user goals, and the plans have mutually exclusive consequences, then generate an AAG to determine a more specific user goal.

Rule 3a accounts for the following exchange:

U1: How do I get to the center of the bay?
S2: Why do you want to go there?

The system is able to build a number of plans, including driving to Treasure Island, chartering a yacht, taking a bay cruise tour, and windsurfing. Each plan's consequences are over-specific in comparison with the known goal of the user, which is to get to the center of the bay; thus, each plan strongly constrains the possible future actions. Moreover, each plan constrains future actions differently; one results in the user being on Treasure Island, another results in the user being on a boat, and so on. Since the user presumably has a particular purpose for getting to the center of the bay, the plan selected should be the one whose constraints most closely match his specific goal. However, because the system does not know the user's specific goal, it is unable to evaluate which plan's consequences provide the best match.

- *Rule 3b*: If, while comparing the difficulty of the highest-utility domain plans, some plans have a precondition that others do not, and the precondition is neither known to hold nor assumable by default, then generate an AAG to determine the ease of achieving the precondition.

The following example illustrates rule 3b:

U1: How do I get to the Marina?
S2: Do you drive?

---

[1] There is no subcase for the degree of goal conflict attribute because, if the system has no information about a conflicting goal, it is assumed that no conflict exists.

The system needs to discover whether the user can satisfy the precondition of a plan, in order to select the easiest plan.

- *Rule 4*: If the highest-utility domain plans are non-comparable, then generate an AAG to determine the user's preference.

For example, the system asks "Is it sufficient to drive to Treasure Island, or is it necessary to take a cruise?" because the Treasure Island plan is superior in the difficulty attribute, but the cruise plan is superior in the goal conflict and/or over-specificness attributes since taking the picture from the water constrains the angle less than the island.

- *Rule 5*: If the dialog planner lacks information to compare the highest-utility dialog plans, then generate an AAG to obtain the needed information.

In the following example, the system builds several different plans for expressing a location, but cannot judge their difficulty until it ascertains whether the user's knowledge satisfies the plans' preconditions.

U1: Where is Cafe Rigoletto?
S2: Do you know where Symphony Hall is?
U3: Yes.
S4: Cafe Rigoletto is in the alley across the street from Symphony Hall.

EVALUATING AND SELECTING AAGS

The rules above will generate too many AAGs, and most should be rejected for one reason or another. The problem of evaluating and selecting AAGs is a subcase of the same goal selection problem faced by any general dialog planner. The utility criteria discussed above thus also apply to comparing and selecting plans for satisfying AAGs.

If a failure that generates an AAG is encountered, processing continues as normally as possible. In this way, all AAGs potentially relevant to the user's utterance are collected before any decisions are made. The motivation for this delayed-commitment strategy is as follows. The AAGs selected are those which maximize expected utility *with regard to the system's top level goals*, i.e., cooperative goals such as briefness, comprehensibility, and relevance. The maximum-utility AAG (or set of AAGs) cannot be determined *a priori*; this is especially obvious in cases where a single plan satisfies multiple AAGs. For example, S2 in

U1: Do you know good restaurants in this neighborhood?
S2: Yes, are you interested in a particular style of cuisine?

satisfies the AAG to determine whether the user's specific goal is to discover the system's breadth of knowledge or to discover good restaurants (rule 2), as well as the AAG to determine information needed to compare the utility of various restaurant-going plans (rule 3a). S2 is a better response than "Yes, do you want to know some?" or "What style of cuisine?" which satisfy only one or the other AAG.

Moreover, many AAGs are not worth satisfying because the expected utility is lower than that of an alternative non-active-acquisition plan. For example, it is often easier to correct a suspected misconception rather than take the trouble to verify whether the user actually believes it. Another case occurs when the system produces two non-comparable plans for the user, and simply verbalizes both plans briefly, rather than determining the user's preference. Thus, AAGs are evaluated against all the system's goals and not just against themselves.

To collect the set of candidate AAGs, it is useful to maintain a dependency graph that reflects which AAG plans subsume or exclude others. The dependency graph may also include plans for the system's other goals. In this way, poorer courses of action can be immediately pruned when the utilities are comparable, and the subset that subsumes the most goals can be selected.

CONCLUSION

We have presented the beginnings of a theory of active acquisition for building the user model, as an extension of current theories of cooperative dialog planning and plan recognition. The dialog system generates such AAGs in response to failures detected during utility comparisons in its normal reasoning processes. Plans to achieve an AAG are executed when the expected utility exceeds that of alternative courses of action. Although the classification of active acquisition goals is by no means comprehensive, it covers a large proportion of cases where active acquisition is desirable. The communicative efficiency and naturalness of existing dialog systems can be substantially enhanced by the addition of AAG generation and selection capabilities.

Further work to refine the utility metric is needed. One open issue is the computation of the individual plan utility attributes; numeric values will almost certainly be needed, the source of which must be justified. Possibly, n-tuples with more sophisticated attributes are necessary. The definition of utility also needs to be expanded for the case of plans that partially satisfy goal-sets comprising multiple goals.

Another issue is how to reduce spuriously generated AAGs. Rule 5, for example, must be triggered sparingly to avoid recursive application. Sharper appropriateness conditions should be found for the rules.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. Robert Wilensky, Prof. Armin B. Cremers, and Rainer Waschkowski.

REFERENCES

Allen, James F., and Raymond C. Perrault. [1980]. "Analyzing intention in utterances". *Artificial Intelligence* 15: 143-178.

Austin, J. L. [1962]. *How to do things with words*. London: Oxford University Press.

Carberry, Sandra. [1988]. "Modeling the user's plans and goals". *Computational Linguistics* 14(3): 23-37.

Chin, David. [1988]. *Intelligent agents as a basis for natural language interfaces* (Ph.D. Thesis). Comp. Sci. Div., Univ. of California at Berkeley, Report No. UCB/CSD 88/396.

Kobsa, Alfred. [1985]. "VIE-DPM: A user model in a natural-language dialogue system". In J. Laubsch (ed.), *GWAI-84, 8th German Workshop on Artificial Intelligence*. Berlin: Springer-Verlag.

Litman, Diane J., and James F. Allen. [1984]. *A plan recognition model for subdialogues in conversation*. Dept. of Comp. Sci., Univ. of Rochester, TR 141.

Mayfield, James. [1989]. *Goal analysis: Plan recognition in dialogue systems* (Ph.D. thesis). Comp. Sci. Div., University of California at Berkeley, report forthcoming.

Rich, Elaine. [1979]. *Building and exploiting user models*. Ph.D. Thesis, Comp. Sci. Dept., Carnegie-Mellon Univ.

Rosch, E., C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. 1976. "Basic objects in natural categories". *Cognitive Psychology* 8: 382-439.

Searle, J. R. [1969]. *Speech acts: An essay in the philosophy of language*. Cambridge: Cambridge University Press.

Wahlster, W., H. Marburger, A. Jameson, and S. Busemann. [1983]. "Over-answering yes-no-questions: Extended responses in a NL interface to a vision system". *Proc. IJCAI-83*, 643-646.

Wahlster,W., and A. Kobsa. [1986]. *Dialog-based user models*. Dept. of Comp. Sci., Universitaet des Saarlandes, West Germany, XTRA Report No. 3.

Wilensky, Robert, David Chin, Marc Luria, James Martin, James Mayfield, and Dekai Wu. [1988]. "The Berkeley UNIX Consultant project". *Computational Linguistics* 14(4): 35-84.