SOLUTIONS TO ISSUES DEPEND ON THE KNOWLEDGE REPRESENTATION

Frederick B. Thompson
California Institute of Technology
Pasadena, California

In organizing this panel, our Chairman, Bob Moore, expressed the view that too often discussion of natural language access to data bases has focused on what particular systems can or cannot do, rather than on underlying issues. He then admirably proceeded to organize the panel around issues rather than systems. In responding, I attempted to frame my remarks on each of his five issues in a general way that would not reflect my own parochial experience and interest. At one point I thought that I had succeeded quite well. However, after taking a clearer eyed view, it was apparent that my remarks reflected assumptions about knowledge representation that were by no means universal. This suggests a sixth issue which I would like to nominate:

> Are there really useful generalizations about computational linguistic issues that are independent of assumptions concerning knowledge representation?

I will come back to this sixth issue after discussing the five chosen by our Chairman.

Issue #1: Aggregate Functions and Quantity Questions

First, let us cast this issue in a somewhat different way. In many data base situations, there are classs of individuals all of whose members share the same attributes and thus, from the point of view of the data base, are indistinguishable. Thus there is no need to add all of these individuals as separate entities. To use Bob Moore's example, if a DEPARTMENT file has a field for NUMBER-OF-EMPLOYEES, it stands to reason that the particular individuals who actually existed in the various departments would not be separately represented in the database (for otherwise there would be a redundancy whose consistency would be hard to police). In such situations we need the notion of a "collective," namely a single data base object that takes the place of a number of individuals and which can carry their common attributes together with one additional item of information, namely their number. Thus a DEPARTMENT could have as a single member such a collective of employees, indeed it could have several such collective members and other individual members as well. The procedure that is called when answering "how many" and "number of" questions would know the difference between subclasses, individual members and collective members; it would know to recurse on subclasses, add one to its count for individual members and add the indicated number to its count for collective members. This appears to be a unified framework that will handle all of the cases mentioned in Bob Moore's statement of Issue #1.

Issue #2: Time and Tense

I should like to split this issue into two. The first sub-issue is the problem of handling continously varying phenomena, such as the movement of ships, the changing of relative amounts of ingredients in chemical reactions, or the percent completions of tasks. Here it is apparent that each instance will require a specialized procedure to handle interpolation. Ships cannot sail across land, thus an interpolation procedure that produces the position of a ship on the basis of its points of departure and destination will need to know about the coastlines of continents; movements to chemical equilibriums are not linear; task completions depend on changing personnel assignments. Just as we computational linguists provide to our system user the capability to introduce into his data base system such notions as locations of ports and ships, etc., we must also provide the means by which he can define such continuously varying parameters as position in such ways that appropriate interpolations can be made by the general system in conjunction with the particular definition. For example, the user may define: "position of X" in terms of calculations, perhaps extensive, involving the actual geometry of the seas.

The second sub-issue on which I would like to comment concerns those cases where discrete time intervals provide an adequate representation of the time aspects relevant to the data base. In these cases, if the time information is complete, i.e., actual starting and ending times of all events are recorded in the data base, the handling of time is rather straightforward. However this case often does not apply. Consider the following example:

> "The Kittyhawk arrived in London Monday. The Maru will sail from London Friday. Will the Kittyhawk and Maru have been in London at the same time?"

One is tempted to allow the computer to give a response: "Possibly," however the introduction of a three valued logic is fraught with well known dangers of its own. A more protracted response gets in the way of clause imbedding; how does one handle:

> "Will ships that have been in London together sail together?"

One answer would be:

> "The Kittyhawk arrived last Monday; the Maru will sail next Friday. If they will have been there at the same time, then not all ships that were in London together will sail together, but they would be the only exceptions."

Choosing a relevant diagnostic message, as above, is a major and difficult computational liguistic issue going well beyound questions concerning time and tense.

Issue #3: Quantifying into Questions

This is a deep, philosophical question. Computational linguists have progressed beyond the consideration of single sentences, and are seeking to follow the focus of a dialogue and identify the theme of a discourse. This is eventually an infinite regress, ultimately involving cross cultural backgrounds, the (perhaps Machiavellian) intent of those who control the use of a particular application, etc. But the engineering problem, at least at the present state of the art, is simple: what response is most useful to the user? Consider two possible answers to the following question:

> "Who manages each department?"

A1: "No single person manages all of the departments."

A2: "

department	manager
dept. A	manager A
...	...

Unless there were an undue number of departments involved, the second is clearly preferred, for it suffices even if the first were intended. In our own experience, "each" can usefully be interpreted as calling for a labeled list as answer in almost all cases. The difficulties of being more clever are great and will often result in a combinatorial explosion. I am sure, for a long time into the future, we will be seeking simple solutions that (a) are responsive in most cases, (b) provide the needed information, even though redundant in some case, and (c) make clear the misinterpretation in the few case where this arises, even though these solutions may violate strict linguistic analysis.

Issue #4: Querying semantically Complex Fields

In presenting this issue to the panel, Bob Moore used the following three questions as an example:

> "Is John Jones a child of an MIT alumnus?"
> "Is one of John Jones's parents an MIT alumnus?"
> "Did either parent of John Jones attend MIT?"

The apparent problem is the possibility of multiple descriptions, often involving disparate words, for getting at data in the data base. In designing our systems, we recognize two truths which appear to conflict: (a) the value of minimizing the redundancy of information in the data base, (b) the necessity of non-independent words in the vocabulary. In our own work, as most of you know, we have stressed the use of definitions as a means of achieving a synthesis of these two principles. I recommend it to you as a very useful tool in handling problems like Bob presents. We illustrate how Bob's example can be handled:

> "definition:child:converse of parent verb:John "attend"s MIT:John is a student of MIT
> definition:alumnus:person who had been a student"

The above three questions then are analyzed as:

"John Jones is (converse of parent) of a person who had been a student of MIT?"
"One of John Jones's parents is a person who had been a student of MIT?"
"Was either parent of John Jones a student of MIT?"

I do not wish to slur over the fact that a definition mechanism must be highly sophisticated in its handling of free variables, but our experience indicates that this can be done quite satisfactorily.

Issue #5: Multi-File Queries

This issue has been stated by Bob in terms of a traditional multiple file data base structure. This issue has its counterpart in semantic net data base structures discussed in papers on knowledge representation. Since we use such a semantic net structure for our data, let me rephrase the issue in those terms. In Bob's statement of the issue, he uses the example of the SHIP file and the PORT file, where the SHIP file has fields for home port, departure port and destination port. Paralleling his example, let us consider the phrase: "London ship". Suppose that (a) there was a ship named London, and (b) London was a home port, port of departure and destination, not necessarily of the same ship. Then "London ship" is four ways ambiguous, meaning: (1) the ship London, (2) London (home port) ships, (3) London (departure port) ships and (4) London (destination port) ships. In this formulation of the problem, all is easy; insofar as the phrase "London ship" is not disambiguated in context, the user is informed of the ambiguous meanings and the associated responses. The difficulty arises when there are possibile interpretations farther afield. Fort Collins is neither a port nor a ship, however the headquarters of the ABC Shipping Company is there and they own several ships. What are we to mean by "Fort Collins ship"? These are problems that were first attacked by Quillian, and I am not sure that anyone has added to his seminal analysis of them. In our own work, we have stopped at "once removed" connections, as illustrated by the four-way ambiguity above.

Issue #6: Solutions to Issues Depend on Knowledge Representation

As I look back on the above remarks concerning Bob's five issues, it becomes apparent that the usefulness of these remarks depends on the degree one is aware of the knowledge representation that underlies the solution suggested. For example, in the case of the last issue, if one only knew about traditional file structures, finding paths that link fields in more than one file appears all but unsolvable. Even if one is accustomed to semantic net structures, the viability of finding connective paths is highly dependent on the existence of back links between attributes and their arguments and values. Adding a definitional capability, other than simple abbreviations and synonyms, turns on the way free variables are handled in general and on the apparatus for binding them; for example, in processing the definition:

"definition:area:length times width"

when applied to a class, say "areas of ships", how does one ensure that he will obtain:

"length(i) * width(i)
for i = 1 to number of ships"

rather than:

"length(i) * width(j)
for i,j = 1 to number of ships?"

It comes down to how variables are maintained in the underlying knowledge representation.

One is forced to conclude that the basis for the integration of the syntax and semantics of computational linguistic systems is accomplished when the decisions on knowledge representation are made. Discussions of our various solution to the issues of computational linguistics can meaningfully take place only in terms of these underlying knowledge representations.