AI Magazine Volume 12 Number 3 (1991) (© AAAI)

Knowledge Interchange Format: The KIF of Death

Matthew L. Ginsberg

There has been a good deal of discussion recently about the possibility of standardizing knowledge representation efforts, including the development of an interlingua, or knowledge interchange format (KIF), that would allow developers of declarative knowledge to share their results with other AI researchers.

In this article, I examine the practicality of this idea. I present some philosophical arguments against it, describe a straw-man KIF, and suggest specific experiments that would help explore these issues.

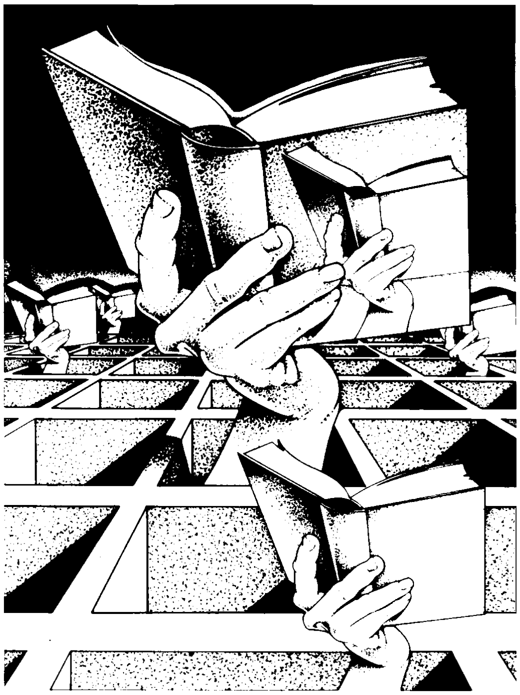
There has been a flurry of interest recently in the possibility of standardizing existing work on knowledge representation; this interest is supported by the Defense Advanced Research Projects Agency (DARPA) and other funding agencies. One focus of this effort has been the development of an interlingua, or knowledge interchange format (KIF), by which knowledge could be moved from one declarative system to another.

This article touches on a variety of issues. In Philosophical Arguments, I discuss general concerns about both the practicality of developing KIF and the effect that a partially successful development might have on the knowledge representation community in general. An Interlingua Proposal attempts to address the technical concerns raised, presenting a straw-man KIF proposal that might provide a way around these difficulties. In An Experiment, I discuss a specific translation experiment that, I hope, will settle some of the questions raised in the first two sections. Finally, Standards Effort or Research? discusses the question of whether the development of KIF should be viewed as a standardization or a research effort and the consequences of the natural conclusion.

Philosophical Arguments

An examination of recent work on knowledge representation makes it clear that there are deep differences among the approaches taken. It is these differences that underlie our expectation that attempts to automatically translate between one knowledge representation scheme and another are premature.[1]

Those supporting knowledge representation standards are attempting to address this difficulty by creating a single language in which all knowledge representation schemes can be expressed (Genesereth 1990), but this task seems impossible given the current state of the field. Perhaps it is possible to construct a single language that can



0738-4602/91/$4.00 ©1991 AAAI

capture all existing knowledge representation schemes (although I expect that such a language would be an unmanageable Hydra). However, it is surely not possible to construct a language that will also incorporate all future knowledge representation work, other than in the trivial sense guaranteed by the universality of some specific method, such as first-order logic or a general-purpose programming language. Furthermore, attempts in this direction will inevitably constrain future knowledge representation efforts; even gentle constraints might have a stifling impact on future knowledge representation work. This position is especially true given that it is to our funders' advantage to encourage the emergence of a standard (even a premature one, sad to say) and that increasingly tight AI funding will compel researchers to conform to any perceived standard that does exist.

However, let me return to my remarks regarding the universality of some specific methods, such as first-order logic or a general-purpose programming language. Why wouldn't one of these methods serve as a knowledge interchange format?

With regard to the use of a programming language, this approach is little more than a suggestion that researchers exchange knowledge by swapping both the knowledge bases that they use and the programs that manipulate these knowledge bases. This sort of exchange is too limited to be of interest; the aim of the standardization effort is to allow the users of inference tools developed by one group direct access to the knowledge developed by another group.

The argument against using first-order logic as KIF is rather different. The reason is not that it will be impossible to translate from any specific knowledge representation formalism into a universal language such as first-order logic but that connections to special-purpose inference mechanisms (semantic networks or relevance logic, for example) are likely to be lost by doing so. As an example, a translation into first-order logic of a probabilistic database will, of necessity, include an axiomatization of probability itself. Many knowledge representation schemes involve other extensions to first-order logic, such as nonmonotonic or limited-resource reasoning. Again, because the intent of KIF is to allow users of one language (and the associated inference mechanism) access to the knowledge encoded by others, this problem needs to be addressed.

Some of these extensions are treated in Genesereth (1990), and some are not. However, the point is that the treatments that do exist are—and inevitably will be—hopelessly idiosyncratic. The presentation in Genesereth (1990) deals with nonmonotonic reasoning using ideas that are less than a year old and with quotation-using techniques that have never even appeared in the literature. I am not saying that these ideas are without merit, only that their inclusion in a standard is obviously premature.

I return to this issue in Standards Effort or Research. In the meantime, my own approach to the idiosyncrasies of Genesereth (1990) is to describe a language that refuses to commit on any topic that is the focus of current knowledge representation research. I do this by acknowledging that different development groups will use different knowledge representation schemes and that if they are to exchange their information with other researchers, they will need some way to remove their conventional operators or methods. Ideally, however, backward pointers will be retained so that groups that share ideas can recover one another's original formulations. Therefore, the proposed interlingua is a minimalistic language that is nearly equivalent to first-order logic, with specific notational conventions introduced to allow users of special-purpose methods to include them.

Note also that the adoption of this sort of language—one with a facility that allows individual users to extend it as necessary—addresses concerns regarding an interlingua's negative impact on the field as a whole because novel ideas can be developed and freely included in the overall language. All sorts of approaches to knowledge representation problems are encouraged, not just the specific ones that were selected by the KIF designers. The minimality of the core language prevents it from being used as an overall standard, and it is clear from the outset that no lofty goals are being set.

An Interlingua Proposal

As stated earlier, one can take two approaches to the problem of anticipating in KIF all the possible methods that might be used by developers of special-purpose knowledge engineering techniques. On the one hand, one might try to design KIF that anticipated special-purpose techniques by including specific facilities to describe existing approaches. On the other hand, one might accept special-purpose methods as currently outside the scope of information that can legitimately be exchanged and resign oneself to the ensuing

… the basic KIF language will be first-order predicate calculus…

inaccuracies in exchanged information.

This article embodies a strong commitment to the second of these two approaches. As already argued, the reason is simply that the state of the art in knowledge representation work does not support any other option.[2] However, given that KIF cannot contain a declarative description of special-purpose techniques, what should it contain? I suggest the following:

First, the basic KIF language should be one that is accepted by the entire knowledge representation community. Thus, it is ensured that any recipient of a KIF database can do something useful with the knowledge received.

Second, it should be possible to extend the language as the knowledge representation community comes to accept more things as standard. In addition, if some subgroup comes to share specific understandings, it should be able to include these understandings in its KIF databases, although this inclusion should be transparent to any users that do not share them.

The way in which I propose to achieve these goals is as follows:

First, the basic KIF language will be first-order predicate calculus, with specific notational conventions to handle variables, quantification, and so on. This basic language is described in Basic KIF. Note that I am subscribing to the syntax of first-order logic only, not necessarily to its semantics.

Second, all KIF documents will include information indicating the knowledge representation commitments made by the underlying system. Any recipient who can support all the commitments made by the sender is assured that no information has been lost in the transmission of the system; recipients who cannot support all the commitments can receive approximate information in some sense (see Semantics).

Third, the additional information passed by KIF documents making specific knowledge representation assumptions is clearly separated from the declarative information that is also being passed. This approach ensures that a recipient not sharing these commitments retains partial access to the knowledge being sent. The syntax used for the additional information is described in Semantics.

Overall Syntax

A KIF document is a sequence of ASCII characters that can be processed by a Lisp reader; we concern ourselves here with the results returned by the reader instead of the character stream itself, viewing a KIF document as a stream of s-expressions. The following s-expressions have a special meaning in KIF:

First, any keyword (that is, any top-level s-expression that is an atom beginning with a ":") other than :label, :cancel, and :nonsemantic is the name of a knowledge representation commitment made by the system. The next s-expression should be used to convey additional information, as appropriate (see Semantics). These keyword–s-expression pairs will typically contain system-dependent information concerning subsequent sentences.

Second, :label should be followed by an atom that labels the last keyword encountered and also indicates that this keyword is "sticky" in that it applies to all subsequent sentences until canceled. The keyword :cancel should also be followed by an atom and indicates that the keyword information so labeled should not be applied to the sentences that follow. Once again, examples are in Semantics.

Third, :nonsemantic indicates that the most recent keyword has no semantic content (for example, control information; see Semantics for an example). This keyword indicates that a system that doesn't understand the previous keyword is still safe in using the information.

All other s-expressions are interpreted as basic KIF sentences with a declarative semantics.

Basic KIF

The legal sentences in basic KIF are precisely those of first-order logic, expressed in prenex normal form (in other words, the usual representation of sentences from first-order logic using a Lisp-like syntax).

To be precise, we need to specify the fashion in which variables are distinguished from constants as well as specify alphanumeric replacements for symbols such as ¬, ∀, and ∃.

Variables. Any atom beginning with the character ? is a variable. *Sequence variables*, which can match arbitrary sequences of atoms instead of just single ones, are also supported. There is substantial evidence from the PROLOG community that such variables are useful in constructing declarative databases.

Unlike the developers of PROLOG, the Interlingua Group does not assume that sequence variables appear only at the end of argument lists or that only one sequence variable appears in any particular expression. This approach makes unification more difficult than it typically is; however, it is possible to construct a unifier that handles this more general problem and only incurs significant computational overhead when attempting to unify expressions that actually do contain multiple sequence variables. This unifier is available using anonymous ftp from t.stanford.edu.

Any variable whose second character is * is a sequence variable; all other variables are not. Thus, ?* and ?*x are examples of sequence variables; ? and ?x are nonsequence variables. The character * was chosen because of the obvious analogy with BNF expressions; I have not used a distinct initial character for sequence variables because it is important that all variables have a uniform and easily identified representation.

Connectives. The replacement of connectives with alphanumeric expressions is shown in the following table of standard logical sentences and their KIF equivalents:

Standard	KIF
$\neg p$	(not p)
$p_1 \wedge \ldots \wedge p_n$	(and p1 … pn)
$p_1 \vee \ldots \vee p_n$	(or p1 … pn)
$p \supset q$	(if p q)
$p \leftrightarrow q$	(iff p q)
$\forall xy.p(x,y)$	(forall (?x ?y) (p ?x ?y))
$\exists xy.p(x,y)$	(exists (?x ?y) (p ?x ?y))

The first argument to ∀ and ∃ can be an atom if only a single variable appears under the scope of the quantifier.

Semantics

The syntax of KIF is as given in the preceding two subsections; s-expressions following keywords have no restrictions (hence the inherent flexibility of this proposal), but the syntax of other sentences is dictated by the rules of first-order logic.

In the absence of information to the contrary, the semantics of a KIF database are also given by the rules of first-order logic; the extension facility describes situations in which first-order logic is inappropriate for some system-dependent reason.

The form of the additional information used by a KIF database is

```
:extension s-exp,
```

where :extension is the name of a particular extension to first-order logic, and s-exp is additional information that is to be used when considering the subsequent sentence or sentences.

It cannot be within the scope of any specific KIF effort to delimit the list of possible extensions; although I will give a variety of examples showing the flexibility of the scheme, it should not be in the scope of this effort to even propose such extensions. Rather, group members expect that these extensions will be developed by the designers of the various knowledge bases themselves; as common agreements are reached by these designers, they will settle on specific extensions that sanction them.

In the remainder of this section, I consider the use of the extension facility to treat four separate semantic extensions: definitions, probabilities, procedural attachments, and relevance logic (Belnap 1977). I attached "mlg" to the extension names to indicate that they are mere trial balloons that might be supported by only a single researcher. In some cases, I would hope that the community interested in the ideas (probabilities, for example) would rapidly settle on an agreed language that would describe shared semantic commitments.

Definitions A KIF document that contained definitions might look like the following:

```
:definition-mlg predicate
:nonsemantic
:label  d001

(forall   ?x (iff  (bachelor ?x)
                   (and (single ?x) (male ?x))))
(forall   ?x (iff  (old-maid ?x)
                   (and (single ?x) (female ?x)
                        (old ?x))))
:cancel d001.
```

The definition-mlg keyword indicates that the following sentences are to be interpreted according to a particular set of rules that are agreed on by all users of the definition-mlg package. In this particular case, the following atom, predicate, indicates that the rules are being used to define predicates—bachelor and old-maid in the example. The label d001 is used to delimit the scope of the definitional declaration.

Any system that recognizes the definition-mlg commitments is now free to interpret the previous two rules as definitions, presumably obtaining some computational benefit as a result. A system that doesn't recognize these commitments will still be able to make sense of the rules because the nonsemantic keyword indicates that the keyword doesn't affect the meaning of the following information. Other control information (indicating that a certain rule is to be used for backward chaining, for example) can similarly be handled.

Probabilities The interesting extensions are those that actually change the semantics of the knowledge in some way; for example:

```
:probability-mlg .75
(flies Fred)

:probability-mlg .80
(forall ?x (if (bird ?x) (flies ?x)))

(forall ?x (if (ostrich ?x) (bird ?x))) .
```

This database tells us that Fred flies with probability 0.75, birds fly with probability 0.8, and ostriches are birds. The interpretation of the sentence "Birds fly with probability 0.8" is up to the conventions of probability-mlg; this particular system might assume that the probability actually labels the entire sentence or might remove the leading quantifier and interpret the result as a conditional probability.

Now imagine that this knowledge base is sent to a system that has no probabilistic facility. The recipient, on seeing the probability-mlg keyword, will presumably realize that the database being received uses probabilistic information. One of three things can then be done:

First, the probabilistic information might be accepted as valid, concluding in this example that Fred flies and so do birds. The resulting database is likely to contain information not intended by the sender but this option might nevertheless be the most attractive one available.

Second, the probabilistic information might be ignored, retaining only the statement that ostriches are birds. Only knowledge thought by the sender to be valid is used, but the overall value of the information received might be reduced.

Third, probability-mlg might be subscribed to in that some sense is made of the additional information being received. Perhaps any sentence stated with probability in excess of some threshold should be accepted and the rest ignored. Perhaps a default reasoning facility will be used instead, and so on.

Of course, the point is that the recipient of the system is able to interpret the incoming information in a convenient and rational fashion. Note, incidentally, that the first two options described here (accept or ignore suspect sentences) can be done automatically; the third requires some sort of human intervention.

Many other existing declarative schemes can similarly be handled. Assumption-based truth maintenence systems, for example, need to label sentences either as assumptions or not; nonmonotonic reasoning schemes typically need to label sentences as defaults.

Procedural Attachment and Relevance Logic. Suppose that we want to evaluate the truth of the predicate subsetp using Lisp as opposed to evaluating it inferentially. We might write this evaluation as

```
:attach-mlg    subsetp
:label a001 .
```

This statement presumably means that subsetp is procedurally attached. Because the label is never canceled, all subsequent information might be affected. Alternatively, we might have

```
:attach-mlg    lisp!
:label a002 ,
```

indicating that any Lisp predicate without side effects is procedurally attached in this way.

Once again, other examples are similar. If we intend that the database be interpreted using Belnap's (1977) four-valued relevance logic instead of conventional methods, we might write

```
:relevance-ndb nil
:label b001 .
```

A modal operator of knowledge with an S5 semantics might lead us to write

```
:modal-sak     (L S5)
:label s001 ,
```

but if L is described truth functionally instead, we would have

```
:modal-mlg     (L lisp-fn)
:label g001 ,
```

where lisp-fn is the function that actually does the computation. If mlg and sak manage to resolve their differences, they might settle on a single package, :modal.

Summary of Proposal. Standards are appropriate only where there is consensus.

Is the KIF work to be viewed as a standards effort or as research?

The aim of the interlingua that I propose is to standardize what really is standard: the fact that existing declarative databases are largely —but not wholly—equivalent to collections of first-order sentences. The inability to transmit declarative information between large systems might reflect more the fact that there is no common notation than the existence of fundamental differences between the systems being used.

I have not tried to settle the differences that do exist because these differences are not yet ripe for standardization. Instead, I demonstrated a general way for users to include their nonsemantic knowledge when transmitting declarative databases. I showed translation examples of several nonstandard knowledge representation mechanisms using these methods and hope that this approach will encourage the emergence of common understandings in these areas from the knowledge representation and engineering community itself.

An Experiment

Will it work? Is it reasonable to expect it to be possible to transmit useful knowledge from one group to another using the above interlingua or any other, or are there simply too many special-purpose idiosyncrasies to which any practical system is committed?

The developers of the hypothetical KIF standard (Genesereth 1990) proposed answering this questioning by using their standard to translate a declarative knowledge base from one special-purpose language to another. This experiment proves nothing; however, the same group is translating both into and out of KIF!

To see why this assumption invalidates the KIF experiment, suppose that we take a semantic network language, translate it into an interlingua, and then translate back into a semantic network. Presumably, the first translation will produce sentences such as

```
(subclass ostrich bird)
```

that will need to be retranslated into a semantic network.

How is an automatic translation mechanism to recognize the subclass predicate? What if is-a is used instead? Is the translator into the interlingua expected to provide a declarative description of the predicates used (for example, this subclass is transitive and so on)? Is the translator out of the interlingua supposed to recognize this description? There is no obvious solution to any of these difficulties; they are only obscured by an experimental validation of the KIF idea that fails to isolate the users of the special-purpose languages involved.[3]

To experimentally validate the KIF idea, I suggest the following: Two interlingua developers should be found. Each should write translators to or from the other interlingua, not a special-purpose language. Then, for a variety of special-purpose languages L, one of the interlingua groups should write a translator from L into the interlingua, and the other group should write a translator from their interlingua back into L. Declarative knowledge bases written in L should be mapped into one interlingua, then to the other interlingua, and finally back to L. If the result is equivalent to the original knowledge base (in the sense of the range of conclusions drawn by the special-purpose system), then the interlinguas performed successfully. This experiment avoids the problems of the earlier one because the translations to and from L are developed by isolated researchers.

A drawback to this proposal is that it requires the development of two distinct interlinguas. It might seem more natural to ask two independent research groups to write translators into and out of a single interlingua and then use their results to translate from one language to the other and back again. I have not suggested this approach because the standard developers are all too likely to blame the (to my mind, inevitable) failure of this experiment on inadequacies in the translators developed by the independent research groups. Furthermore, if these groups were to turn to the KIF designers for help, it would invalidate the requirement that they work independently. It is for this reason I require that the interlingua teams write the translators, at least initially, which means that we need at least two interlinguas to work with.

Although I expect this experiment to fail, attempting it is an important activity; the attempt itself amounts to an experimental evaluation of the question of whether knowledge representation is ripe for standardization. Such an experiment—viewed only as research and not as a standardization effort in and of itself—would doubtless add to the understanding of the current state of the art in declarative systems.

Standards Effort or Research?

Is the KIF work to be viewed as a standards effort or as research?

I expect that it will be clear from the arguments that were made throughout this article that I feel that the current state of knowledge representation work makes it impossible to view the development and evaluation of KIF as anything but research. As stated in note 2, this opinion was also expressed by a unanimous vote of the people attending the Knowledge Representation Standards Panel at the 1991 International Conference on Knowledge Representation and Reasoning.

This observation has some important consequences. First, the work on KIF should not be reported as a standard because those funding the work may misperceive what has been accomplished. The work should be reported as research and funded and evaluated as research as well. In this vein, I would like to applaud the tone taken in the article by Neches et al. in this same magazine issue; there, it is made clear that research is being pursued and that standards are not. I am far less sympathetic with Genesereth (1990), who claims no fewer than 17 times that a standard is being developed.

If we view the KIF effort as research, what exactly is the question being investigated? Clearly, the question is one of knowledge interchange: Is it practical for one group to make use of the declarative knowledge that was obtained and encoded by another?

This question is important, and it should be investigated. However, the development of a wide-ranging KIF such as that pursued in Genesereth (1990) has little to do with this question! If neither of the groups trying to exchange knowledge use nonmonotonic reasoning, including it in a KIF is completely immaterial to the question of whether these groups can effectively share knowledge.

The argument for a wide-ranging KIF is that it can potentially reduce the effort needed for n groups to exchange knowledge from $o(n^2)$ to $o(n)$. In the first case, each group needs to write a translator into each other group's language; given the KIF, translators into and out of KIF are all that is needed.

However, this is not research! The research question is whether knowledge sharing is possible at all; reducing the amount of effort needed from $o(n^2)$ to $o(n)$ is a standards issue, not a research one—therein lies the fundamental difference between the ideas presented in this article and those presented in Genesereth (1990). By proposing a minimalistic language and describing specific experiments, I am trying to further a research investigation into the practicality of knowledge sharing. By proposing the adoption by fiat of a specific maximalistic language, Genesereth (1990) is clearly in the business of forming a standard.

One of these activities is appropriate. The other is not.

Acknowledgments

I want to thank Jim Bennett, Ron Brachman, Johan deKleer, Robert Engelmore, David Etherington, Richard Fikes, Ken Forbus, Don Geddis, Pat Hayes, David Israel, Ben Kuipers, Bill Mark, John McCarthy, Karen Myers, Peter Patel-Schneider, Ramesh Patil, Elaine Rich, Rich Washington, and the members of Stanford's PRINCIPIA Group for discussing these issues with me. The suggestions of several anonymous reviewers were also helpful. However, the opinions expressed are mine.

References

Belnap, N. D. 1977. A Useful Four-Valued Logic. In *Modern Uses of Multiple-Valued Logic*, eds. G. Epstein and J. Dumm, 8–37. Boston: Reidel.

Genesereth, M., et. al. 1990. Knowledge Interface Format, Technical Report, Logic-90-04, Dept. of Computer Science, Stanford Univ.

Notes

[1] These differences make the interlingua effort different from typical standardization efforts in computer science (for example, standardization of programming languages). DARPA is aware of this fact and has discouraged any analogy between the development of KIF and the development of a language such as Common Lisp.

[2] For what it's worth, this opinion is not only mine as author but is also the opinion expressed by the attendees at the 1991 International Conference on Knowledge Representation and Reasoning. I return to this issue in Standards Effort or Research?.

[3] I do not mean to imply that this problem will not arise using the language proposed in An Interlingua Proposal, only that I have provided a clear method by which different groups can attempt to express their individual representational commitments.

Matthew L. Ginsberg received his doctorate in mathematics from Oxford University in 1980 at the age of 24. He was on the faculty at Oxford until 1983, doing research in mathematical physics and computer science; during this period, he wrote a program that was successfully used to trade stock and stock options on Wall Street. Ginsberg's continuing interest in AI brought him to Stanford University in late 1983. He is currently a senior research associate at Stanford. His interests include real-time planning and problem solving, multivalued and modal logics, and nonmonotonic reasoning. He is the author of numerous publications in these areas, the editor of *Readings in Nonmonotonic Reasoning* and the author of the forthcoming *Foundations of Artificial Intelligence*.