19

Understanding Queries and Signals

19.1 The Setting

Up until about the mid-1970s, DARPA managers were able to cushion the impact of the Mansfield Amendment (which required that Defense Department research be relevant to military needs) by describing computer research programs in a way that emphasized applications. Larry Roberts, the Director of DARPA's IPTO during the late 1960s and early 1970s, wrote[1]

> The Mansfield Amendment created a particular problem during my stay at DARPA. It forced us to generate considerable paperwork and to have to defend things on a different basis. It made us have more development work compared to the research work in order to get a mix such that we could defend it. I don't think I had to drop a project in our group due to the Mansfield Amendment, however. We could always find a way to defend computer science . . .
>
> The formal submissions to Congress for AI were written so that the possible impact was emphasized, not the theoretical considerations.

Cordell Green, working under Roberts at IPTO, wrote[2]

> Generally speaking, anything that came along in the AI field that we thought looked good was supported . . .
>
> One of my jobs was to defend the AI budget but that wasn't terribly difficult . . . all sorts of computer science is relevant because it will have a high impact on any large information-processing organization, and the Defense Department is certainly such an organization . . . all of this research should be kept alive because it had potential military relevance.

By the mid-1970s, however, the pressure to produce militarily useful systems became much more intense. DARPA, which had been generously supporting rather undirected basic AI research, started to focus instead on solving "pressing DoD problems." Although the director of DARPA's IPTO at the time, J. C. R. Licklider, was as sympathetic as ever to basic research in AI, DARPA's top management had entirely different attitudes. Licklider was having difficulties explaining his AI program to DARPA's "front office." The DARPA Director during the early 1970s, Stephen Lukasik, was (according to Licklider[3])

> neither for nor against AI. He was for good management and he got the idea that maybe some of the AI stuff wasn't being very well managed. . . . [He] had a fixed idea that a proposal is not a proposal unless it's got milestones. I think that he believed that the more milestones, the better the proposal. . . . I think he was not developing a distaste for AI but a conviction that this is such an important field that the researchers have got to learn to live in a bigger, more rigid, more structured bureaucracy."

https://doi.org/10.1017/CBO9780511819346.023 Published online by Cambridge University Press

Figure 19.1. George Heilmeier. (Photograph courtesy of DARPA.)

Lukasik's view about how projects should be managed had a direct effect on DARPA-supported basic research in AI. For example, a "Quarterly Management Report" that I submitted in February 1975 describing progress on the SRI computer-based consultant caused Licklider to ask how the report might be recast to emphasize progress along certain paths in a "PERT Chart." "What I would like to have," he wrote me in a letter dated March 3, 1975, "is the PERT Chart – so that I can mark the accomplishments in red and see where you stand with respect to the overall pattern. . . . Do you have such a chart? If so, please send me a copy. If not, how about making one? It would really help us greatly here at ARPA."[4]

Of course, in basic research, although one can describe generally the problems one is trying to solve, one can't describe (ahead of time) what the solutions are going to be. In fact, as exploratory research progresses, new problems become apparent, so one can't even describe all the problems ahead of time. One can't make the kind of detailed plan for basic research that one can make for applying already developed technology to specific applications. Unfortunately, the management of DARPA was shifting from people who understood how to initiate and manage basic research to people who knew how to manage technology applications.

The shift toward shorter term, intensely managed research became more pronounced when George Heilmeier (Fig. 19.1) replaced Stephen Lukasik as DARPA Director in 1975. Heilmeier came from RCA, where he had headed the research group that invented the first liquid crystal display. Licklider later wrote that Heilmeier "wanted to understand AI in the way he understood liquid crystal displays . . ."[5]

One of the tasks that Heilmeier gave IPTO was to produce a "roadmap" (that is, a detailed plan) for its AI research program (and its other computer science programs too). This roadmap should summarize past accomplishments, indicate areas where existing technology could be applied to military problems, and show milestones along the way. This "guidance" from DARPA management caused great difficulties for Licklider, some of which were explained in an e-mail he sent to some leaders of

https://doi.org/10.1017/CBO9780511819346.023 Published online by Cambridge University Press

AI research in April of 1975. (I was among the recipients of his "Easter Message," e-mailed on April 2, 1975.) Here are some excerpts:

> The purpose of this Easter note is to bring you up to date on a development in ARPA that concerns me greatly – and will, I think, also concern you. . . .
>
> . . . the prevailing direction in ARPA is to do research within the specific contexts of military problems and not to do research that does not have a military 'buyer' ready to take it over as soon as the concept gets well formulated. . . .
>
> [there are] strong pressures from the new Director, George Heilmeier, that IPTO 'redirect' the university AI efforts to work on problems that have real DoD validity . . .
>
> . . . the situation is complicated by the fact that ARPA has been supporting basic research at a rather high level for more than ten years (has spent more than $50 million on it), and it is natural for a new director, or even an old one, to ask, "What have we gotten out of it in terms of improvements in national defense?"

According to Licklider's Easter note, some of the things that Heilmeier thought IPTO could do for the Defense Department were the following:

- get computers to read Morse code in the presence of other code and noise,
- get computers to identify/detect key words in a stream of speech,
- solve DoD's "software problem,"
- make a real contribution to command and control, and
- do a good thing in sonar.

Even though one of the items on Heilmeier's list involved speech processing, one of the casualties of his tenure as Director of DARPA was the SUR Program. None of the systems that had been developed under the program could respond in real time, nor could they deal with large enough vocabularies. Heilmeier believed (probably with good reason) that speech understanding was still a basic research activity. Thus, he thought, it should be supported, say, by the National Science Foundation (NSF), and he rejected proposals for DARPA to continue it.

Unfortunately, most of the research areas that were on Licklider's own list (which was also mentioned in his Easter note) were not explicitly on Heilmeier's. (I can't resist mentioning one of the items on Licklider's list: "Develop a system that will guide not-sufficiently-trained maintenance men through the maintenance of complex equipment.") One of Heilmeier's items was sufficiently vague, however, to justify work both in NLP and in computer vision. That was "command and control," an activity that involves getting and presenting relevant information to commanders so that they can control military forces effectively.

DARPA program officers Floyd Hollister and Col. David Russell were able to persuade DARPA management that text-based, natural language access to large, distributed databases would be an important component of command and control systems. They argued that the technology for such access was sufficiently far along for it to be applied in what they called "command-and-control test-bed systems." After all, Bill Woods and colleagues at BBN had already demonstrated LUNAR, a natural language "front end" to databases about moon rocks. Several other researchers had also begun work on the problem of how to communicate with computers using

https://doi.org/10.1017/CBO9780511819346.023 Published online by Cambridge University Press

Figure 19.2. Gary Hendrix. (Photograph courtesy of Gary Hendrix.)

English or some other natural language. (For example, there were over forty papers on NLP presented at the Fifth IJCAI in 1977 at MIT, and the February 1977 issue of the ACM's *SIGART Newsletter* published 52 summaries of ongoing research on "Natural Language Interfaces.") In the next part of this chapter, I'll describe some of the accomplishments during this period on communicating with computers using natural language.

A second area of great importance in command and control was automating the analysis of aerial photos. Spotting targets of military interest in these photos, such as roads, bridges, and military equipment, typically required hours of effort by intelligence analysts. Because techniques being developed by researchers in computer vision might provide tools to help human analysts, DARPA had good reasons to continue funding computer vision research. In 1976, it began the "Image Understanding" (IU) program to develop the technology required for automatic and semi-automatic interpretation and analysis of military photographs and related images. Although initially conceived as a five-year program, it continued (with broader objectives) for well over twenty years. I'll summarize the image understanding work, along with other computer vision research, in a subsequent chapter.

Doing something about sonar was one of the items on Heilmeier's list. In fact, in his Easter note Licklider wrote "One of [Heilmeier's] main silver-bullet areas is underwater sound and sonar, and IPTO is in the process of 'buying in' on the HASP project (Ed Feigenbaum's AI approach)." I'll describe HASP and how DARPA "bought in" to the project toward the end of the chapter.

19.2 Natural Language Access to Computer Systems

19.2.1 LIFER

At SRI, Gary Hendrix (Fig. 19.2) had been developing a system called LIFER (an acronym for Language Interface Facility with Elliptical and Recursive Features), programmed in INTERLISP, for rapid development of natural language "front ends" to databases and other software. LIFER allowed a nontechnical user to specify a subset of a natural language (for example, English) for interacting with a database system or other software. A parser contained within LIFER could then translate sentences

https://doi.org/10.1017/CBO9780511819346.023 Published online by Cambridge University Press

and requests in this language into appropriate interactions with the software. LIFER had mechanisms for handling elliptical (that is, incomplete) inputs, for correcting spelling errors, and for allowing novices to extend the language through the use of paraphrases.

An interesting feature of LIFER was that the language it could handle was defined in terms of "patterns," which used semantic concepts in the domain of application. One such pattern, for example, might be

```
WHAT IS THE <ATTRIBUTE> OF <PERSON>
```

where the words WHAT, IS, THE, and OF are actual words that might occur in an English query and <ATTRIBUTE> and <PERSON> are "wild cards" that could match any word in predefined sets. <ATTRIBUTE> might be defined to match words such as AGE, WEIGHT, HEIGHT, etc., and <PERSON> might match JOHN, SUSAN, TOM, etc. This pattern would then "recognize" a sentence such as

```
WHAT IS THE HEIGHT OF SUSAN
```

This method of defining a "grammar" is to be contrasted with the usual syntactic phrase-structure rules such as S <= NP VP. As I mentioned earlier, grammars based on concepts in the domain of application are called "semantic grammars."

LIFER used a simplified augmented transition network (like those I described in a previous chapter) to analyze an input sentence. Each pattern defined by the grammar corresponded to a possible "path" in the transition network. An input sentence was analyzed by attempting to match it with one of these paths, noting which specific instance of a wild card, such as <ATTRIBUTE>, was used in the match. Depending on the path taken and on the values of wild cards in the path, software was automatically created that was then used to make the appropriate database query or to carry out an appropriate command.[6] In 1982, Hendrix left SRI to form Symantec, a company that planned to develop and market a natural language question-answering system based on semantic grammars such as LIFER. [Perhaps natural language processing (or the intended market) was not quite ready, because Symantec was later reorganized to market computer security and anti-virus software.]

LIFER was used at SRI as the natural language component of a system called "LADDER" for accessing multiple, distributed databases.[7] LADDER (an acronym for Language Access to Distributed Data with Error Recovery) translated the English query into a hypothetical database query that assumed a very simple database organization. Using a system called IDA (an acronym for Intelligent Data Access), that hypothetical query was transformed into a series of actual database queries that took into account the actual organization of the database. It also took account of syntactic and semantic knowledge to attempt to produce very efficient queries and to detect any erroneous updates to the database content. (More research on systems similar to IDA was performed in a joint program between Stanford University and SRI, named KBMS, an acronym for Knowlege Based Management System, with support from DARPA.)

Consistent with DARPA's focus on military applications, LADDER was able to answer questions about naval ships using information about ship sizes, types, locations, and so on from various databases. Some sample interactions with an early

```
-What is the speed of the Kitty Hawk
PARSED!
((SPEED 35 KNOTS))

-Of the Ethan Allen
TRYING ELLIPSIS:  WHAT IS THE SPEED OF THE ETHAN ALLEN
((SPEED 30 KNOTS))

-Displacement
TRYING ELLIPSIS:  WHAT IS THE DISPLACEMENT OF THE ETHAN ALLEN
((STANDARD-DISPLACEMENT 6900 HUNDRED-TONS))

-length of the fastest Soviet sub
TRYING ELLIPSIS:  WHAT IS THE LENGTH OF THE FASTEST SOVIET SUB
((LENGTH 285 FEET / SPEED 30 KNOTS))

-Who onws the KIEV
     OWNS <==(assumed spelling error)
PARSED!
((COUNTRY USSR))

-who owns the JFK
TRYING ELLIPSIS:  ELLIPSIS HAS FAILED
THE PARSER DOES NOT EXPECT THE WORD "JFK" TO FOLLOW "WHO OWNS THE"
OPTIONS FOR NEXT WORD OR META-SYMBOL ARE:
<SHIP-NAME>

-Define JFK to be like Kennedy
PARSED!
 . {JFK is now a synonym for KENNEDY, which is a ship name}
 .
-REDO -2           {that is, parse WHO OWNS THE JFK}
PARSED!
((COUNTRY USA))

-? BUILT LAFAYETTE
TRYING ELLIPSIS:  ELLIPSIS HAS FAILED
 . {error message omitted}
 .
-Let "? built Lafayette" be a paraphrase of "who built the Lafayette"
PARSED!
 .
 .
-? built Lafayette
PARSED!
((BUILDER GENERAL.DYNAMICS))

-owns longest nuclear submarine
TRYING ELLIPSIS:  ? OWNS LONGEST NUCLEAR SUBMARINE
((COUNTRY USSR / LENGTH 426 FEET))
```

Figure 19.3. Sample interactions with LADDER. (Used with permission of SRI International.)

version of LADDER are shown in Fig. 19.3. Note the ability of the system to correct spelling errors, to deal with incomplete questions, and to accept paraphrases.[8]

19.2.2 *CHAT-80*

Between 1979 and 1982, Fernando Pereira (1952–) and David H. D. Warren (circa 1950–) developed a system called CHAT-80 at the University of Edinburgh as part of Pereira's Ph.D. dissertation there. CHAT-80 was able to answer rather complex questions, posed in English, about a database of geographical facts.

According to Pereira's dissertation,[9] work on CHAT-80 started as "an attempt to clarify and improve some previous NL work of Colmerauer." CHAT-80 was written in PROLOG, the logic-based programming language developed originally by Alain Colmerauer. In fact, the grammar used by CHAT-80 consisted of logical formulas stated in the PROLOG language. For example,

```
sentence(s(NP,VP), S0,S):-noun_phrase(NP, N, S0,S1),
verb_phrase(VP, N, S1,S)
```

is CHAT-80's way of stating that "there is a sentence between points S0 and S in a string (of words) if there is a noun phrase with number N (that is, singular or plural) between points S0 and S1, and a verb phrase with number N between points S1 and S." Grammars defined by PROLOG clauses of this kind are called *Definite Clause Grammars* (DCGs). Several clauses of this sort were used by CHAT-80 to parse English sentences. The actual parsing was done by the PROLOG program consisting of these clauses.

In CHAT-80, computation of the meaning (that is, the semantics) of an English query was guided by the syntactic structure of the query (as computed by the PROLOG program) and was expressed as a logical formula. This formula was then transformed into the individual queries of the database needed to answer the original question.[10] (For information about how to get a running version of CHAT-80, see http://www.cis.upenn.edu/˜pereira/oldies.html.)

Here are a few examples (from Chapter 5 of Pereira's dissertation) of queries that CHAT-80 was able to answer:

```
Q: What is the capital of Upper Volta?
A: Ouagadougou
Q: Which country's capital is London?
A: united kingdom
Q: What is the ocean that borders African countries and that
  borders Asian countries?
A: indian ocean
Q: What are the capitals of the countries bordering the
  Baltic?
A: denmark:copenhagen; east germany:east berlin;
  finland:helsinki; poland:warsaw; soviet union:moscow;
  sweden:stockholm; west germany:bonn
Q: What is the total area of countries south of the Equator
  and not in Australasia?
A: 10,228 ksqmiles
Q: What are the continents no country in which contains more
  than two cities whose population exceeds 1 million?
A: africa, antarctica, australasia
Q: Which country bordering the Mediterranean borders a
  country that is bordered by a country whose population
  exceeds the population of India?
A: turkey
```

WORLDC

NAME	CONTINENT	CAPITAL	AREA	POP
Afghanistan	Asia	Kabul	260,000	17,450,000
Albania	Europe	Tirana	11,100	2,620,000
Algeria	Africa	Algiers	919,951	18,510,000

BCITY

NAME	COUNTRY	POP
Brussels	Belgium	1,050,787
Buenos Aires	Argentina	8,925,000
Canberra	Australia	210,600

CONT

NAME	HEMI	AREA	POP
Africa	S	11,500,000	41,200,000
Antarctica	S	5,000,000	500
Asia	N	16,990,000	2,366,000,000

PEAK

NAME	COUNTRY	HEIGHT	VOL
Anocagua	Argentina	23,080	N
Annapurna	Nepal	26,504	N
Chimborazo	Ecuador	20,702	Y

Figure 19.4. Files used in a TEAM database. (Used with permission of SRI International.)

Although these examples indicate rather impressive performance, CHAT-80s abilities were constrained by its limited vocabulary and grammar. These limitations are described in detail in Pereira's dissertation.

19.2.3 *Transportable Natural Language Query Systems*

As I have described it, CHAT-80 was implemented as a system for querying a database of geographical facts. However, since much of its design was not specific to geography, it could rather easily be modified to be able to deal with other databases. CHAT-80 was just one of several query systems that were "transportable" in the sense that they could be adapted to serve as natural language front ends to a variety of different databases. Other such systems were ASK developed at Caltech,[11] EUFID developed at SDC,[12] IRUS developed at BBN,[13] LDC-1 developed at Duke University,[14] NLP-DBAP developed at Bell Laboratories,[15] and TEAM developed at SRI.[16]

Since I know more about TEAM than I do about the others, I'll say a few things about it as representative of its class. TEAM (an acronym for Transportable English Database Access Medium) was supported by DARPA and was designed to acquire information about a database from a database administrator and to interpret and answer questions of the database that are posed in a subset of English appropriate for that database. TEAM, like many other transportable systems, was built so that the information needed to adapt it to a new database and its corresponding subject matter could be acquired from an expert on that database even though he or she might know nothing about natural language interfaces.

To illustrate the operation of TEAM, its designers used a database consisting of four "files" (or "relations") of geographic data. Partial versions of these files are shown in Fig. 19.4. I'll trace through some of the steps TEAM used to answer the query "Show each continent's highest peak."

TEAM used a subsystem called DIALOGIC[17] to convert the English query into a logical expression. Within DIALOGIC, a subsystem based on DIAMOND[18] performed syntactic analysis using the DIAGRAM grammar.[19] The highest scoring parse tree is shown in Fig. 19.5.

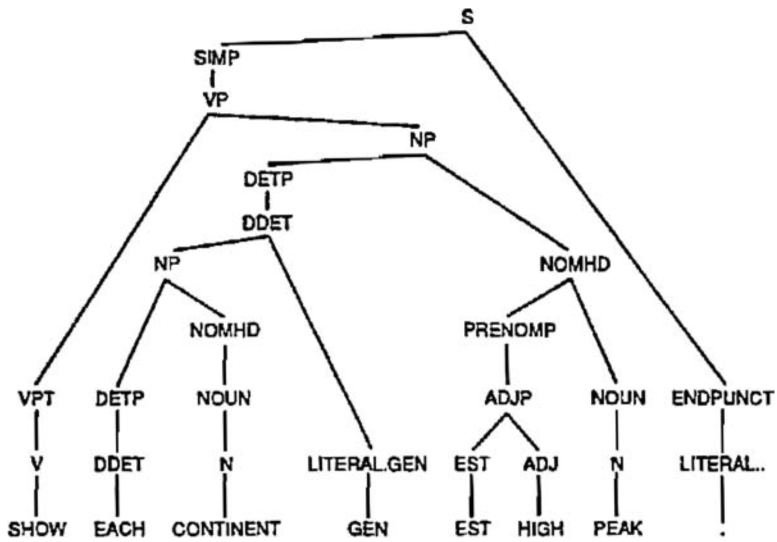

Figure 19.5. A parse tree for "Show each continent's highest peak." (Used with permission of SRI International.)

Based on this parse tree and knowledge about the concepts used in the database, a semantic analysis system converted the query into the following logical expression (here restated in an English-like form for better understandability):

```
FOR EVERY CONTINENT
    WHAT IS EACH PEAK
    SUCH THAT THE PEAK IS THE HIGHEST PEAK SUCH THAT
        THE CONTINENT IS CONTINENT OF THE PEAK?
```

TEAM then used its knowledge about the structure of the database and about how components of this logical expression are associated with relations in the database to generate the actual database query and construct an answer.

19.3 HASP/SIAP

In 1972, while Larry Roberts was still the Director of IPTO, he asked Ed Feigenbaum at Stanford to think about applying the AI ideas so successfully used in DENDRAL to the problem of identifying and tracking ships and submarines in the ocean using acoustic data from concealed hydrophone arrays.

Some of the acoustic data picked up by the hydrophone arrays come from rotating shafts and propellors and reciprocating machinery on board ships. Different ships emit sounds with their own characteristic identifying fundamental frequencies and harmonics. Human specialists who analyze this sort of surveillance data look at the sonogram displays of ocean sounds and, by matching sound spectra to stored references, attempt to identify and locate ships that might be present (if any). Making these decisions often requires using information not present in the signals themselves, information such as reports from other sensor arrays, intelligence reports, and general knowledge about the characteristics of ships and common sea lanes.

The analysis problem is complicated by several factors:[20]

> The background noise from distant ships is mixed with storm-induced and biological noises. Sound paths to the arrays vary with diurnal and seasonal cycles. Arrival of sound energy over several paths may suddenly shift to no arrivals at all, or arrivals only of portions of vessel radiation. Sound from one source can appear to arrive from many directions at once. Characteristics of the receivers can also cause sound from different bearings to mix, appearing to come from a single location. Finally, the submarine targets of most interest are very quiet and secretive.

Supported by DARPA, work on this problem began in 1973 at Systems Control Technology, Inc. (SCI), a Palo Alto company with expertise in this area that could work on classified military projects. (SCI was later acquired by British Petroleum.) Feigenbaum, and his colleagues at SCI, soon realized that the "generate-and-test" strategy of DENDRAL would not work for the problem of ocean surveillance because there was no "legal move generator" that could produce candidate ship positions and their tracks given the surveillance data. However, noting that the overall analysis problem could be divided into levels similar to those used in the Blackboard architecture of HEARSAY-II (a system shown to be good at dealing with signals in noise), the team thought that something similar would work for their problem. The team developed a system called HASP (an acronym for Heuristic Adaptive Surveillance Program) based on the Blackboard model. Follow-on work that would process actual ocean data began at SCI with SIAP (an acronym for Surveillance Integration Automation Program) in 1976. I'll give a brief description of the HASP/SIAP system design and then summarize how it performed.

The top level of the Blackboard was a "situation board" – a symbolic model of the unfolding ocean situation, built and maintained by the program. It described all the ships hypothesized to be out there with a confidence level associated with each of them.

Just below the situation board level was a level containing the individual hypothesized vessels. Each vessel element had information about its class, location, current speed, course, and destination, each with a confidence weighting. Below the vessel level was a level containing hypothesized sound sources: engines, shafts, propellers and so on with their locations and confidence weightings. Spectral features abstracted from the acoustic data were at the lowest level.

The levels were linked by knowledge sources (KSs) that were capable of inferring that if certain elements were suspected to be present at one level then other elements could be inferred to be present at another level (or if they were already present at that level, their confidence could be adjusted). Just as in HEARSAY-II, the links could span multiple levels and make inferences upward, downward, or within a level. An inference caused by one KS might allow another KS to draw an additional inference, and so on in cascade, until all relevant information had been used. In this manner, new information could be assimilated and expectations concerning possible future events could be formulated.

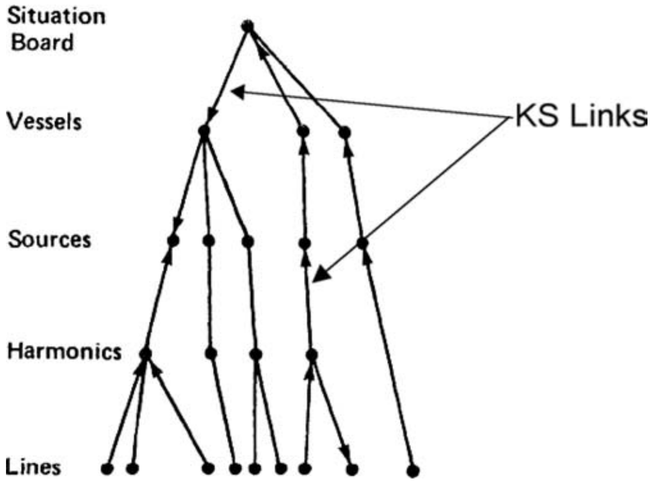

Figure 19.6. A network structure linking data at different levels. (Illustration from H. Penny Nii, Edward A. Feigenbaum, John J. Anton, and A. J. Rockmore, "Signal-to-Symbol Transformation: HASP/SIAP Case Study," *AI Magazine*, Vol. 3, No. 2, p. 26, Figure 2, © 1982, Association for the Advancement of Artificial Intelligence. Used with permission.)

One type of KS was composed of IF–THEN rules. (Other types were used also.) For example, here is an IF–THEN rule (translated into English for readability) that acted within the source level:

IF: a source was lost due to fade-out in the near-past, and a similar source started up in another frequency, and the locations of the two sources are relatively close,

THEN: they are the same source with confidence of 3.

HASP/SIAP had several kinds of knowledge sources, each represented in a way appropriate to the level(s) involved. Some KSs were based on information about the environment, such as common shipping lanes, location of arrays, and known maneuver areas. Others had information about vessels and vessel types, their speeds, component parts, acoustic characteristics, home bases, and so on. In addition to KSs dealing with knowledge appropriate to the various levels, there were "meta" KSs that had information about how to use other KSs.

The actions of the KSs in linking information at the various levels can be represented as a network, such as the one shown schematically in Fig. 19.6. At the end of an analysis session, when all KSs have had a chance to participate and the action dies down, the resulting network is called the "current best hypothesis" (CBH) about the current ocean situation. Here is a partial sample (translated into English) of how a CBH for a particular run of HASP/SIAP might be described:[21]

The class of Vessel-l, located in the vicinity of Latitude 37.3 and Longitude 123.1 at time day 2, 4 hours, 55 minutes, can be either Cherry, Iris, Tulip, or Poppy class. Two distinct acoustic sources, supported by respective harmonic sets, have been identified for Vessel-l. Source-l could be due to a shaft or propeller of vessel class Cherry or Poppy. Similar source

https://doi.org/10.1017/CBO9780511819346.023 Published online by Cambridge University Press

possibilities exist for Source-5. These two sources were assimilated into Vessel-l because of the possibility of a known mechanical ratio that exists between the two sources.

The MITRE Corporation conducted several experiments to compare the performance of HASP/SIAP against that of two expert sonar analysts. In one of these experiments, MITRE concluded that "HASP/SIAP has been shown to perform well on ocean derived data . . . For this restricted ocean scene, the program is not confused by extraneous data and gives results comparable to an expert analyst." In another experiment, it concluded that "HASP/SIAP understood the ocean scene more thoroughly than the second analyst and as well as the first analyst. . . . The program can work effectively with more than one acoustic array. SIAP classified an ocean scene over a three hour time period indicating the plausibility of SIAP efficacy in an evolving ocean situation." The third experiment led to the conclusions that "with the exception that the SIAP program obtained significantly more contacts than the human analysts, the descriptions of the ocean scene are very similar." Moreover, "SIAP can perform vessel classification in increasingly difficult ocean scenes without large increases in the use of computer resources."[22]

As mentioned earlier, the Blackboard model has been applied in several other areas as well. Examples include protein crystallographic analysis,[23] image understanding,[24] and dialog comprehension.[25] Interestingly, the Blackboard architecture has impacts beyond technology. Donald Norman, a cognitive psychologist, has said that HEARSAY-II has been a source of ideas for theoretical psychology and that it fulfills his "intuitions about the form of a general cognitive processing structure."[26] Also, as I'll mention in a later chapter, several models of the neocortex involve interacting layers resembling both the form and the mechanisms of Blackboard systems.

Notes

1. Lawrence G. Roberts, "Expanding AI Research and Founding Arpanet," in Thomas C. Bartee (ed.), *Expert Systems and Artificial Intelligence: Applications and Management*, pp. 229–230, Indianapolis, IN: Howard W. Sams & Co., 1988. **[244]**
2. C. Cordell Green, "AI During IPTO's Middle Years," *ibid*, pp. 238–240. **[244]**
3. J. C. R. Licklider, "The Early Years: Founding IPTO," *ibid*, pp. 225–226. **[244]**
4. Licklider letter in my file. **[245]**
5. J. C. R. Licklider, *op. cit.*, p. 226. **[245]**
6. For technical details about LIFER, see Gary G. Hendrix, "LIFER: A Natural Language Interface Facility," SRI AI Center Technical Note 135, December 1976 (available online at http://www.ai.sri.com/pubs/files/1414.pdf); Gary G. Hendrix, "The LIFER Manual: A Guide to Building Practical Natural Language Interfaces," SRI AI Center Technical Note 138, February 1977 (available online at http://www.ai.sri.com/pubs/files/749.pdf); and Gary G. Hendrix, "Human Engineering for Applied Natural Language Processing," *Proceedings of the 5th IJCAI*, pp. 183–191, 1977 (which also appeared as SRI AI Center Technical Note 139, available online at http://www.ai.sri.com/pubs/files/748.pdf). **[248]**
7. Earl D. Sacerdoti, "Language Access to Distributed Data with Error Recovery," *Proceedings of the 5th IJCAI*, pp. 196–202, 1977, and reprinted as SRI AI Center Technical Note 140, February 1977 (available online at http://www.ai.sri.com/pubs/files/747.pdf); Earl D. Sacerdoti, "A LADDER User's Guide (Revised)," SRI AI Center Technical Note

163R, March 1980 (available online at http://www.ai.sri.com/pubs/files/735.pdf); and Gary G. Hendrix *et al.* "Developing a Natural Language Interface to Complex Data," *ACM Transactions on Database Systems,* Vol. 3, No. 2, pp. 105-147, June 1978 (available online as SRI AI Center Technical Note 152, August 1977, at http://www.ai.sri.com/pubs/files/741.pdf). **[248]**

8. For a more extensive interaction with a later version of LADDER, see Appendix A of Earl D. Sacerdoti, "A LADDER User's Guide (Revised)," SRI AI Center Technical Note 163R, March 1980. **[249]**
9. Fernando Pereira, "Logic for Natural Language Analysis," Ph.D. dissertation, University of Edinburgh, 1982. A slightly revised version of the dissertation was published as Technical Note 275 of the SRI AI Center and is available online at http://www.ai.sri.com/pubs/files/669.pdf. **[250]**
10. Readers interested in the details of these rather technical processes might refer to Pereira's dissertation or to David H. D. Warren and Fernando Pereira, "An Efficient Easily Adaptable System for Interpreting Natural Language Queries," *Computational Linguistics*, Vol. 8 , Nos. 3–4, pp. 110–122, July–December 1982. **[250]**
11. Bozena H. Thompson and Frederick B. Thompson, "Introducing ASK, A Simple Knowledgeable System," *Conference on Applied Natural Language Processing*, pp. 17–24, 1983. Available online at http://ucrel.lancs.ac.uk/acl/A/A83/A83-1003.pdf. **[251]**
12. Marjorie Templeton and John Burger, "Problems in Natural Language Interface to DBMS with Examples from EUFID," *Proceedings of the First Conference on Applied Natural Language Processing*, pp. 3–16, 1983. Available online at http://www.aclweb.org/anthology-new/A/A83/A83-1002.pdf. **[251]**
13. Madeleine Bates and Robert J. Bobrow, "A Transportable Natural Language Interface," *Proceedings of the 6th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 81–86, 1983. **[251]**
14. Bruce Ballard, John C. Lusth, and Nancy L. Tinkham, "LDC-1: A Transportable, Knowledge-Based Natural Language Processor for Office Environments," *ACM Transactions on Information Systems*, Vol. 2, No. 1, pp. 1–25, January 1984. **[251]**
15. Jerrold M. Ginsparg, "A Robust Portable Natural Language Data Base Interface," *Conference on Applied Natural Language Processing*, pp. 25–30, 1983. Available online at http://ucrel.lancs.ac.uk/acl/A/A83/A83-1004.pdf. **[251]**
16. Barbara J. Grosz *et al.*, "TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces," *Artificial Intelligence*, Vol. 32, No. 2, pp. 173–243, May 1987. Available online as SRI Technical Note 356R, October 20, 1986, at http://www.ai.sri.com/pubs/files/601.pdf. **[251]**
17. Barbara Grosz *et al.*, "DIALOGIC: A Core Natural-Language Processing System," *Proceedings of Ninth International Conference on Computational Linguistics*, pp. 95–100, 1982. Available online at http://www.aclweb.org/anthology-new/C/C82/C82-1015.pdf. **[251]**
18. DIAMOND was developed at SRI by William Paxton and is described in Ann E. Robinson *et al.*, "Interpreting Natural-Language Utterances in Dialogs About Tasks," AI Center Technical Note 210, SRI International, March 1980. Available online at http://www.ai.sri.com/pubs/files/709.pdf. **[251]**
19. Jane J. Robinson, "DIAGRAM: A Grammar for Dialogs," *Communications of the ACM*, Vol. 25, No. 1, pp. 27–47, January 1982. Available online as SRI AI Center Technical Note 205, February 1980, at http://www.ai.sri.com/pubs/files/712.pdf. **[251]**
20. H. Penny Nii, Edward A. Feigenbaum, John J. Anton, and A. J. Rockmore, "Signal-to-Symbol Transformation: HASP/SIAP Case Study," *AI Magazine*, Vol. 3, No. 2, pp. 23–35, 1982. **[253]**

21. From *ibid*, p. 28. **[254]**
22. From *ibid*, p. 34. **[255]**
23. Robert S. Engelmore and H. Penny Nii, "A Knowledge-Based System for the Interpretation of Protein X-Ray Crystallographic Data," Stanford Computer Science Department Technical Report CS-TR-77-589, 1977; available online at ftp://reports.stanford.edu/pub/cstr/reports/cs/tr/77/589/CS-TR-77-589.pdf. **[255]**
24. A. R. Hanson and E. M. Riseman, "VISIONS: A Computer System for Interpreting Scenes," in A. Hanson and E. Riseman (eds.), *Computer Vision Systems*, pp. 303–333, New York: Academic Press, 1978. **[255]**
25. W. C. Mann, "Design for Dialogue Comprehension," in *Proceedings of the 17th Annual Meeting of the Association of Computational Linguistics*, pp. 83–84, La Jolla, CA, August 1979; available online at http://ucrel.lancs.ac.uk/acl/P/P79/P79-1020.pdf. **[255]**
26. Donald A. Norman, "Copycat Science or Does the Mind Really Work by Table Lookup?," in R. Cole (ed.), *Perception and Production of Fluent Speech*, Chapter 12, Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., 1980. **[255]**