

NATURAL LANGUAGE INTERACTION WITH DYNAMIC
KNOWLEDGE BASES: MONITORING AS RESPONSE+

Eric Mays, Sitaram Lanka, Aravind K. Joshi, and Bonnie L. Webber

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, Pennsylvania 19104

ABSTRACT

In this communication, we discuss an interesting aspect of natural language interaction with dynamically changing knowledge bases - the ability to monitor for relevant future changes in that knowledge. We also indicate the status of our current work in this area and the overall goals of our research on question-answering and monitoring dynamic knowledge bases.

1. INTRODUCTION

The overall goal of this research is support for natural language interactions with a dynamically changing knowledge base.* One significant aspect of this environment is the added dimension along which a system can respond: that is, it can take the initiative and attempt to provide additional information presumed relevant to the user. In particular, the system can offer to inform the user of any information it may learn in the future, that is relevant to the user's query. For example, (S: System, U: User)

U: Did A T & T common stock go up today?

S: Yes. Shall I let you know if it keeps rising?
or

S: No. But shall I let you know if it starts a comeback?

In order to take such initiatives the system must be competent enough to monitor only for conditions which might possibly occur. Of course, this monitoring behavior must also be relevant to the user's intentions. The major issues that must be addressed here are as follows:

1. identifying those properties of actions, events, and processes that influence what monitoring behavior is appropriate, e.g. what aspects of

⁺This work is partially supported by a grant from the National Science Foundation, MCS 79-08401.

* For some work on cooperative responses in static knowledge bases, see [6], [8], and [9]. Some issues concerning updatability are discussed in [5]. For some work on temporal aspects of knowledge bases, see [1], [2], [4], and [10].

the world may change and hence, what updates to the knowledge base are expected or possible, how those aspects of the world may change and hence, what constraints exist on updates, etc. The latter are also necessary for maintaining the consistency of the knowledge base as updates occur. Given that the events, processes and actions that the knowledge base reflects may not (or perhaps should not) be modelled as independent, knowing exactly what needs to be monitored will involve addressing the frame problem as well [7].

2. developing an adequate knowledge representation capable of representing such properties. (Since our primary concern is appropriate monitoring behavior, a full representation of actions, events and processes may not be necessary).

3. developing a language for expressing monitors.

4. identifying natural language cues to the kinds of additional information a user would find relevant**.

5. designing and implementing a system which exhibits appropriate monitoring behaviors.

We have begun research in these five areas, and the brief examples which follow illustrate not only our current status but also the larger goals of the research on question-answering and monitoring over dynamic knowledge bases.

2. EXAMPLE

For the purpose of illustration, we assume an entity-relationship type of model [3], as illustrated in Figure 1. Entity sets are shown in rectangles, relationships set in diamonds with arrows from participating entity sets, and attributes

** Complete knowledge of the user's intentions would clearly help to frame appropriate monitoring behavior. However achieving this unambiguously is a formidable if not impossible problem. On the other hand, it is possible for the system to presume a certain intent and then exhibit the corresponding monitoring behavior. What is necessary here is that the system inform the user of its presumptions so that s/he had the opportunity to correct them.

with dotted links to the entity sets. So, for example, this particular schema shows that some users may run some programs, some have a name and a password, some users register for some courses, etc. The crucial aspect is that the possible changes in the world represented in this database (i.e., its updatability) can be encoded in the schema. Updatability here corresponds to the ability to update instances of an entity set, relationship set or attribute.

One method of encoding updatability information is by means of the form (square brackets indicate choice and angle brackets, variables).

(UPDATABILITY <entity> [ADD, NO-ADD] [DELETE, NO-DELETE])

(UPDATABILITY <relationship> [ADD, NO-ADD] [DELETE, NO-DELETE])

(UPDATABILITY <attribute> [UPDATE, NO-UPDATE])

These assertions basically state whether or not additions and deletions can be made to a set and whether or not the value of an attribute may be updated. The following updatability assertions might be made for the schema shown in Figure 1.

(UPDATABILITY USERS ADD NO-DELETE)
 (UPDATABILITY USER-NAME NO-UPDATE)
 (UPDATABILITY USER-PASSWORD UPDATE)
 (UPDATABILITY PROJECTS NO-ADD NO-DELETE)
 (UPDATABILITY PROJECT-NAME NO-UPDATE)
 (UPDATABILITY PROGRAMS ADD DELETE)
 (UPDATABILITY PROGRAM-NAME NO-UPDATE)
 (UPDATABILITY COURSES NO-ADD NO-DELETE)
 (UPDATABILITY COURSE-NAME NO-UPDATE)
 (UPDATABILITY WORK NO-ADD NO-DELETE)
 (UPDATABILITY RUN ADD DELETE)
 (UPDATABILITY REGISTER ADD DELETE)

However, such assertions are insufficient since updatability may itself change over time. For example, at many universities, a student's ability to change his/her registration status depends on whether it is registration period, break, drop-add period, etc. Thus the updatability of the registration relationship set is better represented as

PERIOD	UPDATABILITY
Pre-registration	ADD NO-DELETE
Break	NO-ADD NO-DELETE
Registration	ADD DELETE
Drop	NO-ADD DELETE
After drop	NO-ADD NO-DELETE

Now consider the following question and the various periods during which it may be posed:

U: "Has John registered for CSE110 yet?"

(During pre-registration or registration (add)...)

S: "No, but I'll tell you when he does."

(During break, drop, or after drop (no-add) ...)

S: "No".

Contrast that behavior with

U: "Is John still registered for CSE110?"

(During registration or drop (delete)...)

S: "Yes, but I'll let you know when he isn't."

(During pre-registration, break or after drop (no delete)...)

S: "Yes."

The rule in operation here is to set up a monitor when a situation expected by the user might possibly occur. While the rule is clearly oversimple, it does lead to the relevant behavior. A crucial assumption here is that the length of interaction is short enough that updatability will not itself change during the interaction. Although this is adequate for displaying such behaviors during short interactions, say a terminal session of a few hours, it is clearly not the case for those job functions that span the entire life of the knowledge base. The determination of the interaction length is therefore critical in the inference of appropriate monitors.

In addition to illustrating how outside properties shape appropriate monitoring behavior, the previous examples illustrate two syntactic forms (is/still, has/yet) that suggest the user's interests and hence what monitor might be relevant. That is, "yet" conveys that an event is expected to occur and "still", that a state is expected to continue holding. Appropriate monitoring behavior then in the first case involves an offer to announce the event (or such an event) when it occurs (e.g. "but I'll tell you when he does"), and in the second case, an offer to announce when that state no longer holds (e.g., "I'll let you know when he isn't"). In the absence of such cues as "yet" and "still", the system might query the user about the possibility of setting a monitor. However, to allow maximum flexibility, we include an extra slot in the updatability assertion indicating whether a monitor should be set, not set, or a question posed to the user.

3. SUMMARY

Natural language interactions with dynamically changing knowledge bases, invite types of co-operative behavior beyond those needed for interacting with static knowledge bases. We have briefly indicated the status of our work in this area and our larger goals for the research on monitoring aspects of dynamic knowledge bases.

REFERENCES

- [1] Anderson, T.L., The Database Semantics of Time, Ph.D. Dissertation, Computer Science Department, University of Washington, 1981.
- [2] Bruce, B.C., "A model for temporal reference and its realization in a question answering program", Artificial Intelligence, Vol. 3, 1977.
- [3] Chen, P.P.S., "The entity-relationship model -- towards a unified view of data", ACM Transactions on Database Systems, Vol. 1, No. 1, 1976.
- [4] Fagan, L.M., VM: Representing Time-Dependent Relations in a Medical Setting, Ph.D. Dissertation, Computer Science Department, Stanford University, Stanford, CA., 1980.
- [5] Joshi, A.K., "Mutual beliefs in question-answer systems", in Mutual Knowledge (ed. N. Smith), Academic Press, to appear in 1982.
- [6] Kaplan, S.J., Cooperative Responses From a Portable Natural Language Data Base Query System, Ph.D. Dissertation, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA., 1979.
- [7] McCarthy, J. and Hays, P.J., "Some philosophical problems from the standpoint of artificial intelligence", in Machine Intelligence (ed. B. Meltzer and D. Michie), Vol. 4, Edinburgh University Press, Edinburgh, 1969.
- [8] McKeown, K.R., "Generating relevant explanations: natural language responses to questions about data base structure", Proc. First Meeting of the American Association for Artificial Intelligence (AAAI), Stanford, California, 1980.
- [9] Mays, E., "Failures in natural language systems: application to data base query systems", Proc. First Meeting of the American Association for Artificial Intelligence (AAAI), Stanford, California, 1980.
- [10] Sernadas, A., "Temporal aspects of logical procedure definition", Information Systems, Vol. 5, 1980.

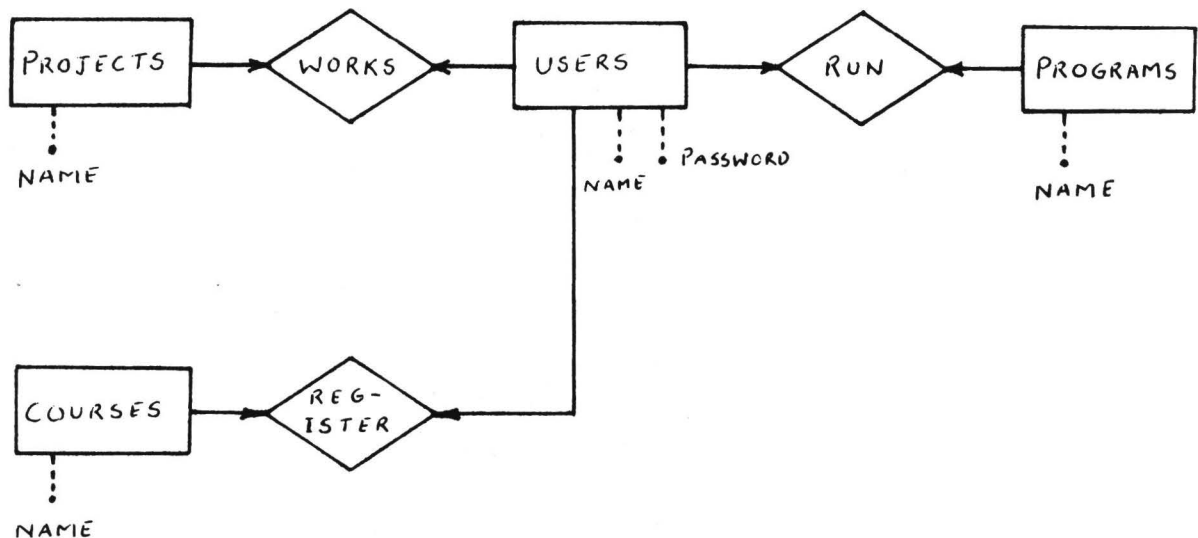


FIGURE 1