Book Reviews

NETL: A System for Representing and Using Real-World Knowledge

Scott E. Fahlman

The MIT Press, Cambridge, Mass., 1979, 285 pp., $17.50, ISBN 0-262-0609-8.

This is a "slightly edited version" of Fahlman's 1977 Ph.D. dissertation. Fahlman presents a knowledge representation organized as "a pattern of interconnections of very simple parallel processing elements: node units which can store a dozen or so distinct marker-bits, and link units which can propagate these markers from node to node, in parallel through the network" [p. 1]. Fahlman considers it to be important that "using these marker-bit movements, the parallel network system can perform searches and many common deductions very quickly: the time required is essentially constant, regardless of the size of the knowledge-base" [p. 1]. However, this timing result depends on the existence of parallel network hardware. An appendix unconvincingly presents a design of such hardware, but the body of the book presents a knowledge representation structure which can be considered on its own merits, independently of such timing considerations.

NETL is a paradigm example of what I have called path-based reasoning [Shapiro, S.C. Path-based and node-based inference in semantic networks. In D. Waltz, ed., *TINLAP-2,* ACM, New York, 1978, 219-225. Also in *AJCL,* Microfiche 80 (1978), 38-44.], but it is worth noting that what Fahlman calls "links" would more properly, I feel, be called "nodes", and his true links are what he calls "wires". Fahlman says, "I have spoken of the node and link units as separate entities, and will continue to do so, but the difference is largely one of emphasis ... every link has an associated *handle-node* that represents the statement that the link implements ... The nodes, in turn, sometimes need a few link-type wires for special-purpose connections to other nodes ... Since every true link has a handle-node and every normal node has a few special link-wires, it seems only natural to use the same piece of hardware to represent them both. I call this piece of hardware an *element* when I want to refer to it without indicating whether it represents a node or a link. An element has the marker-bit storage and the single terminal of a node (the handle-node if it is representing a link) and a set of connecting wires that it uses in the manner of a link" [p. 35-36, italics in the original].

The central organizing principle of NETL is a property inheritance hierarchy using nodes representing typical members of sets. For over ten years, I have thought that the "typical-member" technique of representing information was very attractive, but that it couldn't be made to work in general. Fahlman's is the best attempt I have seen to use this technique, and he discusses all the problems. Finally, however, he uncovers problems which convince him that the technique is inadequate. Considering the attractiveness of the idea, and Fahlman's valiant attempt to get it to work, it is worthwhile to devote a substantial part of this review to analyzing the technique, Fahlman's approach, and the problems he discusses, in the hope that any future researchers tempted to try the technique will do so with a full knowledge of where the problems lie.

The first problem, inadequately handled by many researchers who discuss inheritance hierarchies, is that sets and individuals (equivalently, the set membership relation and the subset relation) must be distinguished. As Fahlman points out, "The set of elephants has a certain size, expressed as the number of elephants in the world; the typical elephant has a certain size expressed in meters or kilograms" [p. 16]. Note, also, that one set can be both an element of a second set and a subset of a third set. Fahlman handles this problem by having two types of nodes: **INDV-nodes* for individuals and **TYPE-nodes.* A set is represented by a *INDV-node with an associated *TYPE-node representing the typical member of the set. An actual member of the set is represented by a *INDV-node with a *VC (for *virtual-copy)* link to the *TYPE-node. Fahlman uses *VC rather than ISA, or the like, because he likes to think of the implicit information about an individual as being virtual copies of the information stored about all *TYPE-nodes of those sets the individual is a member of. To summarize, using Fahlman's well known example, CLYDE is a *INDV-node representing the elephant Clyde; CLYDE has a *VC relation to TYPICAL-ELEPHANT, a *TYPE-node representing the typical elephant; TYPICAL-ELEPHANT is associated with (by having a set-wire to) ELEPHANT-SET, a *INDV-node representing the set of all elephants. The set of elephants may be made an element of the set of sets by placing a *VC link from ELEPHANT-SET to TYPICAL-SET, and the set of elephants may be made a subset of the set of mammals by placing a *VC link from TYPICAL-ELEPHANT to TYPICAL-MAMMAL.

The next problem arises from the simplistic slogan that nodes inherit properties from higher nodes in the hierarchy. The problem is that if TYPICAL-ELEPHANT has the relation R to node N, does CLYDE inherit the relation R to N, or to some modified copy of N? Consider "all elephants hate all aardvarks." This can be stored as the HATES relation between TYPICAL-ELEPHANT and TYPICAL-AARDVARK. If each elephant inherits the HATES relation to TYPICAL-AARDVARK, it works correctly, as does inheriting the RESPECTS relation to DARWIN if we represent "all elephants respect Charles Darwin" by a RESPECTS relation from TYPICAL-ELEPHANT to DARWIN. However, we would not want to handle "every elephant has a trunk" or "every elephant has a mother" this way. This problem involves both the representation and the way inheritance is done. One would not want to give TYPICAL-ELEPHANT either TYPICAL-TRUNK or a specific individual trunk. Fahlman solves this problem by having another kind of node, a *role-node,* which has an *EXISTENCE-link* to a type-node. "A description, then, consists of a *base-node* representing the thing itself (TYPICAL-ELEPHANT is the base-node for its description) and a set of *role-nodes,* connected to the base-node by EXISTENCE-links and representing the various things that every copy of the base-node has one of. Most role-nodes specify individual entities, but there are also *set-roles* (with attached type-nodes) for representing such things as the typical elephant's set of teeth and the typical member of this set" [p. 26-27, italics in the original].

The typical-member technique may be thought to allow a more economical representation than some variant of predicate calculus. Note, however, that a *TYPE-node is a universally quantified node ranging over the set represented by its associated set-node, and a role node is an existentially quantified node whose EXISTENCE-link is a Skolem arc pointing to the universally quantified node it depends on.

So now consider the role-node TYPICAL-ELEPHANT'S-LEG-SET, and its associated *TYPE-node TYPICAL-ELEPHANT'S-TYPICAL-LEG. If we want to know something about Clyde's left front leg that is inherited from TYPICAL-ELEPHANT and TYPICAL-ELEPHANT'S-TYPICAL-LEG, no additional structure need be built. Fahlman's virtual copy technique allows us to view TYPICAL-ELEPHANT'S-TYPICAL-LEG virtually as Clyde's left front leg. However, if exceptional information is to be stored about Clyde's left front leg, a node is needed to represent it. There are now two choices: actually copy all the non-exceptional information about TYPICAL-ELEPHANT'S-TYPICAL-LEG down to CLYDE'S-LEFT-FRONT-LEG, or let CLYDE'S-LEFT-FRONT-LEG continue to inherit non-exceptional information. Fahlman goes with the latter approach, which is more economical -- information needn't be copied -- and allows new non-exceptional information to be stored and inherited with no extra work. The problem now is what is the relation between CLYDE'S-LEFT-FRONT-LEG and TYPICAL-ELEPHANT'S-TYPICAL-LEG? It cannot be *VC, because that would make CLYDE'S-LEFT-FRONT-LEG one of the legs of the typical elephant, not one of Clyde's legs. In fact, if general information is to be stored about elephants' left front legs, we need TYPICAL-ELEPHANT'S-LEFT-FRONT-LEG, an *INDV role-node with *VC to TYPICAL-ELEPHANT'S-TYPICAL-LEG, and we cannot confuse this with Clyde's left front leg. Fahlman's solution is to make CLYDE'S-LEFT-FRONT-LEG another kind of node, a **MAP-node,* with a *map-wire* to TYPICAL-ELEPHANT'S-LEFT-FRONT-LEG. Meanwhile, CLYDE'S-TYPICAL-LEG is a **TMAP-node* with map-wire to TYPICAL-ELEPHANT'S-TYPICAL-LEG. All *MAP and *TMAP nodes also have an *owner-wire* to the individual which owns them, paralleling the EXISTENCE-link of their parents. Because of Fahlman's marking inheritance scheme, it is not necessary to connect CLYDE'S-LEFT-FRONT-LEG to CLYDE'S-TYPICAL-LEG as well. I leave it to the reader to examine the following alternative proposal. Give CLYDE'S-LEFT-FRONT-LEG a *VC link to TYPICAL-ELEPHANT'S-LEFT-FRONT-LEG. This requires TYPICAL-ELEPHANT'S-LEFT-FRONT-LEG to be a *TYPE-node, which is not the case in NETL. The associated set-node would then be ELEPHANT'S-LEFT-FRONT-LEG-SET, a node representing the set of all left front legs of elephants, and quite distinct from the set of left front legs of the typical elephant, for which a node would not exist since it is a singleton set. ELEPHANT'S-LEFT-FRONT-LEG-SET is not a subset of TYPICAL-ELEPHANT'S-LEG-SET, because the latter represents the set of legs of the typical elephant, whereas the former represents the set of left front legs of all elephants. They both must be subsets of the set of legs of all elephants, represented by ELEPHANT-LEG-SET, a set-node with associated type-node TYPICAL-ELEPHANT-LEG. Certainly, one role-node of the description based on TYPICAL-ELEPHANT-LEG is TYPICAL-ELEPHANT-LEG'S-ELEPHANT. It would be nice if this were just our old friend TYPICAL-ELEPHANT but this does not seem to be the case since TYPICAL-ELEPHANT is a base-node and TYPICAL-ELEPHANT-LEG'S-ELEPHANT is a role-node. From a predicate calculus point of view, what we have done is construct formulas for "For every elephant there is a leg ..." and "For every elephant leg there is an elephant ..."

If one of the advantages of the typical-member technique seems to be the ability to collapse *VC chains, and thus see an individual as a virtual copy of its hierarchical ancestors, consider what Fahlman calls the "copy-confusion" problem. This problem appears in several guises. In one, we try to find the weight of

Clyde's trunk. CLYDE and CLYDE'S-TRUNK are both *VCs of TYPICAL-PHYSICAL-OBJECT, which has a weight as one of its role-nodes. In the process of collapsing *VC chains we lose the distinction between Clyde's trunk's weight and Clyde's weight. In another version of the problem, the typical family has both a father and a child. Since Clyde is the father in one family and a child in another, copy-confusion causes Clyde to be seen as his own father. Fahlman considers four solutions to this problem. For the first solution "I am not yet sure whether it is impossible to do this, or just very difficult. A second possibility is to abandon the virtual-copy semantics ... I feel that this approach should be taken only as a last resort ... A third approach ... seems a needlessly complex and economically unattractive solution ... the fourth solution is the one that seems to me the most promising, and is the one that I am using in the current version of NETL ... A few possible relevant nodes will be missed by this approach ... The parallel portions of the system are not complete, in the logician's sense, but they were never intended to be; we wanted to be able to do the most important deductions very fast, and I believe that NETL still does that" [p. 148-53]. Even the fourth solution seems to compromise on some features earlier considered beneficial, and even on the virtual-copy idea itself, "In some cases, it is useful to create pseudo-individuals to fill roles in the middle of long role-chains; this tends to break up the chains into more manageable sections" [p. 152].

Fahlman feels that "The copy-confusion problem, in its various guises, is principally a problem of properly implementing an essentially correct semantic notation in a parallel manner; the binding-ambiguity problem, on the other hand, results from a shortcoming of the semantic notation itself" [p. 153]. This problem arises when, in a predicate calculus approach there would be a statement with two or more universally quantified variables ranging over the same set. Consider the HATES relation between TYPICAL-ELEPHANT and TYPICAL-ELEPHANT. Does this mean that every elephant hates himself, or that every elephant hates every elephant including himself? What about "every elephant hates every elephant other than himself"? Fahlman decides by fiat (there is no other way) that it means the first. To get the third, he introduces **OTHER-nodes.* Every *OTHER-node has a *type-wire* to a *TYPE-node and represents every other element of the associated set. "Every elephant hates himself" is represented by TYPICAL-ELEPHANT HATES TYPICAL-ELEPHANT. "Every elephant hates every other elephant" is represented by TYPICAL-ELEPHANT HATES OTHER-TYPICAL-ELEPHANT. The second, "every elephant hates every elephant including himself," is represented by the conjunction of the two. If more than two are needed, "We might, for instance, want to say that for any *three* distinct elephants, X, Y, and Z, the combination of X and Y can defeat Z at tug-of-war" [p. 158, italics in original], there is a chain of *OTHER-nodes, each one representing every member of the set except for any earlier one. "The scans responsible for all of this can become extremely complex as they try to consider all possible bindings, but that seems reasonable: beyond two or maybe three variables, people -- especially children -- have trouble as well. The notation, at least, seems clear and unambiguous, even if the associated processes are difficult to construct and use" [p. 158]. From my point of view, it seems as though the typical-member technique was pushed so far that it simply broke down. It is thought to be elegant and economical, but it needs patch upon patch. Perhaps the alternatives are indeed better.

Fahlman, himself, goes on to introduce another representational technique, the *defined class.* Consider asserting "Every purple mushroom is poisonous." Using a predicate calculus-like representation, there would be no problem, but Fahlman, of course, wants a more object oriented representation. One could create a new subset of the set of mushrooms, and have its typical member purple and poisonous. However, this does not distinguish this set from the representation of "every poisonous mushroom is purple" or even from "a set of purple, poisonous mushrooms." The defined set uses a new node type, the **EVERY-node,* a special kind of *TYPE-node. A *EVERY-node has two kinds of information stored about it, a *specification* and some *incidental information.* Clauses in the specification are tied to their *EVERY-node by a *spec-wire.* There can also be *INDV-nodes in the specification tied to the *EVERY-node by a spec-wire. These nodes "serve as additional variables to use during the matching process" [p. 161] when a new individual node is being tested to determine if it is in the defined class. We can recognize this kind of spec-wire as another Skolem arc, and analyze Fahlman's defined class as a representation of the statement schema (Ax){(Ey)S(x,y) -> I(x)}, where S(x,y) is the specification, I(x) is the incidental information, x is the *EVERY-node, and y is a *INDV-node tied to x by a spec-wire. The relation of this object oriented representation to proposition oriented representations can now be seen by considering statements (deduction rules, production rules) of the form (Ax){S(x) -> I(x)}. A proposition oriented representation would use a node to represent the statement, with arc types to differentiate the antecedent(s) from the consequent(s), and to indicate the universally quantified variable. An object oriented representation such as Fahlman's has no node representing the statement as a whole, has a specially typed node representing the universally quantified variable, an arc type between the variable and the antecedent(s), and no special arc between the variable and the consequent(s). Both representations have nodes representing the propositions of the antecedent(s) and the consequent(s). Looked at in this way, Fahlman's restriction that "No clause may be part of the specification of more than one *EVERY-node" [p. 161] can be seen as eliminating the possibility of representing statements of the form (Ax)(Ay) {S(x,y) -> I(x,y)}.

Before concluding, I must make one comment on a passage which I fear illustrates a common attitude among AI researchers. Fahlman points out that, "If an object is said to *exist* within an area, that means that it exists *somewhere* within the area; if a statement is said to be *valid* within an area, it means that it is valid *everywhere* within the area" [p. 114, italics in the original]. He then comments that "This difference in the behavior of existence and statement-scoping came as something of a surprise to me -- I am told that the logicians have known something of this sort all along, but they tend not to express it in these terms. This difference caused a lot of trouble until I understood what was happening" [p. 115]. The first point is that this is a nice statement of the reason AI researchers should study logic. Perhaps, if he had, Fahlman would have been saved a lot of trouble. The other point is that if a researcher discovers that he has rediscovered a distinction or issue already discussed by others in different terms, he should rewrite his own discussion in the existing terms, giving the appropriate credit and providing the connections to his readers, not dismiss the older work in a couple of sentences and contribute to the proliferation of jargon. The issue is no less than the respectability of AI.

What makes these complaints more serious is that this book represents a good piece of work, already having an impact on the "knowledge representation" community. It is certainly *must* reading for anyone also involved in the inheritance hierarchy brand of representation.

Stuart C. Shapiro, SUNY-Buffalo

Understanding Spoken Language

Donald E. Walker, Editor

Elsevier North-Holland, New York, 1978,
420 pp., Paperback, $9.95, ISBN 0-444-00287-1.

In 1970 the Advanced Research Projects Agency decided to fund six research projects aimed at developing systems that were capable of understanding connected speech. From 1971 to 1975, this research was carried out. This book is a collection of articles (most of which have been published separately) which grew out of the final report of the speech understanding group at SRI International. Despite its title, its stated purpose is to describe SRI's speech understanding system rather than speech understanding in general, and it contains much material pertinent to understanding written language as well as speech.

Although the introductory and concluding material attempts to unify the book, it remains a collection of very separate articles rather than a unified whole. As a consequence it suffers from the common problems of books of this type: inadequate cross referencing, poor transitions between chapters, and no index. The absence of an index is a serious problem that is compounded by the fact that the table of contents contains only three levels of structure; more detailed outlines of the contents are found at the beginning of each chapter. Fortunately the references have been merged into a single list. The reference list is very good, in part because it is not too long to scan easily.

The signal processing part of speech understanding is given the barest mention because SRI did not do work in that area; the book concentrates on the higher level aspects of the understanding process.

The first chapter is a nicely written introduction and overview by Donald Walker. It describes the organization of the ARPA speech understanding effort and outlines the SRI system. The second chapter, by William Paxton, quickly plunges the reader into a rather detailed description of the language definition system which was used to define the language that the system would understand. These definitions were then compiled into a form that the executive system, which controlled the other components of the system, would understand. The language definition consisted of a lexicon (words and "multiwords" with grammatical categories, grammatical features, and associated semantic information) and composition rules (phrase structure rules augmented by procedures to be executed whenever the rule constructs a phrase). The procedures gave values to attributes of the phrase as a function of the attributes of its constituents and judged the acceptability of the phrase on a number of grounds such as acoustic properties, syntactic properties (such as mood and number), semantic properties (using the semantic network representation discussed further on), and discourse information to handle anaphora and ellipsis. Much of the complexity of the language definition derives from the fact that it must screen out bad input rather than just recognize good input as many grammars do.

In discussing the executive system itself, much space is devoted to historical background, comparisons with other speech understanding systems, and the experimentation (using analysis of variance) that was