AN ENGLISH-LIKE LANGUAGE FOR QUALITATIVE SCIENTIFIC KNOWLEDGE

Douglas Skuce
Dept. of Electrical Engineering and Montreal Neurological Institute
McGill University
3801 University St., Montreal, P.Q., Canada

Abstract

KAL (Knowledge Acquisition Language) is a language intended not for programming but for the expression of qualitative scientific knowledge. It is designed to accommodate both the semantic constructs most frequently found in scientific text (at present, excluding the description of change) and the knowledge storage and retrieval mechanisms of languages like CONNIVER, in which it may be implemented. Emphasis is placed on using "real" text, and in interacting with scientists (particularly biologists) to discover their particular needs. Though KAL is English-like, most natural languages could be substituted with no difficulty.

Introduction

It is generally agreed that mathematics is not well suited to describe most biological knowledge, though whether this is an (unfortunate) historical accident or a reflection of how our minds relate to the world is an open question. What appears to be more appropriate, however, are the formalisms which are becoming increasingly well developed through AI research into representing non-quantitative knowledge. For many reasons, however, some of which are justifiable, these languages (e.g., CONNIVER) tend to remain within AI laboratories. I wish to suggest that this tendency of "growing outward from the AI lab to someday intersect the real world", evidenced by the comparative lack of attempts to apply any developments "within AI" to problems not invented by AI researchers may, in fact, be detrimental to both AI and the rest of the world. With this in mind I have tried to place myself in an "interface" position, to try to bring together the types of knowledge that biologists need and the tools developed within the AI community for expressing qualitative knowledge.

I am therefore in the process of defining a language, called KAL (Knowledge Acquisition Language), which is not a programming language but a language intended to be used by biologists or physicians to express as much as possible of the qualitative knowledge they use in their everyday work. The development is at the point (December, 1974) of making initial attempts to expose some biologists to the language, by means of a number of hours of explanation plus a 50 page "user's manual", written for persons unfamiliar with computers. To parallel KAL's evolution through interaction with biologists, I have begun implementing some of it, using UCI LISP and CONNIVER, so that it can be developed both "on paper" and as a program, in the question answering system tradition. It is not anticipated that all of what can be defined on paper will be implemented.

Design Criteria for KAL

The following desiderata were foremost in the design of the language:

1) It must be suited primarily for the expression of that portion of biological (including medical) knowledge which is now expressed in natural language text. This excludes all forms of graphical and mathematical expressions, since the problem of representing graphical knowledge (pictures and diagrams) is a flourishing field in AI, and representing mathematical knowledge is the bread and butter of computers.

2) It must be comprehensible to persons having no experience with computers. Hence, I have chosen to make KAL look as much like a natural language (English) as possible, so that most KAL sentences can be immediately understood by anyone familiar with KAL who knows the technical vocabulary. Most motivated biologists or physicians should be able to learn to write KAL in a reasonable amount of time.

3) The technical vocabulary must be determined by the user, in an open ended (i.e. continually extensible) manner. Thus, the major parts of speech, used in an appropriate context, should be automatically added to the machine's lexicon. This forces some rigidity on the syntax, but this is not considered detrimental to content, only style. The meaning of any word is given by the totality of sentences in which it has appeared. The skeleton of the syntax and semantics are a reserved group of English* words and phrases (keywords) around which the user may invent arbitrary flesh.

4) If it is to be LISP/CONNIVER based, KAL should interface as simply as possible to those aspects of the host languages which are useful, without becoming subjugated by them. One must resist the temptation to the seduced by the so-called "power" of such languages into doing only what they make "natural", and thereby avoiding what would be more appropriate to the task.

5) Deduction, used to answer questions and to partially check the consistency of new sentences, should rely primarily on a variety of special mechanisms (e.g., set hierarchies, property lists and functional correspondences), and secondly, on the "problem reduction" paradigm a la PLANNER. Question posing is to be in the form of normal declarative sentences having "blanks" to be filled in.

6) Constant attention is maintained while encoding a text for those sentences which cannot be properly handled in the current (static) version of KAL, particularly those which describe actions or change. These are carefully noted in anticipation of a future "dynamic" version.

*They could be just as well chosen from many other languages, suggesting the attractive possibility of easy translation for KAL-based knowleoye.

Major Features of KAL

Noun Phrases and Sets

The user is encouraged to think in terras of set hierarchies. Sets are denoted by noun phrases *(HP)*, in which the noun may be either singular or plural, either denoting the same set. The usual terminal V rule applies for the plural form unless otherwise stated, as in

1 MAN
MEN

Underlined phrases are keywords. The complement of a set does not exist as a set; the notion of a "constraint" (see below) handles negation, as well as certain union-like constructs. Sets are not intersected directly; for example, if CANADIANS (equivalently, CANADIAN) is a set, one may use CANADIAN as an adjective as well (before a noun) and write either of:

2 EVERY CANADIAN MAN IS A CANADIAN

2a CANADIAN MEN ARE CANADIANS

These sentences *are* synonymous, and either creates the subset CANADIAN MEN of CANADIANS. (When discussing KAL sets we use the plural form by convention). It is automatically known to be a subset of MEN. 21 d (equivalently ARE) is a keyword indicating set containment which causes efficient storage of this fact.

NPs may have keyword quantifiers (e.g., EACH, EVERY, SOME, etc.) at the beginning, followed by arbitrarily many adjectives, a noun, and then possibly some subordinate clauses *or* prepositional phrases. One either creates a new set by using a new NP, which must be *defined* to be contained in some existing set, or refers to *an* existing *one.* The following sentences illustrate some *of* these ideas.

3 JOHN IS A MAN

This sentence creates the set MAN if it is new.

4 JOHN BOUGHT CAR/1

JOHN is *an* individual (a singleton), but sets without *proper* names may not be created (e.g., A CAR), hence we invent the name CAR/1 (the machine *may* assign the number). Definite articles are used in restricted contexts; e.g. THE and A may be used to *refer* to a set but not to create one*, as in:

5 THE MAN WHO BOUGHT A CAR |S RICH

The NP here denotes JOHN, since the subordinate clause is used in a pattern matching fashion to locate the X which IS A MAN such that X BOUGHT Y and Y IS A CAR. IS in this context is interpreted as a keyword associating the adjective RICH with the set JOHN. Sets may be second order, i.e., may have sets as instances, and may appear quantified in certain contexts.

*except in certain "generic" contexts.

One-to-One and Many-to-One Correspondence

The user is also encouraged to think 1n terms of one-to-one and many-to-one links, or functional correspondence between instances of sets (be they individuals or other sets). These links arise 1n several ways, e.g., through "For All ... There Exists ..." statements, such as

6 EACH PERSON HAS A HOME

7 EACH WOMAN LIKES HER MEN

HAS is a keyword signifying any functional (many-one) correspondence. Sentence 7 1s an example of second order quantification, i.e., a separate subset of MEN may be inferred to exist for each individual WOMAN. HER 1s a keyword denoting the existential quantifier following the universal EACH.

A second source of correspondence links arises in the notion of structures, i.e. either physical structures (THINGS) or mental ones (IDEAS). The keyverb HAP, for HAS AS PARTS denotes this relation, as in:

8 EACH PERSON HAP 1 BODY, 1 SOUL

The comma construction saves writing 2 sentences. If there are BODIES other than human ones, one must write BODY OF A PERSON to distinguish this one. HAP links are 1-1, unlike HAS. One might define a family thus:

9 EACH COUPLE HAP 1 MAN, 1 WOMAN

10 EACH FAMILY HAP 1 COUPLE, SOME CHILDREN

If it is known that BILL IS A MAN, then

11 SMITHS ARE A FAMILY

12 SMITHS HAP (BILL, MARY, (PETER, ANN))

may be understood, with the machine warning of the assumptions it has made: MARY IS A WOMAN and the extensional set {PETER, ANN} ARE CHILDREN. Such references to individuals are relatively rare 1n technical literature, however, where by individual I mean of course, not a human in general, but a specifically named (proper noun) member of a set. Thus KAL semantics, unlike those of most natural language systems which are oriented towards discussing human relations, are biased towards discussion at the set (generic) level. Its use of articles, (A, THE) which we haven't space to discuss, reflects this.

Adjectives and Attributes

The keyverb IS (or ARE) is used to associate an adjective with a noun. This association may also be (equivalently) effected by placing adjectives before the noun, as In:

13 BIG BLACK BEARS ARE DANGEROUS

which creates the subset of BEARS denoted by DANGEROUS BIG BLACK BEARS, with the adjectives in any order. (The problem of how to deal with the resulting six Intervening subsets of BEARS is an annoying one which I have not yet resolved. Fortunately, three or more IS type adjectives seem to be rare).

Usually, one wishes to associate a property with a set under some named relation, usually called an attribute, which may be introduced by the user by hyphenating with IS:

14 EACH MAN'S SEX-IS MALE

15 EACH APPLE'S COLOR-IS RED

One may also hyphenate IS the other way, as in

16 SKY/I IS-LOCATED ABOVE GROUND/1

The use of prepositions like ABOVE (and cases) is described below. Note that THE SKY is not allowed here - one may only the THE to create a generic noun, not an individual.

Constraints and Incremental Learning

Much of our knowledge comes to us in stages, each being in some way a refinement of what was previously known. Knowledge acquisition systems ought to be designed to accept information in this way, yet little seems to have been done to discover the underlying processes that adults use, though much has been done by psychologists in establishing how concepts evolve in the child. Within AI, the thesis of P. Winston (70) is one of the few places in which learning by stages is investigated, although he does not discuss formal linguistic aspects there to any degree.

In this regard, I have found that it is useful to regard the subject-verb pair, for many verbs, as a function (the verb) - argument (the subject) pair. This is usually useful for verbs which define attributes, and particularly those which have either numerical values or those which may take only a small finite number of values. One can then define a number of keywords called constraints which specify the allowed "values" for these subject-verb pairs. For example, instead of 15, we might prefer:

17 EACH APPLE'S COLOR-IS ONE-OF (RED YELLOW GREEN)

Further examples include:

18 EACH ADULT'S AGE-IS GT(20)

19 RICHARD IS N0NE-0F(H0NEST CLEVER LOVED)

20 CHILDREN LIKE AT-LEAST-ONE-QF(CANDY SPINACH DOCTORS)

21 JUST-ONE-0F(BILL JOHN) IS A LIBERAL

22 THE PRICE OF A CAR IS GT(THE PRICE OF A BICYCLE)

Many details surrounding the use of such constraints must be carefully spelled out, particularly when they are used in the more general ways illustrated here. For example, 20 clearly should mean that each child may have her/his own choice, so that the implicit quantification rules must be stated. It is in these kinds of details that I anticipate the most difficulty both in teaching such a language to non-mathematically experienced persons, and in implementing it. Each contains considerable logical meaning, whose function in the computer must be to assist deductions, and monitor future Inputs. By the latter, I intend that each new statement be monitored so that if it is a further constraint on a previous one, or a contradiction, the user is to be notified. Clearly, it is not possible to do this to arbitrary logical depth; some practical compromise will have to be accepted.

While learning through constraints is an essential part of any knowledge acquisition system, it is given a secondary role in KAL, since it was felt that other considerations are more fundamental.

Time, Change and Actions

The problem of representing change has been given surprisingly little attention in the AI literature. The series of three papers by Hayes (69,70,71) is essentially the only theoretical discussion of the logic of change in this literature. Philosophical logicians have written at length about such matters; I have not been able to find anything in an admittedly imperfect search of this literature (e.g., the books edited by Rescher) which would give a hint as to how to write programs.

One turns then to the current "brute force" approach to change used in AI programming: a situation or context is a group of "true" statements corresponding to a fixed state of the world, and an event (a change or action) is simulated by adding to or removing from this group to produce a new "after" situation. Some recent reports using this approach, originally popularized in STRIPS (Fikes and Nilsson, 71) are Sacerdoti (73), Biss et al (73) and Hendrix et al (73). It is proposed to follow suit by providing such rudimentary descriptions of sequences of events via the context mechanism of CONNIVER. In this respect, the design of KAL will necessarily be influenced by the implementation language, a potential impediment to finding "what you really want" rather than "what's available".

KAL verbs do not at the moment have any "deep semantic" structure associated with them. There 1s no attempt to handle time reference, tense, or verb hierarchies; verbs serve merely as a constant in a pattern, along with appropriate keywords. It is felt that the *very* complex problem of a deep representation of change and the associated natural language structures could not be given prominence without jeopardizing the many other aspects of this work. Thus KAL statements now express that appreciable part of knowledge which is timeless (without reference to change). I term this version "static" KAL.

Verbs: Static Worlds, Cases and Locations

Now I will describe "static" KAL verb structure. The user is directed to choose verbs in the present active plural (e.g., "run", rather than "runs" or "is running", etc.) and to use as obvious ones as possible (e.g., "run" rather than "canter"). Following the verb there are one or more cases, each denoted by certain keywords (usually prepositions), except the "direct object" case, which has no keyword and must come immediately after the verb.* Several alternate construc-

*For a discussion of the linguistic notion of case, see Bruce (73), Kintsch (72), and Fillmore (68).

tions are possible, such as "IS-XXXXED" (e.g., "IS-LOCATED").

The cases are noun phrases, as In:

23 EVERY COMMUTER DRIVE(S) HIS CAR FROM HT5~HOME TO HIS JOB VIA HT| STREFFS" DSTNG HIS CAR

In 23, the direct object HIS CAR, is associated with COMMUTER through a "For all there exists" link, and occurs as well as instrument. The unnatural HIS STREETS is necessary to associate a set of streets with each commuter. Any verb suffix in ( ) is to be ignored by the machine; thus there is no check for subject-verb agreement.

24 EACH AMOUNT-OF LIQUID FLOW(S) FROM ITS SOURCE TO ITS SINK THROUGH ITS CHANRET

In 24, the complexities of dealing with mass nouns are involved. This problem is vital to biological knowledge, and has received no attention in the AI literature. I am trying to deal with it by keywords such as AMOUNT-OF which identifies LIQUID as a mass noun, and creates the equivalent of the individual for discrete nouns.

The prepositional keywords which imply a location in fact refer to the implied location associated with each thing; i.e. there is an implicit "physical" statement:

25 EACH THING HAS A LOCATION

underlying all THINGS. Problems associated with underlying physical assumptions are discussed further below.

Implication Statements

Up to now, the only logical connections between KAL statements (which have all been propositions) are those which are "built in" to the semantics, such as the implications allowed through set hierarchies and correspondence links. To relate arbitrary statements, implication statements are introduced, either with IF ... THEN ..., or IFF, allowing AND or OR groupings of the antecedent. The semantics are borrowed directly from PLANNER. Thus, if we have

26 IF A PERSON IS RICH OR A PERSON IS HFALTHY THENTHAT PElBOfi IS FORTUNATE

then any request to establish \S_ FORTUNATE for any subset of PERSONS will redirect the search (establish the subgoal) for first ^ RICH, or failing that, IS, HEALTHY. This approach is well established, and is easily handled by CONNIVER. THAT is a keyword binding the following PERSON to the previous one, i.e., identical substitutions must be made.

Lacking the mechanism to describe actions as sequences of static contexts, one may nevertheless use chains of implication statements to determine if a particular statement, representing a desired situation, can come about. It 1s felt, however, that this is not sufficiently satisfactory to represent any but rather trivial actions.

It appears that implication statements should only be used within a situation, just as 1n the STRIPS approach, predicate calculus deductions do not transcend changes in the world state.

An Example from Basic Neurology

A few sentences will now be presented which have been taken directly from a basic neurology textbook (Gatz, 71) used by many medical students. After each sentence (in quotes) follows the KAL translation, omitting obvious or trivial sentences for brevity (such as plural definitions), plus a discussion of various features.

"The neuron (nerve cell) is the functional and anatomical unit of the nervous system".

27 EACH NERVOUS SYSTEM HAP SOME NERVE CELLS

This sentence introduces four sets: SYSTEMS and NERVOUS SYSTEMS, CELLS and NERVE CELLS, with the obvious set containment. Note that NERVOUS and NERVE are adjectives. The 1 to 1 link from EACH NERVOUS SYSTEM to ITS own subset of NERVE CELLS is established.

28 A NEURON IS-SAME-AS A NERVE CELL

IS-SAME-AS is a keyverb denoting synonomy. The phrase NERVE CELL thus becomes interchangeable with NEURON.

29 THE FUNCTIONAL UNIT OF THE NERVOUS SYSTEM IS-SAME-AS THOFUffON

The sets UNIT and FUNCTIONAL UNIT are introduced, the latter being related to each NERVOUS SYSTEM by HAP (Implied by OF THE)- FUNCTIONAL UNIT OF THE NERVOUS SYSTEM becomes another name for NEURON.

"Each consists of a cell body (cyton) and one to several dozen processes of varying length called nerve fibers".

30 EACH NEURON HAP 1 CELL BODY, 1 TO 40 PROCESSES

In contexts which take a numerical value, a looser constraint may be used, such as upper and lower bounds. Of course these constraints ideally should not be sharp; the use of such constraints is a rudimentary attempt to introduce fuzzy knowledge. Each neuron now has associated, through 1-1 links, an individual CELL BODY (a subset of BODY) and a subset of PROCESSES not exceeding 40 in size. The comma construction saves writing two sentences. To continue with this sentence:

31 THE CELL BODY OF A NEURON IS-SAME-AS THE moN

32 THE PROCESSES OF A NEURON ARE-SAME-AS JiEffVE FIBERS

33 THE LENGTH OF A PROCESS IS FROM 0 TO T5ff CM.

Only 33 requires comment. Every PROCESS is defined to have associated with it (HAS) a LENGTH, which is known from the remainder bT~the sentence to be a numerical attribute. It may therefore appear hyphenated with IS, as, for example, 1n:

34 PROCESS/1 LENGTH-IS 200 CM

If sentence 34 were entered, the violation of the constraint established in 33 should be noted.

Numerical attributes may carry units.

"Dendrites are short branching fibers which normally receive impulses at their peripheral terminals and conduct them toward the nerve cell body".

35 EACH CYTON HAP SOME PERIPHERAL TERMINALS

36 IF A DENDRITE RECEIVE(S) SOME IMPULSES ST 50ME-0F ITS PERIPHERAL TERMINALS THEN THAT DENDRITE CONDUCT(S) SOME IMPULSES TWRD ITS CYTON.

Here we run into the problem of expressing a dynamic fact (a sequence of events) in static KAL. One must use an implication statement in order to obtain the desired cause-effect relation between the two statements, as discussed earlier. The machine can only use this statement "backwards" (i.e. to search for an appropriate instance of the antecedent to establish an instance of the consequent) since it functions as a CONSEQUENT theorem. What is really intended by this sentence is more like a "demon", "triggering" the assertion of the consequent should the antecedent be asserted. Rather than hastily introduce such machinery (a temptation when using a language like CONNIVER) to appear to have some capability of handling sequences of events, 1 prefer to first gain more experience with the static worlds from which dynamic worlds are probably to be built, and then to introduce problem-determined language (rather than CONNlVER-determined) for events.

Although the two occurrences of DENDRITE are bound by THAT, the two subsets of IMPULSES are not necessarily the same. We are trying to treat the notion IMPULSES as if they are the THINGS when, in fact, they are a sequence of events. One can get away with this to some extent; a proper solution will have to be found in the dynamic version of KAL.

"The term axon, in a strict sense, applies to a single long fiber conducting impulses away from a nerve cell body".

37 X IS, A NERVE FIBER AND
X "LENGTH-IS GT(1 CM"TAND
X CONDUCT(S) SOME IMPULSES AWAY-FROM JTTS CYTON
IFF X IS-SAME-AS AXON

Here we have an appropriate use of an implication statement, in fact an equivalence. (There remains, however, the necessity of treating the conducting of impulses as a static phenomenon). The reader may wonder: "Why not just have three statements about AXONS?" The distinction lies in the manner in which the fact retrieval algorithm ought to treat the two possibilities. Given 37, the algorithm knows these three facts to be tightly associated, i.e., should facts involving AXONS appear during some discourse, these three facts are to be immediately given prominence (over other facts about AXONS) in subsequent processing. The same should occur if, say, two of the three appear, i.e., the third might be sought, to see if we are talking about AXONS.* Given the three facts independently, no such connection would be inferred. The use of X as a bound variable, while looking less like English than previous statements, makes possible the neat conjunction of conditions without messy "English-like" syntax. One should only carry along such syntax as long as its clarity to complexity ratio is rewarding. Statements such as these, involving considerable complexity of at least one side, pose difficult problems for the question-answering algorithm, and should be good counter-examples against a pure top-down (problem reduction, or undirectional heuristic search) procedure.

Related Work and Discussion

Several other projects bear similarities to KAL. W. Martin and coworkers at MIT are developing a language OWL (formerly MAPL) in which to discuss the "world of business". Like KAL, it is set oriented, CONNIVER based, and uses a restricted syntax laced with keywords, though it is difficult for non-LISP programmers to read. It also includes second order sets, functional correspondence links, and case structure. They do not say they work with actual business texts (not surprisingly), nor do they claim running programs. MAPL-OWL appears to be the project most similar to KAL.

Another similar undertaking has been described by Isner (72) and Pople and Werner (72), in which neurological knowledge is encoded in a simplified English, for input to a question answering program written in the LISP extension GOL (Pople, 72), which paralleled some of the innovations of PLANNER. The emphasis seems to have been on using actual scientific knowledge, and on getting a program running. There is no discussion of the syntax of the language, nor of the semantic structures needed for such text. The examples given however are among the most extensive that have appeared of this type of NL programming.

A great deal of careful work has been done by the Swedish group (Sandewall, 70,72; Makila, 72,74; Palme, 70,72,73) in defining various formalisms for representing NL semantics, though they consider general NL. In particular, PCF-2 is a predicate calculus-like notation for a broad range of NL constructs, though it does not appear to have been implemented.

There have been numerous other well known attempts which space does not permit discussing. Most of these try to consider NL in its full generality, particularly in its verb structure. Any of these programs, however, will accept only relatively few NL sentences, and the problem of defining just which these are is often ignored, *or* if attempted, is too complex to be readily understood. The result of attempting such generality would then appear to be that for anyone not deeply involved in the programming, writing a body of text acceptable to the program is virtually impossible. Since KAL has a relatively rigid syntax, and semantics restricted to scientific needs, their description is sufficiently succinct that nonspecialists in computers may learn it, increasing the possibility of such persons being able to communicate with a machine.

*(i.e., we're in the AXON frame, to use the current jargonology).

The main advantage of the "artificial" language approach^ however, is that it focuses attention on the semantic essence of each sentence being translated. While encoding text, one must constantly ask which KAL construct is appropriate, noting those sentences which cannot be adequately represented. This exercise is an essential part of the development of artificial languages, and tends to be neglected when the designers invent their own text.

I have found at least four common types of sentences which do not translate well (or at all) into KAL. Obviously, the most serious limitation is with dynamic descriptions, as we have noted. The assertion of a proposition should not be confused with the simulation of the event it describes, nor should implication statements serve as causal links between statements (which, not being situations, should not be "caused"). A second common problem, also illustrated in the example, is in handling fuzzy information - recall how I substituted a sharp value for the adjective "long". A first step toward alleviating this problem would be to introduce *fuzzy* sets and truth values, which has been done by Le Faivre (74) in his language FUZZY, a CONNIVER-like LISP extension. A third problem concerns complex spatial relationships, a frequent requirement in physiology. Probably, the blocks world approach of listing coordinates will not be feasible for tortuous structures. Until more subtle methods of describing complex objects are developed, such knowledge will have to be expressed In propositions with simple prepositions where possible, and ignored otherwise. The fourth problem involves imbedded sentences - sentences which refer to other sentences. I have made no attempt to consider this problem in KAL; Moore (73) has discussed some possible approaches.

Expressing the underlying laws of physics (which include the "frame problem") that a machine must know to make decisions which seem trivial for even animals, is well recognized as one of the most complex tasks in representing knowledge in computers. One such law (statement 25) was briefly touched upon earlier. This is a static law, and is expressable in a KAL statement. There are many such laws, usually implications involving prepositions. One may imagine, for example, a transitivity law in KAL, the kind one finds in blocks worlds. The problem, of course, is to define for the machine all the laws it will need.

It is doubtful that this can be done in any other way than the way it is done now, at least *for* the foreseeable future. That Is, one must postulate a set of laws and monitor the machine's actions to detect errors, omissions and contradictions. (Note that questions of logical consistency become extremely difficult for languages such as KAL in which we have no metatheorems). I suggest, though I do not yet have the experience to confirm it, that debugging a system which operates on KAL statements should be far easier than one using a less transparent notation, in which the human must perform a task which is *wery* unnatural: manipulation of abstract symbols. Since the meaning of a KAL statement can be comprehended at a glance, one can quickly sift through a page of definitions looking for suspicious ones. Thus the syntactic and semantic "distance", if I may coin such a term, between the CONNIVER expressions and the corresponding KAL sentences should be many times "less" than the distance between more general NL sentences and LISP encodings of the equivalent predicate calculus, to take an extreme example. This transparency of notation should be of particular value when it comes to dealing with dynamics, which the state of the AI programming art suggests are to be modelled by situations as defined earlier, since the number of statements that the human must monitor is greatly increased. Rather than conjecture further in this direction, I would like to discuss one additional point.

There is a whole range of knowledge that cannot be expressed even in a dynamic KAL without enlarging it so much as to make it a complete "programming" language. I have in mind the so-called "procedural knowledge", which amounts to introducing the notions of "execution", "flow of control", "variables", etc. If a language is to be practically usable by nonspecialists in computers, 1t must not become as complex as the most complex "programming" language (e.g. CONNIVER). Hence there exists a very murky area of language definition which would be adequate for procedural knowledge without becoming incomprehensible to non-computer specialists.

There is one possible direction in which a language like KAL might be extended without making it too complex, and which I Intend to try to integrate with the dynamic extension. I have in mind the type of facility for simulating dynamics provided by the discrete simulation languages such as GPSS and SIMULA. Persons relatively unsophisticated in computer science seem to derive considerable benefit in describing dynamic phenomena in these languages, and in interacting with "simulation models". It would seem that a synthesis of the AI approach to dynamics and the "simulation" approach is due.

Another possible advantage *of* an artificial language for science should be mentioned in closing. Scientific discourse, both informal and formal, is often alarmingly vague and verbose. Should a language like KAL become well enough developed, sden* tists might be persuaded to express their findings and conjectures directly 1n it, eliminating much volume and ambiguity. The development of global computer networks would make possible instant sharing of such knowledge, and the problem of having the surface structure of KAL-like language look like most NLs is, of course, trivial. Thus scientists of many tongues might communicate directly and clearly. The alternative for such networks is to allow arbitrary NL to flow through them, a measure sure to cause communication blockages as severe as our present paper system.

But that is too futuristic.

Epilogue (June 1975)

In the paper, at the time of writing (Dec. 74) it was indicated that the intention was to attempt to implement some of KAL in CONNIVER. Subsequent experience actually trying to use CONNIVER resulted in the following conclusions:

CONNIVER is not sufficiently well documented and debugged to make using it at "remote" sites (where "remote" probably means "off the ARPA net") a rewarding experience. CONNIVER is possibly adequate for certain toy world problems, but is too inefficient to be a suitable language in which to implement any practical natural language system. This was no surprise, since others had made similar observations. It was felt, though, that the experience was worthwhile so as to better understand the concepts and the pitfalls. A better alternative may exist in the language 2.pak (see the paper by L.F. Melli in this Proceedings).

Acknowledgment

The author gratefully acknowledges the support of the Lkf. Killam Memorial Fund of the Montreal Neurological Institute.

References

1. Biss, K., Chien, R., Stahl, F., Weissman, S. Semantic Modeling for Deductive Question Answering* 31JCAI, pp 356-363, 1973.

2. Bruce, B. Case Structure Systems. 3IJCAI, pp 364-371, 1973.

3. Fikes, R., and Nilsson, N. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. Artificial Intelligence 2, nos. 3/4, pp 189-208, 1971.

4. Filmore, C.J. The Case for Case, in: Universals in Linguistic Theory, (eds: Bach, E. and Harms, R.T.), New York: Holt, Rinehart and Winston 1968.

5. Gatz, A.J. Manter's Essentials of Clinical Neuroanatomy and Neurophysiology. Philadelphia, F.A. Davis, 1971.

6. Hayes, P.J. Robotologic. in: Machine Intelligence 5 (eds. Meltzer, B., and Michie, D.), Edinburgh: Edinburgh University Press, 1970.

7. Hayes, P.J. A Logic of Actions, in: Machine Intelligence 6 (eds. Meltzer, B. and Michie, D.), Edinburgh: Edinburgh University Press, 1971.

8. Hendrix, G.C., Thompson, C.W. and Slocum, J. Language Processing via Canonical Verbs and Semantic Models. 3IJCAI, pp 262-269, 1973.

9. Isner, D.W. An Inferential Processor for Interacting with Biomedical Data Using Restricted Natural Language. AFIPS Conference Proceedings, vol. 40, pp 1107-1123, 1972.

10. Kintsch, W. Notes on the Structure of Semantic Memory. 1n: Organization of Memory, (eds.: Tulving, E., and Donaldson, W.), New York and London: Academic Press. 1972.

11. Le Faivre, R. Fuzzy Problem Solving. (PhD Thesis) Report no. 37, Academic Computing Center, The University of Wisconsin, Sept. 1974.

12. Makila, K. Deduction Procedures in a Question Answering System. March, 1972,*.

13. Makila, K. Storage and Retrieval of Information in an SBP Data Base. June, 1974*.

14. Martin, W. et al. Various memos on the MAPL and OWL projects, available from the author, Project MAC, MIT.

15. McCarthy, J., and Hayes, P.J. Some Philosophical Problems from the Standpoint of Artificial Intelligence, in: Machine Intelligence 4 (eds. Meltzer, B. and Michie, D.) Edinburgh: Edinburgh University Press, 1969 .

16. Moore, R. D-Script: A Computational Theory of Descriptions. 3IJCAI. pp 223-234, 1973.

17. Palme, J. A Simplified English for Question-Answering. Dec. 1970*.

18. Palme, J. From Parsing Tree to Predicate Calculus - A Preliminary Survey. Feb. 1972*.

19. Palme, J. Syntax and Dictionary for a Computer English. July, 1972*.

20. Palme, J. The SQAP Data Base for Natural Language Information. Sept. 1973*.

21. Pople, H. A Goal Oriented Language for the Computer, in: Representation and Meaning-Experiments with Information Processing Systems, (eds. Simon, H. and Siklossy, L.) Englewood Cliffs, N.J.: Prentice Hall, 1972.

22. Pople, H., and Werner, G. An Information Processing Approach to Theory Formation in Biomedical Research. AFIPS Conference Proceedings, vol. 40, pp 1125-1138, 1972.

23. Rescher, N. (ed). The Logic of Decision and Action. Pittsburgh: Pittsburgh University Press, 1967.

24. Rescher, N. and Urquhart, U. (ed.). Temporal Logic. New York and Wein: Springer-Verlag, 1971.

25. Sacerdoti, E. Planning in a Hierarchy of Abstraction Spaces. 3IJCAI, pp 412-422. 1973.

26. Sandewall, E. A Data Base Structure for a Question-Answering System. 1970*.

*These are all reports published by the Research Institute of National Defense, Operations Research Center, S-10450 Stockholm 80, Sweden.

cont'd____

27. Sandewall, E. PCF-2, A First-order Calculus for Expressing Conceptual Information. Uppsala, Sweden, 1972.

28. Winston, P. Learning Structural Descriptions from Examples. PhD Thesis, MIT AI Lab, 1970, (Report AI TR-231).

N.B, 3IJCAI stands *for* the Third International Joint Conference on Artificial Intelligence, August, 1973 (Stanford, Calif.). Enquiries for the Proceedings of the Conference should be addressed to: Stanford Research Institute, Publications Dept., 333 Ravenswood Ave., Menlo Park, Calif., 94025.