INFORMATION RETRIEVAL AND THE QUERY LANGUAGE

Abraham Waksman
Stanford Research Institute
Menlo Park, California

ABSTRACT

This paper addresses itself to the information retrieval problem of a manager within an organization. We feel that the approach taken in specifying such a system is sufficiently general to be suitable for application to other system specifications. It is assumed that the manager wishes to interrogate a system containing a data base that relates to his task of making decisions in the course of running his part of an organization.

To meet the manager's needs we have attempted to define the models a manager uses in making decisions. We have further made an approach toward problem definition for the manager.

To interpret a manager's queries stemming from encountered or anticipated problems, we have attempted to define an inferential system. The inferential system has a further task of specifying the retrieval of information that might exist only in an implicit form. Finally, we demonstrate an approach toward the design of a query language for the manager, which could become "natural" to him after short experience.

INFORMATION RETRIEVAL AND THE QUERY LANGUAGE

Abraham Waksman

I. INTRODUCTION

A formal language used as a query language for data retrieval is limited as well as limiting. It is limited in its ability to convey to the system an accurate translation of the user's intent. Further, it is limiting to the user in terms of developing fully his retrieval needs.

Much has been said about the use of natural language for a query language, and appropriate mechanisms have been proposed, but the fundamental problems in developing a practical system capable of communicating with a user in natural language are still with us. It is not clear whether or not they will disappear in the near future. Such problems include the selection of the proper context when anticipating an oncoming message, and the problem of selecting a relevant subset of the stored information when searching.

We propose in this note an approach to the design of a retrieval system that will allow the user a great degree of freedom in his ability to express his intent, and that will further facilitate the handling of user requests in an efficient manner.

Our proposal hinges on the following steps in the design approach:

(1) Define the models used by the user when interacting with the system.

(2) Define the problems of the user in terms of his models.

(3) Define an inferential system that interprets user queries.

(4) Define an internal system model capable of being consulted in interpreting the user's problems.

(5) Define a conceptually based query language capable of acting as the interface between the user and the inferential system.

To explain the above notions, we will use the hypothetical case of a system devised for a manager within an organization. Our motivation is to display a true information need beyond the contrived need of experimental systems or the highly specialized need of some pragmatic system.

We feel that the world of the manager carries with it a great degree of generality, that is, we see the solutions to his problems as valuable to many other types of users.

II THE MODELS OF THE MANAGER

Managers use several models in dealing with their environment.[1] Our purpose in considering them is to define an internal presentation that can be incorporated as functional elements of a retrieval system.

A. The Historical Model (policy maintenance)

Abelson[2] says: "When observing individuals making decisions, we note that some may never conceptualize anything more abstract than the themistic short-run dynamics of 'The way things are,' without any sense or expectation of evaluation, of change, of things being different."

When observing an ideal manager, we claim that "the way things are" could be considered only one of his models of the world. We call such a model the historical model; it is basically the representation of the current state of affairs and reflects the belief that recent past experience is the best estimate of the short-term future. In most instances, recent past experience also represents the desired situation.

*The work reported here was sponsored by the Office of Naval Research under Contract N00014-71-C-0210.

As an example of the historical model of the manager, let us consider the manager and his activity aimed at executing an established policy.

Consider the manager of an aircraft maintenance facility. We assume that this manager will get a monthly repair action summary, and a monthly maintenance cost summary. We also assume that the manager's policy is such that he will demand to be alerted when the present monthly summary deviates from last month's summary.

B. The Planning Model (policy making)

The planning model could be considered as the "projected state of affairs," i.e., the projection of future performance. Seldom is the planning model self-imposed; most of the time deadlines and schedules have been set by the manager's superiors. It is assumed that the manager would like to be alerted when present performance deviates from projected performance. Furthermore, when near-future extrapolations are being made, it is assumed that the manager checks them against projection or planned performance so that if the present rate is not satisfactory, it represents a problem within this model.

C. The Imposed Model

The imposed model is not the manager's model at all. It is a model of the manager's peers or, more often, a model of the manager's subordinates. According to the model, the subordinates or peers may have a problem for which they solicit the manager's help. Problems arising in personnel assignments, personal conflicts, and unsatisfactory problem solving by subordinates are included.

D. The External Model

This too is, strictly speaking, not the manager's model. Typically, it is the model of an outside, often competing organization. Nevertheless, the manager needs to pay attention to problems that arise from such a model--for example, to compare rate of productivity or costs of operation.

III THE PROBLEMS OF THE MANAGER

To better understand the manager's problem, we need first to realize the complexity of the manager's world. Not only do managers vary as individuals, but also these individuals assume managerial roles that are not suited to their individual capabilities but dictated by the organization to which they belong.

The basic task of the manager is to find the problem he must solve. He must therefore engage in, initiate, and maintain the process of problem finding as well as that of problem solving.

The manager must correctly identify the problem to be solved. He must assess the cost of analysis and the potential return. He must allocate resources to questions before knowing the answer. We can therefore define the manager's problems in general as dealing with the difference between an existing situation and a desired situation as related to the previous models.

Accordingly, problems exist when:

(1) Within the historical model, deviations are discovered between present performance and the performance of the immediate past--i.e., when a change is discovered from the steady state.

(2) Within the planning model, present performance does not meet expectations according to previous plans, or present trends indicate a deviation from projected future performance.

(3) Within the imposed model, a subordinate is apprehensive about meeting his goal.

(4) Within the external model, a competing organization's performance, products, prices, etc., reflect unfavorably on the manager's organization or area of responsibility.

The problem of assigning priorities to the solution of problems is a managerial meta-problem. How does the manager decide how to assign priorities to problems requiring his attention? Partial problem solutions produced by the system could help the manager in defining for himself a process for assigning priorities to a set of simultaneously presented problems.

To help the manager find problems for which management decisions are needed, a retrieval system needs to allow the manager to explore the information available and help direct him to relevant data.

IV THE INFERENTIAL SYSTEM

The tasks of the inferential element in a retrieval system are fundamentally to interpret the user's intent properly and to properly identify implicit information in the data store that may satisfy the user's intent.

We can partition the inferential system into three distinct functional elements: the preprocessor, the main inferential modules, and the translator.

The tasks of the preprocessor are:

(1) To predict--i.e., to set up expectations as to what to expect next from a particular user and at the same time to select the proper context within which such expected input will be processed.

(2) To interpret--i.e., to discover the user's intent by properly parsing the input query and identifying its semantic content.

(3) To expand--i.e., to paraphrase the identified intent both as a possible check with the user as to the fidelity of the interpretation, and to compare with prior knowledge about the query.

The tasks of the inferential module are:

(1) To classify--i.e., to identify relevant materials in store which relate to the present query.

(2) To categorize--i.e., to name and state new information, derived information, and newly generated processes.

(3) To generalize--i.e., to consolidate and rename data and procedures and to discard internally generated data while identifying the generating procedures.

Last, the translator is the element within the inferential system that is responsible for the conversion of the demand for data by the inferential system into executable instructions for the retrieval and processing of data that satisfy the intent of the query on hand.

V THE INTERNAL SYSTEM MODEL

The semantic analysis, involved in the performance of all the above tasks, relies heavily on the availability of an internal model of the user's universe of discourse and model of the data base subject matter. There are examples of recent system developments that include such models, which display an inherent superiority to other semantic interpretation approaches.[3-6]

In the case of our running example of a manager within an organization, we can consider a representation of his decision models as described earlier to stand for the needed internal model. Such representation should be organized according to the following criteria:

(1) State of affairs--What are the definitions of the universe of discourse? For example:

- Deviation of up to 10% from last month's performance is to be considered normal.
- Deviation in trends of up to 5% from last month's performance is to be considered normal.
- Maintenance data and flight data have in common (intersection) readiness data.

(2) Goals--What are our expectations within this universe of discourse? For example:

- The maintenance manager is interested in maximizing the readiness hours per aircraft.
- The operations manager is interested in maximizing the flight time per aircraft.
- The personnel manager is interested in maximizing the training time per aircraft.

(3) Policy--What is our attitude toward elements of the universe of discourse? For example:

- All aircraft should be handled at the lowest practical maintenance level.
- All maintenance personnel should be trained at least 10 hours per month.

(4) Means--What are the tools to realize the stated goals? We can list here the managers resources such as equipment, personnel, money, etc.

VI THE QUERY LANGUAGE

We describe a query language based on functional representation that we feel might be adequate for a manager to express his intent. The language is constructed in two parts. One part is a conceptual grammar that is used to construct declarative statements. The second part is a set of qualifying terms. When a qualifying term replaces an element of a declarative statement, the change transforms the statement into a query about the replaced element.

A. The Conceptual Grammar

We have borrowed from Shank[7] his graphical representation of the structure of the language. Given beside each graphical symbol is a possible machine representation. We give the graphical representation here and in the examples because of the intuitive appeal and because we feel it substantiates our claim that such a functional representation could become sufficiently expressive and comfortable for even the casual user. On the other hand, we hope that the reader will also be convinced that the interpretation of queries expressed in this language does not get out of hand in terms of identifying

the semantic content. Furthermore, the compactness of the representation could reasonably well accommodate a statement generator to respond to the user of this language.

We consider a conceptual grammar consisting of five elements:

Semantic	Graph Representation	Machine Representation
(1) Actor acting	PP ⟶ A	PP → A
(2) Object of an action	PP ⟵ A	PP ← A
(3) Recipient and donor of action	PP ⟶ PP ↑ A	PP * A * PP
(4) Attribute of an action	A ↑ AA	AA(A)
(5) Attribute of a PP	PP ↑ PA	AA(PP)

where PP = A picture producing element, a thing

PA = Attribute of a thing or a PP element, PP aid

A = Action

AA = Attribute of action, action aid.

B. The Qualifying Terms

The following qualifying terms were arrived at by formulating a large set of possible queries by a manager within a universe of discourse as defined earlier and then extracting the generic interrogative elements:

(1) For direct translation to data retrieval procedures, we have

When

How many

Any

Instances

Instance.

(2) For interpretation, followed by translation, we have

Who

Which

What

Does

Is.

(3) For interpretation, followed by the application of the inferencing mechanism, we have

Consequence of

Antecedence of

Related to

Implication of

Expound on.

(4) For evaluation of specific context followed by interpretation and application of the inferencing mechanism, we have

Effective use of

Policy variant

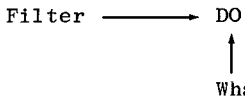
Problem area.

VII EXAMPLES

The following two examples are proposed:

(1) Consider the query: What is the job of a filter?

A conceptual representation of this query could take the form:



which is fashioned after the functional form

Notice that the queries

What does a filter do?

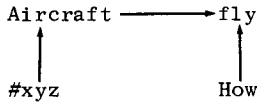
What is the filter's function?

How does a filter function?

can all be represented by the above conceptual form.

(2) Consider the query: How did aircraft number xyz fly?

This query fits into the conceptual representation form:



As with the first example, we expect the user to compose the conceptual representation of the query. We can conceive of the translation into an internal representation of the above query in the form of an incomplete tuple as follows:

Unit	Unit description	Action	Action Description
A/C	#xyz	fly	?

The Predict function could have anticipated queries by the present user. Then, let us assume that the user is a maintenance shop manager so that the system expects queries from him about flight data.

From the model of the data base within the inferential system, it is known that flight data are of the following relational form:

# of A/C	Type of A/C	Time in Flight Last Month	Landing	Purpose of Flight
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

The Interpretation function deduces that performance from flight-data for a shop manager means the last month's summary of the relation:

# of A/C	Purpose of Flight	Flight Time
•	•	•
•	•	•
•	•	•

so that for A/C xyz we might have:

Purpose of Flight	Flight Time
Training	40 hours
Transport	20 hours

as response to the query.

Further processing of the query could take place as follows:

(1) If the system assumes that the user operates within the Historical model, it might analyze the flight data for the month before last also, and add to the above response the statement:

"Total flight time; 40 percent less than the previous month."

(2) In a similar fashion it could have responded under the planning model

"Total flight time; 20 percent short of expected."

VIII SUMMARY

Our approach toward the design of a retrieval system relies heavily on the creation of an adequate internal representation of the user's universe of discourse.

This in our opinion, more than any other available technique, allows for effective restriction of the query language so that full natural-language understanding is unnecessary. On the other hand, sufficient expressive power could be given to the query language, which in turn could be properly interpreted with the aid of the internal model.

REFERENCES

1. W. Pounds, "The Process of Problem Finding," Sloan Management Review, pp. 1-20 (Fall 1969).

2. R. P. Abelson, "The Structure of Belief Systems," in Computer Simulation of Thought and Language (K. Colby and R. Shank, Eds.) to appear Spring 1974.

3. P. Thompson et al., "REL: A Rapidly Expanding Language System." Proc. 24th National ACM Conference (August 1969).

4. T. Winograd, "Procedures as Representation for Data in Computer Programs for Understanding Natural Language," MAC TR-84 Thesis Massachusetts Institute of Technology (1971).

5. K. M. Colby, et al., "Artificial Paranoia," Stanford University Memo. AIM-125 (July 1970).

6. D. A. Norman, "Memory, Knowledge, and the Answering of Questions," Center for Human Information Processing, Chip 25, University of California, San Diego, California (May 1972).

7. R. Shank, et al., "MARGIE: Memory Analysis, Response Generation and Inference in English," Stanford University Memo (February 1973).

QUESTIONS

Jack Minker: You have a relational system I take it, and you can have n-ary relations?

Waksman: Yes.

Jack Minker:

How do you embed the knowledge into this relational system? Your main contention is that you need a knowledge-base system, what do you mean by a knowledge-base system?

Waksman:

By knowledge-base system, I mean like knowing for a given user what relations are relevant, or what union of relations will be relevant or what are the limits in which he is interested. We input this in QA -4 in the form of a list of assertions in the form of a rule. When a question is being asked and before the retrieval operation begins, the question is filtered through the knowledge-base system. I'm not sure if I have answered your question.

Jack Minker:

I'm not sure that you have. When you have a data base system, the user is interested in everything. Now when you put a knowledge-base on top of this, it supposedly imparts some of its understanding to the system so that it can help you in some way. What way does the knowledge-base system help you in understanding the meaning of things within the system?

Waksman:

With the knowledge-base system, if we know which type of the four managers is asking the question, the question is being answered according to his area of interest. For example, we have a query that says "How good does aircraft number XYZ fly?" If the manager is actually a maintenance manager, then "how" refers to the number of flights. For an operations manager, "how" to the number of flights of a particular kind. Thus, the search is directed toward different files.

Gerard Salton:

To use your example: "How did aircraft XYZ fly?" You show the graph that represents this query and you say that you expect the user to compose a conceptual representation of the query. Now, have you any experience at all with users being able, or not being able to compose such conceptional representation of a query? Note that not only must a user know which terms are used, but he must know which terms are used with each other. Also, he must know which direction the relation is made, e.g. the direction is from "how" to "fly" and not vice versa. I simply cannot conceive of any user, no matter how advanced, being able to handle such as that in a reasonable way.

Waksman:

Well, it might be naivete but I am listing a grammar of five operators to which the user could refer in composing his query.

John T. Dockery:

I pose a problem and then maybe you can show me how to phrase it. I want to know the tail numbers, which distinguish jet helicopters for all those that have exceeded the mean life of the rotor blades.

This is a standard request; how do I phrase it for entry in your system. How would it compare for example with the toy department query in the previous discussion? [Referring to the paper by Boyce, et. al.]

Waksman:

If you are that specific, it doesn't vary. It's like an incomplete-tuple and you seek to see if you have such a relation in the file. If you do not, you have to use some intelligence maybe to combine some files.

John Dockery:

Where does what you're describing converge on what was described in the paper by Boyce, et. al.?

Waksman:

I will reconstruct it for you later if you like.

Ben Mittman:

I guess I am asking the same question, but maybe in a different way. Is the objective of your work to remove ambiguity from queries in those cases where the ambiguity might result from the difference in position of an individual? Is it an ambiguity resolving mechanism?

Waksman:

This is one objective, and the other objective is to be able to handle a more free form language, i.e. a language one level above an incomplete-tuple type.

DISCUSSION SESSION

Waksman:

Let me say that the paper was written before the implementation had actually begun. Since writing the paper we have discovered some things that need to be done differently. The intent was to imbed a retrieval system within a system having some knowledge. The major amount of intelligence resides in models of the user; so that the system response depends on the knowledge of the particular user's informational needs. A question asked in the meeting, which I did not answer very well, involved whether the system could always distinguish between the type of user. The answer is "no". But once it distinguishes the user, it attempts to understand the query within the context of the model of the user.

Unidentified Questioner:

Your paper contains some things that remind me of a recent paper, which I have not read, that deals with certain functions like why? when? where? what?, etc. Can you enlighten me as to whether there is a relationship between this work and your own?

Waksman:

I pick these up as the only qualifying terms that allow me to convert a statement into a question. This grammar allows me to construct a declarative statement. I'm allowing a qualifying term to be inserted at any point within the graph in order to make a query. I have selected these because of the context, and they are sufficient to flex the data base in order to construct meaningful responses.

Unidentified Questioner:

Proceeding to the storage question, which method of storage do you believe most appropriate? Would it be the relational kind or more similar to an attribute/value arrangement?

Waksman:

I consider it a data structure into which all other data structures could be mapped. Generality was the only property considered.

Unidentified Questioner:

Could your system actually be used by managers? Perhaps that question should be is there any system that should/could be used by managers? By that I mean users who are really interested in policies rather than day-to-day decisions. Another way to characterize my use of "managers" is those interested in the behavior of data rather than the data itself?

Waksman:

A system like this as it evolves should be able to handle manager-type questions. There is no reason to believe that it could not.