An Adaptable Dialog Interface Agent Using Meta-Rules

Jung-Jin Lee
Dept. of Computer Science and Engineering, The Catholic University of Korea, BuCheon, Korea
JungJin@songsim.cuk.ac.kr

Jaeho Lee
Dept. of Electrical and Computer Engineering, The University of Seoul, Seoul, Korea
jaeho@ece.uos.ac.kr

Abstract

We examine adaptable dialogue interface agents using meta-rules to engage in to converse with information seeking callers. For an adaptable dialogue interface agent to exhibit the intelligence required in a dialogue with a user it needs the ability to handle unanticipated user input gracefully in a proper context, that is, it can support ad hoc requests from a user, maintain the topic of discourse, and keep a user from digressing away from the subject. Toward this goal of participation, we propose a plan-based dialog model that aims at achieving a higher level of adaptability by utilizing a set of meta-rules. We use meta-rules as a strategy of controlling an irregular discourse; these can be applied to unexpected responses from a user to analyze whether the responses are incomplete, ambiguous, or inexact in relation to the expectation. The proposed model will allow us to build a dialog-capable interface agent that functions as a middleman between users and information source agents such as database management systems.

Introduction

The decreasing costs and increasing availability of information processing, storage and communications are also increasing the range of people who need to make direct use of computing systems. Intelligent interface agent development is one way to mitigate user behavior/application complexity and information overload through human-computer interactions by extracting and learning knowledge about users. Heightened user interest and diversity, coupled with the rise in the amount of information available, also has stressed the need of practical real-time agents that are autonomous and adaptive enough to deliver useful information by recognizing users' intentions through human-computer interactions. This work focuses on building an adaptable dialog interface agent that functions as a middleman between a user and an information source. The field of Human Computer Interaction (HCI) proposes several kinds of models for predicting aspects of user behavior. Although these models deal with user behavior with interactive computer systems, they do so rather poorly. HCI should be designed to be versatile enough to handle a broad range of interactions between a man and a machine, rather than depending on a pre-defined user model. A goal in HCI is to develop cooperative systems: for a system to cooperate with a user, it must be able to determine what the user is trying to accomplish; that is, it must recognize user plans.

For a system to exhibit the intelligence required in a dialogue with a user it needs the ability to understand and reply properly to a wide range of responses. Additionally, if a system fully participates in a dialogue, it can support ad hoc requests from a user, maintain the topic of discourse, and keep a user from digressing away from the subject. Toward this goal of participation, we focus on handling a user's answer to a question posed by a system. By its nature, a dialogue-capable interface needs to be modeled dynamically. The adaptability of a system is essential when the interaction between a user and the system is a dialogue. To be flexible and incremental, a system should be able to adapt itself by reconstituting a plan. We use meta-rules as a strategy of controlling an irregular discourse; these can be applied to unexpected responses from a user to analyze if the responses are incomplete, ambiguous, or inexact in relation to the expectation. Further adaptability in the system is provided by the process of plan creation which enhances the plan library over time. In this work we model adaptable dialog interface agents in information operator domain in the States.

A Motivating Example

Given below is a transcript of a dialog recorded while conversing with a Connecticut Information operator in the States, who has been reached by 411 from Storrs, Connecticut. It illustrates how adaptable an information operator can be when s/he is encountered with an unusual request. Note that the caller's initial response is unusual because Amherst is not a city in Connecticut.

```
1) INFORMATION:  What city ?
2) CALLER      :  Amherst.
3) INFORMATION:  Amherst, Massachusetts ?
4) CALLER      :  Yes.
5) INFORMATION:  Where are you calling from ?
6) CALLER      :  Storrs.
```

```
7) INFORMATION:  Storrs, Connecticut ?
8) CALLER      :  Yes.
9) INFORMATION:  You must dial 1-413-555-1212
```

In item 1, the system initiates a dialogue with a default plan, while expecting a city name among 169 cities in Connecticut. If the response fulfills the expectation, then it continues the plan.

In item 2, the response results in Unsuccessful Expectation Fulfillment because the city Amherst is not in CT. the system invokes meta-rules to handle the failure by trying to determine what plan the user is trying to follow.

In item 3, the system determines (via information in its database) that there is an Amherst in Massachusetts, and that it knows a plan for getting telephone information in Massachusetts (from its plan library). PIUM hypothesizes the user's reply and attempts to get a confirmation from a user with "Amherst, MA ?", using the HYPOTHESIZE meta-rule.

In item 4, when the hypothesized intention of a user is confirmed with a user's reply, the action part of the meta-rule is activated. While PIUM is searching for an applicable plan for current situation from a plan library as an action part of the meta-rule, the result of retrieving a plan library shows that there are two different plans available depending upon whether the target's area is equal to the caller's area.

In item 5, a new query by PIUM is created and asked to know a caller's location for clarifying between two plans. The system recognizes that additional information is needed, the current plan is modified by adding new query and this modified plan will be stored as a new plan. With this plan creation, the system then learns and is enforced for the future adaptation.

In item 6, when PIUM receives "Storrs, CT", it decides that OUT-OF- STATE information request plan is suitable over LOCAL information request plan .

In item 7, PIUM concludes the case by responding with the data retrieved from the database for an area code of Amherst.

This work focuses on proposing a model that can account for generating the system's subsequent actions in proper context once specific expectation-response relationship type is identified.

The Model

Limitations of the knowledge base is an important problem in AI, for example, having a closed world or an open world model. However, considering the problems in both of these models [1], leads to the point that the ideal system needs to combine the best of both. Based on this idea, PIUM uses an open world model as it serves, while it starts with a closed world model.

When a system does not have complete information about an area, an expectation failure will likely occur during a dialogue. The ideal system needs to handle the expectation failure based on the extent and limit of available knowledge, what facts are relevant in a given situation, how dependable they are, and how they are to be use. This supports the use of meta-rules.

Task-Oriented Plans

An important premise of the proposed model is that one can identify a set of task-oriented plans that are specific to certain routine tasks. Each task-oriented plan is a description of a sequence of actions or subtasks that the agent should carry out to achieve a certain goal. PIUM starts with an initial set of three task-oriented plans. Without having a rich set of user models, it is guided by knowing where it lacks knowledge and where a user has a misunderstanding. As an initiative interface agent, PIUM initiates a dialogue with a default plan instead of waiting for a user to submit a request. The users can benefit from the guidance of the system which draws the relevant range of a task-oriented plan in a domain. The plans are used to synthesize an instance of a task- oriented plan. Each different interaction is remembered. Each plan related to the instance is also remembered for future adaptation.

Figure 1 includes two plans that are designed to model two different dialog patterns which an information operator might typically engage in to converse with information seeking callers. Plan One lists four subactions that can be considered prescribe the information operator's most common activities. It thus constitutes the default plan. Plan Two accounts for the case where the caller is requesting the target's number that exists outside of the caller's resident state. In this case, the information operator usually refers the caller to the information operator of the target's state.

One novel feature included in the plan given in Figure 1 is that each system's question specified in each step is accompanied by an expectation. Each expectation describes the nature of potential answers to the pending question. Problematic cases occur when the responses does not match, either directly or indirectly, the expectation. The idea proposed in this work is to use a set of meta-rules to handle these problematic cases.

Meta Rules

PIUM uses meta rules as control knowledge to help it decide the next plan to be followed and lead a user's responses to get the final goal given an irregular request. In PIUM, meta rules are primarily used in dealing with a system's expectation failures. These meta rules are utilized to infer applicable plans and, if necessary, plan creation. We assume the following scenario and notations. The system is equipped with a set of task-oriented plans, denoted by P_1, P_2, ... At one point, the system asks a question to the user following what has been prescribed in the j^{th} action A_c^j of some plan P_c. The expectation accompanying A_c^j is denoted by E_c^j. Once the user response, denoted by I, is input, the system compares I against E_c^j to determine the relationship between I and E_c^j.

Each meta-rule is a triple *<IF, THEN, SUBSEQUENT-ACTION (SA)>*, where the *IF*

```
DEFAULT PLAN  (Plan-One)
Assumption : The caller's target resides at Connecticut.

Action-1
Goal :    Get the city name where the target resides.
       [DB: Specify the city-directory relation]
ASK :     What city ?
EXPECT :  A city name among 169 Connecticut cities.

Action-2
Goal :    Get the target's identifier.
       [DB: Specify the selection condition for name]
ASK :     Yes ?
EXPECT :  A name of a person or a business.

Action-3
Goal :    Retrieve the target's phone number from DB.
DB action: RETRIEVE phone #  FROM
   ?city-directory  WHERE name = ? target
EXPECT :   Phone number(s) for the target

Action-4
Goal :    Provide the target's telephone number
RESPOND : ANSWER (DB ACTION)

PLAN - Two: Seeking Out-of-State Info. Operator
Assumption : The caller's target resides at Connecticut.

Action-1
Goal :    Get the city name where the target resides.
       [DB: Specify the city-directory relation]
ASK :     What city and state ?
EXPECT :  Non-CT city name and its state name

Action-2
Goal :    Retrieve the target's area code from DB.
       [DB: Specify the selection condition for name]
DB ACT:   RETRIEVE area-code FROM
  ?state-directory WHERE city-name = ? target
EXPECT:   Area-code for <city-name, state-name>

Action-3
Goal :    Provide the area-code and 555-1212
RESPOND : ANSWER (1-?area-code-555-1212)

Plan-Three: Local Information Request
Assumption:  The target's location is same
     as the caller's state

Action-1
Goal :    Get the city name in which the caller is interested in.
          [DB: Specify the city-directory relation]
ASK :     What city ?
EXPECT :  A city name in conjunction with a state name.

Action-2
Goal :    Provide the local information operator number.
RESPOND:  Dial 1-555-1212.
```

Figure 1: Plan with Expectations in Actions

part specifies the rule invoking preconditions, and the *THEN* part prescribes the activity the system should initiate as a consequence of the rule invoking. The *SA* part is a subordinate IF-THEN structure that describes the type of subsequent condition and action pair the system should employ to handle the user's counter response to the system's action specified in the THEN part.

M1. Hypothesize the user's goal and Verify: There are cases when the user response does not satisfy the pending expectation but may partially matches an expectation embedded in some action of a plan that is different from the pending one. If the match to a different expectation were complete, switching to the corresponding plan could be done, provided such a switch does not contradict any exchange that took place between the user and the system earlier. If the match is incomplete, the plan switch can only be contemplated. Thus the system first hypothesizes a plan switch, but prior to actually doing so the system attempts to verify that this is the correct action to take.

IF I does not match E_c^j, and
I partially matches E_k^l of P_k, $k <> c$,
THEN hypothesize a switch to the plan P_k and
verify whether I can match E_k^l completely.
SA if verification is positive, switch from P_c to P_k; otherwise, invalidate the plan switch.

Here is an example of hypothesizing intention using above meta rule.

```
CALLER          :  Amherst
INFORMATION     :  Amherst, MA ?
```

While the system expects a city name in Connecticut, and the user responds 'Amherst', which is not a city in Connecticut, the system finds possible information by retrieving from the database. When the information is derived from the database, not given by the user, the system needs to confirm the hypothesis with the user. In this case, the system hypothesizes that Amherst refers to Amherst, MA.

M2. Choose one from multiple applicable plans: There could be cases when the input matches directly or indirectly multiple expectations specified in steps of different plans. In this case, the system needs to construct a counter question whose answer could discern a particular plan among many applicable ones. The method of computing the counter question is based on differences in assumptions of the matching plans.

IF I does not match E_c^j, and
I completely matches E_k^l of P_k, $k <> c$, and
...
I completely matches E_n^m of P_n, $n <> c$,
THEN compute and ask a question Q that can
discern E_k^l ... E_n^m.
SA if the response allows to choose one among E_k^l ... E_n^m, perform the plan switch, otherwise, invalidate the plan switch.

M3. Clarify partial matching case: There are cases when the user response matches the expectation partially. In this case, the system constructs a clarification counter question. A few techniques can be employed. One approach is to produce an argument interrogative. Another approach is to form a yes-no answer choice question. The last two approaches are possible only if the system can compute possible additional component that can make the original response complete or unambiguous.

IF I partially matches E_c^j,
THEN compute and ask a question Q whose
answer combined with I can match E_c^j.
SA if the response enables system to determine a complete match, continue on with P_c, otherwise, (no specification is needed).

M4. Rephrase the original question with additional information: there are cases when users provide an information which does not satisfy the current expectation but satisfies instead an expectation associated with one of the upcoming actions of the current plan. While holding on to the current plan, the system needs to repeat the original question. In doing so, instead of merely repeating the same question, it can incorporate the prematurely given answer in a rephrasing of the original question.

IF I does not match E_c^j, and
I matches E_c^u $u > j$,
THEN compute and ask a question Q that rephrases the question specified in A_c^j by incorporating I, and indicate that E_c^u has been fulfilled.
SA (no specification is needed)

An example of rephrasing the unanswered question

```
INFORMATION     :  What city ?
CALLER          :  The last name is Williams,
  and the first initial E.
INFORMATION     :  Which city is E. Williams for ?
CALLER          :  Oh, this is for BERLIN.
INFORMATION     :  The phone number is 264-8117.
```

Referring to the default plan in Figure 1, PIUM initiates a dialogue and receives a response 'E. Williams' from a user. Since a name of a person or a business is expected after a system notices a region of a caller's target, the system rephrases the request with the user's response to get the unanswered question, that is, a target of the caller.

Plan Creation (as mentioned in item 4 and 5 of the motivating example) is similar to Collins' work [2]. In the situation in item 4 a target's location is in Massachusetts and a caller's location is unknown. This causes a plan failure and presents a possible linkage to plan-two or plan-three. Therefore, a new query is made to get missing information to distinguish between two applicable plans. Based on the user's response, if a target's area is the same one as the user's area, then plan -three will be selected, otherwise, plan-two. In order to distinguish between plan-two and plan-three, the following actions are made. Since neither plan can handle current features alone, after the new query is created, the actions are included in a pending plan and the modified plan is stored in the plan base as a new plan for future adaptation. In this way, the system's discourse models are enhanced as PIUM has more experiences. A newly created plan including new actions following is fundamentally a merge of plan-two and plan-three.

```
Action-n
Goal :  Get the location of the caller.
        [Compare the caller's location with
 the target's location]
ASK :   Where are you calling from ?
EXPECT :  A city name in conjunction with a state name.

Action-n+1
COND:   If the caller's location <> the target's location,
           Then  PLAN Two
COND:   If the caller's location = the target's location,
           Then  PLAN Three
```

Prototyping and Testing

The prototype system called PIUM (Plan-based Interface Using Meta-rules) has been built. This section represents control strategy, plan representation, and meta-rule representation in PIUM.

Control Strategy

As an initiative interface, PIUM opens a dialogue with its own questions pre-stored in a default task-oriented plan. During the matching process of a system's expectation to a user's response, meta-rules are invoked to handle a current situation if an expectation failure occurs. Handling incomplete information is pursued to infer a more suitable plan when multiple plans are applicable to settle a current situation. It is done by identifying the crucial information needed to distinguish one plan from the other and making a new query to obtain the information. A new plan is created based on the added information for later use.

Plan Representation

Each plan contains a sequence of actions that the system should initiate to engage in a dialogue with the caller as well as to perform related database access tasks. Each action is made of four parts: goal, ask, expect and done. The goal part specifies the type name of an identifier that is to be elicited from the user. The ask part specifies the logical form of question that should be presented either to the caller, or to the underlying database. The expectation part keeps the data structure that minimally represents the nature of the expectation the system should have with respect to the question specified in the ask part. The done part will be an indicator to show how its action is executed. This structure is continuously modified as the system progresses.

Like most plan recognition systems, PIUM has task-oriented plans which have a rigid temporal ordering of actions. However, utilizing meta rules allows a flexible temporal ordering can be. When an expectation failure happens, PIUM can hypothesize this situation and the confirmation of the hypothesis from a user can bring another plan. This demonstrates an attempt to check the validity of the pending plan once it is chosen.

Meta rule Representation

The representation language used for meta rules is a predicate logic that associates predicates and events in the Lk language form which attempts to combine the advantages of the foundations of a formal logic with the structural flexibility of frames [3]. A meta rule in PIUM consists of a representations of situations and a handling strategy. Each meta rule includes what PIUM knows about a current situation, how PIUM will handle this situation, and what PIUM knows as possible side effects. Any aspectual of a relation among meta rules will be constrained to be selected by a current situation. The structure of meta rules allows branching and

iteration. 2. In a current implementation, the order of meta rules are not so critical in the process of a recovery, since all meta rules are inspected. Moreover, since each different situation has a different strategy, no conflict among meta rules is expected for handling a failure case. Although the efficiency of the order of meta rules is not so critical with limited numbers of rules at this point, it stands to reason that checking a *rephrase* meta rule is preferable in that it allows PIUM to remain in the same plan.

Related Work

Our system is an initiative interface, that is, an interface that actively participates in a dialogue. The work of PIUM encompasses the research of mainstream plan recognition and user modeling: providing more intelligent interfaces for interactive environments.

The work by Litman and Allen [4] discuss how a plan recognition model can account for a variety of subdialogues such as clarification, corrections and topic changes. The work by Lambert and Carberry [5] presents a belief model in an attempt to identify negotiation subdialogue structures. Query clearing house approaches as a mediator between the end users and databases also have been done in [6][7]. None of these work, however, deals with handling of expectations which are inherent in argument interrogative types of questions. The lack of addressing expectations in the literature is due to the different nature of problem the various work attempt to model. For example, Litman and Allen aim at accounting for flow changes in relatively general types of dialogues and thus present a model based on domain plans and discourse plans. Most of above work allow interaction between Human and Interface by weighting more focus on inferring user's intention in a passive way.

The work presented here offers two-way communication by actively participating in a dialogue and addresses a narrowly focused range of communication failures. Specifically, it aims at designing a computational system that is capable of handling a user's unexpected response to an argument interrogative type of questions posed by the system. Allowing users to respond to system-initiated questions in an unconstrained way entails careful analysis of relationships between system expectation and user response. Regularities exist in handling expectation failures of user responses. The work presented here is an attempt to model the regularities in terms of meta-rules which should be applicable on top of task-oriented plans.

Conclusion

This work focuses on building an adaptable dialog interface agent that functions as a middleman between a user and an information source. If we are not to unduly constrain the user's input to the interface agent, the agent should be able to handle unexpected responses from the user in a proper context. The presented model assumes that the system-initiated questions can originate from a sequence of the predetermined actions embedded in a task-oriented plan.

However, the presence of ad hoc user responses increases uncertainty, but such uncertainty as can be recognized can be handled by asking the appropriate questions instead of performing expensive inferences. Unlike most existing recognition approaches, each of agent's question is accompanied by an 'expectation' describing the nature of the potential answers. A set of meta-rules has been proposed which can then be used to direct the system what to do when the response does not completely fulfill the pending expectation. The number of meta-rules can be vary depending on the domain to handle more detail level and to be suitable within broaden domains, while the meta rules are general enough to be domain-independent.

These ideas are applicable to recognition systems that assist users in particular problem domains, where the system has more (and better) information about the problem domain than the user. It is advantageous for a system to maintain relevance here; the goal is that the system provide useful information without confusing the user with unnecessary details.

This work has pointed out the need for some enhancements: For one, we need a richer representation language to be able to handle more complicated situations and plans. Additionally, we need a more robust way of inferring the closest plan, as the number of plans increases.

Acknowledgment

We thank to Dr. Dong-Guk Shin at the University of Connecticut for his insight and advise on this work. This work was supported by the Catholic University Settlement Research Fund Granted in the Program Year of 2001 and by the KOSEF through the Advanced Information Technology Research Center(AITrc).

References

[1] D. Chin. KNOME: Modeling What the User Knows in UNIX. In *User Models in Dialog Systems*, Springer-Verlag, New York, 1989.

[2] G. Collins. Plan Creation. In *Inside Case-based Reasoning*. Lawrence Erlbaum, 1989.

[3] L_k: A Language for Capturing Real World Meanings of the Stored Data. In *Proc. of the Int. Conf. on Data Engineering*, pages 738-745, Kobe, Japan, 1991.

[4] D. Litman and J. Allen. A Plan Recognition Model for Subdialogues in Conversation. In *Cognitive Science*, 11:163-200, 1987.

[5] L. Lambert and S. Carberry. Modeling Negotiation Subdialogues. In *Proc. of the 30th Annual Meeting of the ACL.*, pages 193-200, 1992.

[6] W. Kim and J. Seo. Classifying Schematic and Data Heterogeneity in Multidatabase Systems, *IEEE Computer*, December 1991.

[7] G. Wiederhold. Mediators in the Architecture of Future Information Systems, *IEEE Computer*, pages 38-49, March 1992.