From: AAAI-82 Proceedings. Copyright ©1982, AAAI (www.aaai.org). All rights reserved.

# Competence in Knowledge Representation

Ronald J. Brachman
Hector J. Levesque

Fairchild Laboratory for Artificial Intelligence Research
Palo Alto, CA

## §1 Introduction

The range of domains and tasks for "knowledge-based systems" has been expanding at a furious pace. As we move away from trivial domains, such as the "blocks world", the demands on knowledge representation systems used by expert programs are becoming more extreme. For one thing, the domains themselves are getting so complex that specialized technical vocabularies are unavoidable; consequently, the issue of a system ***talking with an expert in his own language*** cannot be ignored. For another, tasks such as medical diagnosis, scene analysis, speech understanding, and game playing all have as a central feature an ***incrementally evolving model*** representing probably incomplete knowledge of part of the task domain. In this paper, we explore some of the impact of these two critical issues—complexity and incompleteness—on knowledge representation systems. We review some aspects of current representation research that offer a foundation for coping with these problems, and finally suggest a way of integrating these ideas into a powerful, practical knowledge representation paradigm.

## §2 Two Kinds of Adequacy

A major concern when trying to represent an expert's knowledge about a sufficiently complex domain is the management of the ***technical vocabulary*** of that domain. In the blocks world, the technical vocabulary amounted to a handful of terms (like "*block*" or "*color*") almost all of which were primitive. The ability of an expert in medicine or VLSI design, on the other hand, appears to depend in large part on a grasp of a non-trivial number of interrelated terms. In fact, a significant part of the job of becoming an expert is acquiring the technical jargon necessary to understand other experts.

Consider, for example, some of the terms used by an expert in VLSI design.

- an "*enhancement mode transistor*" is a ***kind*** of transistor with specific electrical properties.
- a "*pass transistor*" is any transistor that plays a certain ***role*** in a larger circuit.
- a "*barrel shifter*" is a ***structured configuration*** of components with a certain functionality.
- "*two-phase nonoverlapping clocking*" is a ***method*** of organizing the timing in a circuit.

In order to behave knowledgeably in a real domain, a system will have to interact with experts using specialized terms like the above. Therefore, the application of knowledge representation to expert problems demands of a representation system the ability to ***develop***, ***augment***, and ***maintain*** this kind of technical vocabulary. As the above examples suggest, a representation scheme must allow the introduction of terms that deal with different aspects of the domain: objects, properties, methods, rules, heuristics, and so on. Moreover, it must be possible to specify terms that are related to each other in several different and complex ways.

A second aspect of expert tasks like medical diagnosis is the ***incremental acquisition*** of world knowledge. In the blocks world, a system could be given complete knowledge of the domain in that, for example, it could assume that the only blocks on a table were those it had been told about. Knowledge about the world in a high level recognition task, on the other hand, will be acquired gradually and, at any given time, may be quite incomplete.

Consider the knowledge acquired by a system at some stage of its processing.

- a speech understanding system might know that a phoneme under consideration is a /*b*/ ***or*** a /*v*/, but not which.
- a scene analyzer might theorize that ***some object*** other than the pyramid is casting a

shadow on a box without having identified that object.

- a medical diagnosis consultant may establish that the cause of the metabolic acidosis is *not* shock but not know what the cause is.

A knowledge representation system for tasks like these must have the ability to express what is known about the world (i.e., to form and maintain a ***theory***) however incomplete this knowledge may be. One important consequence of this involves the form of the representation itself: in the case of VLSI design, for example, it may not be possible to simply construct an analogue of a chip where representational objects stand for fragments of the chip and relationships between these objects correspond to relationships between the pieces of the chip. Incomplete knowledge about a domain (in the form of a theory) need not look at all like the domain itself.

The requirements in dealing with realistic application domains amount to this: there are at least two distinct notions of adequacy that a knowledge representation system for expert tasks will have to address. ***Terminological adequacy*** involves the ability to form the appropriate kind of technical vocabulary and understand the dependencies among the terms; ***assertional adequacy*** involves the ability to form the kind of theory appropriate to the world knowledge of a system and understand the implications of the theory. In other words, terminological adequacy means getting the right kind of structured terms and assertional adequacy means putting the terms together to properly express what is known, however incomplete.[1]

## §3 Aspects of Current Research

One of the places to look for assertional adequacy is in the standard first order logical languages. These appear to be ideal for representing knowledge that is potentially very incomplete. For example, one can assert that there is something with a certain property without having to say what thing has that property (existential quantification), or that one of two sentences is true without saying which (disjunction). Indeed, one view of logic is that it is a formal account of what is implicit in statements where very little is explicit.

[1]The issue of ***defaults*** seems to involve both areas. The need for defaults is certainly based on the need for action given incomplete world knowledge. On the other hand, at least one interpretation of defaults is such that it is part of understanding what is involved with a term to know what the typical (and atypical) cases are.

But however assertionally adequate a first order language may be, it has the property that there are no constraints or interdependencies forced on the interpretation of the non-logical (predicate, function, or constant) symbols. In other words, the domain dependent terms in a classical first order language are all completely ***independent***. One can certainly form a theory stating, for example, that every "*enhancement mode transistor*" is a "*transistor*", but the language itself cannot be made to enforce this dependency. Once it is agreed that the theories of a first order logic are intended to represent world knowledge, there is simply no place left to introduce terminology.

With at least some "object-centered" languages (i.e., semantic net and frame approaches), the situation seems completely reversed. To the extent that they allow theories to be formed at all, these tend to be very limited. In particular, the assumption that knowledge is based on explicit representations for all objects of interest rules out sentences that do not have specific individuals as the objects of predication. The emphasis, rather, seems to be on facilities for structuring and organizing a set of terms (frames, concepts, units, classes). There might be various ways of specializing existing terms, of aggregating multiple terms into single structures, of delineating the roles played by the components of a structure and of separating the essential from the merely prototypical in the use of a term. This is done quite independently of any theory about the world. For example, calling any transistor that plays a certain role in an inverter a "*pull down transistor*" is not merely asserting that, as a matter of fact, one set of electrical components is the same as another (as would be the case with a universally quantified biconditional); rather, it involves defining a term so that it is analytically true that any component playing that role is a pull down transistor.

## §4 A Unifying Approach

Given the almost independent concerns of the two (perhaps caricatured) representation methodologies just mentioned, a natural organization for a knowledge representation system suggests itself: combine both representation paradigms into a single, unified framework. Moreover, the unification we are considering is not simply an unstructured amalgamation of two languages (nor is it simply maintaining an alternate, predicate calculus form of representation for a semantic net). The object-centered part will be responsible for organizing in a convenient way the relevant domain dependent terms being used; the logical part, on the other hand, will use these terms to form theories of the application domain. Each of the components can have its own purpose, methodology, and standards of adequacy. Since neither is required to deal with issues for which it was not designed, its advantages can be enjoyed and its limitations minimized.

The elements of each of the two components follow straightforwardly from our separation of concerns. The assertional component consists of a theory:[2] a set of sentences in a logical language (e.g., that of first order predicate logic). The terminological component is a set of terms composed from other terms using a small set of term-composition operators (e.g., those of KL-ONE [1], [5]). These terms then become the non-logical (predicate and function) symbols within the assertional component. The fact that the terms are definitionally interrelated means that the non-logical symbols of the assertional component stand in various analytical relationships to each other.

For example, we might have within the terminological component a KL-ONE Concept called "*transistor*" with a Role "*connection*" differentiated into a "*source,*" a "*drain,*" and a "*gate.*" We can then define a term called "*pull down transistor*" as

(**VRGeneric** *transistor source GND*)

meaning a transistor whose source connection is GND. **VRGeneric** is a term-forming operator that allows a Generic Concept to be defined by ***value restricting*** a Role (in this case, *source*) of a superConcept (here, *transistor*). As with any other KL-ONE Concept, we can ask for the superConcepts or the Roles of *pull down transistor.*

But in addition, within the assertional component, *pull down transistor* becomes a ***unary predicate*** that can be used to make various assertions. Looking at a design, for example, we might want to say that

$$\exists x \exists y (PullDownTransistor(x) \wedge Transistor(y) \wedge Terminal(drain(x), y)),$$

in other words, that there is a pull down transistor whose drain is a terminal of some (yet to be identified) transistor. The key property of this predicate is that any theory using it would automatically contain an additional postulate pertaining to the meaning of the term:

$$\forall x (PullDownTransistor(x) \equiv (Transistor(x) \wedge source(x) = gnd))$$

which guarantees (by virtue of the rules of logical implication) that every pull down transistor is a transistor and that every pull down transistor is connected to GND.

In general, associated with each term-forming operator is an axiom schema stating what the impact of the term in the assertional component should be. In general, for a definition of a Concept $\theta$ of the form

$$\theta \leftarrow (\textbf{VRGeneric}\ \psi\ \xi\ \phi),$$

we would have a ***meaning postulate*** of the form

$$\forall x (\theta(x) \equiv (\psi(x) \wedge \forall y (\xi(y, x) \supset \phi(y)))),$$

which is instantiated for different values of $\psi$, $\xi$ and $\phi$ (as above for *pull down transistor*).[3]

Another Concept forming operator under investigation is **NRGeneric** which allows a Concept to be defined by ***number restricting*** a Role of an existing Generic Concept. The terminological component also includes an operator **PrimGeneric** for forming ***primitive*** terms, especially useful for representing "natural kinds" [2]. For instance, we could define *transistor* by

*transistor* ← (**PrimGeneric** (**NRGeneric** *device connection* 3 3)).

That is, a transistor is, among other things, a device with at least three and at most three connections.[4]

The keystone of our approach is the specification of all sentences that are logically implied by the introduction of both Concepts and Roles in the terminological component. A similar treatment can even be applied to "default" operators. The major difference is that the meaning postulate must refer to the current state of knowledge. For example, instead of having "*all pullups are depletion mode*", the postulate might imply that "*all pullups are depletion mode except those* known *to be enhancement mode.*[5]

[2]We have chosen to deal with a single theory within the assertional component at this stage. This is a pragmatic choice and others would be possible within our methodology.

[3]This is a bit of a simplification since the form of the meaning postulate actually depends on the form of $\xi$ and $\phi$. For example, Roles correspond to either binary predicates or unary functions depending on their cardinality.

[4]**PrimGeneric** is one of the term-forming operators whose meaning postulate is not a quantified biconditional.

[5]See [3] for a formal account of sentences such as these.

## §5 A Note on Competence

Our feeling is that a knowledge base is not simply a passive repository of assertions about a domain; rather, it is best viewed as actively providing a certain service to the knowledge-based system in which it is embedded. The knowledge base can answer questions for the system based on its theory of the application domain. Moreover, it is the responsibility of the knowledge base to revise that theory as knowledge is accumulated.

The competence of a knowledge base can therefore be defined functionally in terms of two operations: TELL takes a knowledge base and an assertion and produces a new knowledge base; ASK takes a knowledge base and a question and produces a suitable answer. In fact, the service provided by a knowledge base is precisely its behavior under these two operations. Following the discipline of abstract data types, the rest of the system can only tell the knowledge base about its world (or define terms) and ask about that world (or terms). It cannot manipulate the contents of the knowledge base directly and has no information about the storage schemes used.

The actual definition of competence in the assertional component is based on logical implication: what the system "knows" (i.e., the answers it provides to questions) includes what follows from what it has been told. Similarly competence in the terminological component depends on closure under subsumption (i.e., the system "knows" when one term conceptually contains another). In addition, because TELL can refer to the current state of knowledge (as in the "default" example above), it cannot be defined simply as adding an assertion to the knowledge base; like ASK, it requires deductive capabilities. Moreover, ASK can also refer to the state of knowledge to allow the system to find out where the knowledge base is incomplete (see [3] for more on this). So while implication and subsumption obviously play an important role, the competence of the system is ultimately defined in terms of the more general TELL and ASK operations.

## §6 Conclusion

We have presented a view of knowledge representation that distinguishes between terminological and assertional competence, and advocates understanding a terminological component in terms of its impact on an assertional one. We see a number of advantages arising from this segmentation of the knowledge representation task. First of all, it allows the effort to be cleanly separated into two distinct subtasks with a well defined interface between them. For example, the terminological component can be extended in any number of ways provided the meaning postulates implied by the new constructs are always specified. Similarly, the expressive power of the assertional component can be refined without side-effects to the terminological operators. Also, the choice of data structures and algorithms can be made separately for each component and can be sensitive to the special needs of the component without adversely affecting what is done in the other. For example, a KL-ONE style classifier [4] can be used to implement terminological subsumption while still allowing a more general theorem prover to be used in the assertional component. But most importantly, by dividing the knowledge base into the two components, we can best answer the needs of knowledge-based expert capabilities by preserving the terminological features of object-centered languages without sacrificing the assertional expressive power of the logical ones.

**Acknowledgements**

Many of these ideas—especially technical refinements to the terminological framework—were developed in concert with Dan Bobrow, Richard Fikes, Austin Henderson, and Mark Stefik. Special thanks also to David Israel for many fruitful discussions about the issues raised in this paper.

## References

[1] Brachman, R. J., Bobrow, R. J., Cohen, P. R., Klovstad, J. W., Webber, B. L., and Woods, W. A. ***Research on Knowledge Representation for Natural Language Understanding, Annual Report***, BBN Report No. 4274, Bolt Beranek and Newman Inc., Cambridge, MA, 1979.

[2] Israel, D. J. "On Interpreting Network Formalisms." To appear in ***The International Journal of Computers and Mathematics***, Special Issue on Computational Linguistics. Forthcoming.

[3] Levesque, H. J. ***A Formal Treatment of Incomplete Knowledge Bases.*** Technical Report No. 3, Fairchild Laboratory for Artificial Intelligence Research, Palo Alto, CA, 1982.

[4] Lipkis, T. "A KL-ONE Classifier." In [5].

[5] Schmolze, J. G., and Brachman, R. J., ***eds.***, ***Proceedings of the 1981*** KL-ONE ***Workshop.*** Technical Report No. 4, Fairchild Laboratory for Artificial Intelligence Research, Palo Alto, CA, 1982. (Also, BBN Report No. 4842.)