

A HYBRID APPROACH TO REPRESENTATION IN THE JANUS NATURAL LANGUAGE PROCESSOR

Ralph M. Weischedel
BBN Systems and Technologies Corporation
10 Moulton St.
Cambridge, MA 02138

Abstract

In BBN's natural language understanding and generation system (Janus), we have used a hybrid approach to representation, employing an intensional logic for the representation of the semantics of utterances and a taxonomic language with formal semantics for specification of descriptive constants and axioms relating them. Remarkably, 99.9% of 7,000 vocabulary items in our natural language applications could be adequately axiomatized in the taxonomic language.

1. Introduction

Hybrid representation systems have been explored before [9, 24, 31], but until now only one has been used in an extensive natural language processing system. KL-TWO [31], based on a propositional logic, was at the core of the mapping from formulae to lexical items in the Penman generation system [28]. In this paper we report some of the design decisions made in creating a hybrid of an intensional logic with a taxonomic language for use in Janus, BBN's natural language system, consisting of the IRUS-II understanding components [5] and the Spokesman generation components. To our knowledge, this is the first hybrid approach using an intensional logic, and the first time a hybrid representation system has been used for understanding.

In Janus, the meaning of an utterance is represented as an expression in WML (World Model Language) [15], which is an intensional logic. However, a logic merely prescribes the framework of semantics and of ontology. The *descriptive constants*, that is the individual constants (functions with no arguments), the other function symbols, and the predicate symbols, are abstractions without any detailed commitment to ontology. (We will abbreviate *descriptive constants* throughout the remainder of this paper as *constants*.)

Axioms stating the relationships between the constants are defined in NIKL [8, 22]. We wished to explore whether a language with limited expressive power but fast reasoning procedures is adequate for core problems in natural language processing. The NIKL axioms constrain the set of possible models for the logic in a given domain.

Though we have found clear examples that argue for more expressive power than NIKL provides, 99.9%

of the examples in our expert system and data base applications have fit well within the constraints of NIKL. Based on our experience and that of others, the axioms and limited inference algorithms can be used for classes of anaphora resolution, interpretation of highly polysemous or vague words such as have and with, finding omitted relations in novel nominal compounds, and selecting modifier attachment based on selection restrictions.

Sections 2 and 3 describe the rationale for our choices in creating this hybrid. Section 4 illustrates how the hybrid is used in Janus. Section 5 briefly summarizes some experience with domain-independent abstractions for organizing constants of the domain. Section 6 identifies related hybrids, and Section 7 summarizes our conclusions.

2. Commitments to Component Representation Formalisms

We chose well-documented representation languages in order to focus on formally specifying domains and using that specification in language processing rather than on defining new domain-independent representation languages.

A critical decision was our selection of intensional logic as the semantic representation language. (Our motivations for that choice are covered in Section 2.1.) Given an intensional logic, the fundamental question was how to support inference for semantic and discourse processing. The novel aspect of the design was selecting a taxonomic language and associated inference techniques for that purpose.

2.1. Why an Intensional Logic

First and foremost, though we had found first-order representations adequate (and desirable) for NL interfaces to relational data bases, we felt a richer semantic representation was important for future applications. The following classes of representation challenges motivated our choice.

- Explicit representations of time and world. Object-oriented simulation systems were an application that involved these, as were expert systems supporting hypothetical worlds. The underlying application systems involved a tree of possible worlds. Typical questions about these included *What if the stop time were 20 hours?* to set up a possible world and run a

simulation, and *In which situations is blue attrition greater than 50%?* where the whole tree of worlds is to be examined. The potential of time-varying entities existed in some of the applications as well, whether attribute values (as in *How often has USS Enterprise been C3?*) or entities (*When was CV22 decommissioned?*) The time and world indices of WML provided the opportunity to address such semantic phenomena (though a modal temporal logic or other logics might serve this purpose).

- **Distributive/collective quantification.** Collective readings could arise, though they appear rare, e.g., *Do USS Frederick's capabilities include anti-submarine warfare* or *When did the ships collide?* See [25] for a computational treatment of distributive/collective readings in WML.
- **Generics and Mass Terms.** Mass terms and generally true statements arise in these applications, such as in *Do nuclear carriers carry JP5?*, where JP5 is a kind of jet fuel. Term-forming operators and operators on predicates are one approach and can be accommodated in intensional logics.
- **Propositional Attitudes.** Statements of user preference, e.g., *I want to leave in the afternoon*, should be accommodated in interfaces to expert systems, as should statements of belief, *I believe I must fly with a U.S. carrier*. Since intensional logics allow operators on predicates and on propositions, such statements may be conveniently represented.

Our second motivation for choosing intensional logic was our desire to capitalize on other advantages we perceived for applying it to natural language processing (NLP), such as the potential simplicity and compositionality of mapping from syntactic form to semantic representation and the many studies in linguistic semantics that assume some form of intensional logic.

However, the disadvantages of intensional logic for NLP include:

- The complexity of logical expressions is great even for relatively straightforward utterances using Montague grammar [21]. However, by adopting intensional logic while rejecting Montague grammar, we have made some inroads toward matching the complexity of the proposition to the complexity of the utterance; that simplicity is at the expense of using a more powerful semantic interpreter and of sacrificing compositionality in those cases where language itself appears non-compositional.
- Real-time inference strategies are a challenge for so rich a logic. However, our hypothesis is that large classes of the linguistic examples requiring common sense reasoning can be

handled using limited inference algorithms on a taxonomic language. Arguments supporting this hypothesis appear in [2, 13] for interpreting nominal compounds; in [6, 7, 29], for common sense reasoning about modifier attachment; and in [32] for phenomena in definite reference resolution.

This second disadvantage, the goal of tractable, real-time inference strategies, is the basis for adding taxonomic reasoning to WML, giving a hybrid representation.

2.2. Why a Taxonomic Language

Our hypothesis is that much of the reasoning needed in semantic processing can be supported by a taxonomy. The ability to pre-compile pre-specified inferential chains, to index them via concept name and role name, and to employ taxonomic inheritance for organizing knowledge were critical in selecting taxonomic representation to supplement WML.

The well-defined semantics of NIKL was the basis for choosing it over other taxonomic systems. A further benefit in choosing NIKL is the availability of KREME [1], which can be used as a sophisticated browsing, editing, and maintenance environment for taxonomies such as those written in NIKL; KREME has proven effective in a number of BBN expert system efforts other than NLP and having a taxonomic knowledge base.

In choosing NIKL to axiomatize the constants, one could use its built-in, incomplete inference algorithm, the classifier [27]. In Janus, the classifier is used only for consistency checking when modifying or loading the taxonomic network; any concepts or roles identified by the classifier as identical are candidates for further axiomatization. Our semantic procedures do not need even as sophisticated an algorithm as the NIKL classifier; pre-compiled, pre-defined inference chains in the network are simpler, faster, and have proven adequate for NLP in our applications.

2.3. Two Critical Choices in the Hybrid

2.3.1. Representing Predicates of Arbitrary Arity

Choosing a taxonomic language, at least in current implementations, means that one is restricted to unary and binary predicates. However, this not a limitation in expressive power. One can represent a predicate P of n arguments via a unary predicate P' and n binary predicates, which is what we have done. $(P r_1, \dots, r_n)$ will be true iff the following expression is.

$$(\exists b) (\wedge (P' b) (R_1 b r_1) (R_2 b r_2) \dots (R_n b r_n))$$

Davidson [5] has argued for such a representation of processes on semantic grounds, since many event descriptors appear with a variable number of arguments.

2.3.2. Time and World Indices

Any concept name or role name in the network is a constant in the logical language. We use concepts only to represent sets of entities indexed by time and world. Roles are used only to represent sets of pairs of entities, i.e., binary relations. Given time and world indices potentially on each constant in WML, we must first state the role those indices play in the NIKL portion of the hybrid.

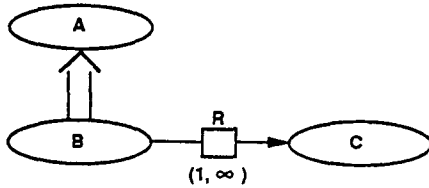


Figure 1: Two Typical Facts Stated in NIKL

In a first-order extensional logic, the normal semantics of SUPERC and of roles in NIKL are well defined [26]. For instance, the diagram in figure 1 would mean

$$(\forall x)((B x) \supset (A x))$$

$$(\forall x)((B x) \supset (\exists y)(\wedge(C y) (R x y))).$$

Due to a suggestion by David Stallard, we have chosen to interpret SUPERC and the role link similarly, *but interpreted under modal necessity*, i.e., as propositions true at all times in all worlds. Thus in the diagram in Figure 1, $(A z)$, $(B z)$, $(C z)$, and $(R x y)$ are intensions, i.e., functions with arguments of time and world $[t, w]$ to extensions. Rewriting the axioms above by quantifying over all times and worlds, the axioms for the diagram in Figure 1 in the hybrid representation are

$$(\forall x)(\forall t)(\forall w)((B x)[t, w] \supset (A x)[t, w])$$

$$(\forall x)(\forall t)(\forall w)((B x)[t, w] \supset (\exists y)(\wedge(C y)[t, w] (R x y)[t, w])).$$

Though this handles the overwhelming majority of constants we need to axiomatize, it does not allow for representing constants taking intensional arguments because the axioms above allow for quantification over extensions only.¹The semantics of predicates which should have intensions as arguments are unfortunately specified separately. Examples that have arisen in our applications involve changes in a reading on a scale, e.g., *USS Stark's readiness downgraded*

from $C1$ to $C4$.² We would like to treat that sentence as:

$$(\wedge (\text{DOWNGRADE } a) \\ (\text{SCALE } a (\text{INTENSION Stark-readiness})) \\ (\text{PREVIOUS } a C1) \\ (\text{NEW } a C4)).$$

That is, for the example we would like to treat the scale as intensional, but have no way to do so in NIKL. Therefore, we had to annotate the definition of *downgrade* outside of the formal semantics of NIKL. Only 0.1% of the 7,000 (root) word vocabulary in our applications could not be handled with NIKL. (The additional problematic vocabulary were *upgrade*, *project*, *report*, *change*, and *expect*.)

3. Example Representational Decisions

Here we mention some of the issues we focussed on in developing Janus. The specification of WML appears in [15]; specifications for NIKL appear in [22, 26].

Few constants. One decision was to use as few constants as possible, deriving as many entities as possible using operators in the intensional logic. In this section we illustrate this point by showing how definitely referenced sets, information about kinds, indefinitely identified sets, and generic information can be stated by derivation from a single constant whose extension is the set of all individuals of a particular class.

Some of the expressive power of the hybrid is illustrated below as it pertains to minimizing the constants needed. From the constants BLACK-ENTITIES, GRAY-ENTITIES, CATS and MICE, the operators THE, POWER, KIND, and SAMPLE are used to derive the entities corresponding to definite sets, generic classes, and indefinite sets. In a semantic network without the hybrid, one might choose (or need) to represent each of our derived entities by a node in the network. Our use of the operator THE, and the operator POWER for definite plurals follows Scha [25]. The operators KIND and SAMPLE follow Carlson's analysis [10] of the semantics of bare plurals.

THE, as an operator, takes three arguments: a variable, a sort (unary predicate), and a proposition. Its denotation is the unique salient object in context such that it is in the sort and such that if the variable is bound to it, the proposition is true. POWER takes a sort as argument and produces the predicate corresponding to the power set of the set denoted by the sort. These operators are useful for representing definite plurals; *the black cats* would be represented as $(\text{THE } x (\text{POWER CATS}) (\text{BLACK-ENTITIES } x))$.

¹It is possible that one could extend NIKL semantics to allow for intensional arguments, but this has not been done.

²An analogy in more common terminology would be *His temperature dropped from 104 degrees to 99 degrees*.

SAMPLE takes the same arguments as THE, but indicates some set of entities satisfying the sort and proposition, not necessarily the largest set. KIND takes a sort as argument, and produces an individual representing the sort; its only use is for bare plurals that are surface subjects of a generic statement. If we are predicating something of a bare plural, KIND is used; for instance, *cats* as in *cats are ferocious* is represented as (KIND CATS). An indefinite set arising as a bare plural in a VP is represented using SAMPLE; for instance, *gray mice* as in *Cats eat gray mice* is represented as (SAMPLE x MICE (GRAY-ENTITIES x)).

The examples above demonstrate that an intensional logic enables derivation of many entities from fewer constants than would be needed in NIKL or other frame-based systems. The next example illustrates how the intensional logic lets us express some propositions that can be stated in many semantic network systems, but not in NIKL.

Generic assertions. Generic statements such as *Cats eat mice* are often encoded in a semantic network or frame system. This is not possible in the semantics of NIKL, but is possible in the hybrid. The structure in Figure 2 would not give the desired generic meaning, but rather would mean (ignoring time and world) that

$$(\forall x)((CATS\ x) \supset (\exists y)(\wedge (MICE\ y)(EAT\ x\ y))),$$

i.e., every cat eats some mouse.

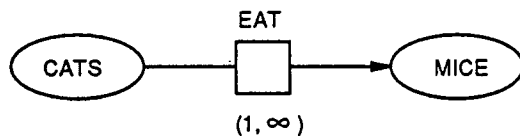


Figure 2: Illustration Distinguishing NIKL Networks from other Semantic Nets

Again, following Carlson's linguistic analysis [10], in the hybrid we would have a generic statement about the kind corresponding to cats, that these eat indefinitely specified sets of mice. GENERIC is an operator which produces a predicate on kinds, intuitively meaning that the resulting predicate is typically true of individuals of the kind that is its argument. Our formal representation (ignoring tense for simplicity) is

$$(GENERIC (LAMBDA (x) (EAT\ x (SAMPLE\ y\ MICE)))) (KIND\ CATS).$$

Next we illustrate a potential powerful feature of the hybrid which we have chosen not to exploit.

Derivable definitions. The hybrid gives a powerful means of defining lexical items. To define *pilot*, one wants a predicate defining the set of people that typically are the actors in a flight, i.e.,

$$(LAMBDA (x) \{ \wedge (PERSON\ x) (GENERIC (LAMBDA (x) (\exists y)(\wedge (FLYING-EVENT\ y) (ACTOR\ y\ x))) x \})$$

Though the hybrid gives us the representational capacity to make such definitions, we have chosen as part of our design not to use it. For to use it, would mean stepping outside of NIKL to specify constants, and therefore, that the reasoning algorithms based on taxonomic semantics would not be the simple, efficient strategies, but rather might require arbitrarily complex theorem proving for expressions in intensional logic.³

4. Use of the Taxonomy in Janus

By *domain model* we mean the set of axioms encoded in NIKL regarding the constants. The domain model serves several purposes in Janus. Of course, in defining the constants of our semantic representation language, it provides the constants that can appear in formulae that lexical items map to. For instance, *vessel* and *ship* map to VESSEL. In the example above regarding *pilot*, the constants were PERSON, FLYING-EVENT, and ACTOR; in the formula above stating that cats eat mice, the constants were EAT, MICE, and CATS.

In this section, we divide the discussion in three parts: current uses of the domain model in Janus; a plausible, but rejected use; and proposals for its use, but not yet implemented.

4.1. Current Uses

4.1.1. Selection Restrictions

The domain model provides the semantic classes (or sorts of a sorted logic) that form the primitives for selection restrictions. Its use for this purpose is neither novel nor surprising, merely illustrative. In the case of *deploy*, a MILITARY-UNIT can be the logical subject, and the object of a phrase marked by *to* must be a LOCATION. Almost all selection restrictions are based on the semantic class of the entities described by a noun phrase. That is, almost all may be checked by using taxonomic knowledge regarding constants.

A table of semantic classes for the operators discussed earlier is provided in Figure 3. Though the logical form for *the carriers*, *all carriers*, *some carriers*, *a carrier*, and *carriers* (both in the KIND and SAMPLE case) varies, the selection restriction must check the

³USC/ISI [19] has proposed a first-order formula defining the set of items that have ever been the actor in a flight. Their definition is solely within NIKL using the QUA link [14], which is exactly the set of fillers of a slot. While having ever flown could be a sense of *pilot*, it seems less useful than the sense of normally flying a plane.

NIKL network for consistency between the constant CARRIERS and the constraint of the selection restriction. To see this, consider the case of *command* (in the sense of a military command) which requires that its direct object in active clauses be a MILITARY-UNIT and that its surface subject in passive clauses be a MILITARY-UNIT, i.e., its logical object must be a MILITARY-UNIT. Suppose *USS Enterprise*, *carrier*, and *aircraft carrier* all have semantic class CARRIER. Since an ancestor of CARRIER in the taxonomy is MILITARY-UNIT, each of those phrases satisfy the aforementioned selection restriction on the verb *command*. Phrases whose class does not have MILITARY-UNIT as an ancestor or as a descendent⁴ will not satisfy the selection restriction. That is, definite evidence of consistency with the selection restriction is normally required.

<u>Expression</u>	<u>Semantic Class</u>
(THE x P (R x))	P
(POWER P)	P
(KIND P)	P
(SAMPLE x P (R x))	P
(LAMBDA x P (R x))	P

Figure 3: Relating Expressions to Classes⁵

There are three cases where more must be done. For pronouns, Janus saves selection restrictions that would apply to the pronoun's referent, later applying those constraints to eliminate candidate referents. Metonymy is an exception, discussed in Section 4.3.2. There are cases of selection restrictions requiring information additional to the semantic class, but these are checked against the type of the logical expression⁶ for a noun phrase, rather than its semantic class only. *Collide* requires a set of agents. The type of a plural, for instance, is (SET P), where P is its semantic class. The selection restriction on *collide* could be represented as (SET PHYSICAL-OBJECT).

4.1.2. Highly Polysemous Words

Have, *with*, and *of*, are highly polysemous. Some of their senses are very specific, frozen, and predictable, e.g., *to have a cold*; these senses may be itemized in the lexicon. However, other senses are vague, if considered in a domain-independent way; nevertheless, they must be resolved to precise meanings if accessing a data base, expert system, etc. *USS Frederick has a speed of 30 knots* has this flavor, for the general sense is associating an attribute with an entity.

⁴We check whether the constraint is a descendent of the class of the noun phrase to determine whether consistency is possible. For instance, if *decommission* requires a VESSEL as the object of the decommissioning, *those units* and *they* satisfy the selection constraint.

⁵The rules may need to be used recursively to get to a constant.

⁶Every expression in WML has a type.

To handle such cases, we look for a relation R in the domain model which could be the domain-dependent interpretation. If *A has B*, *the B of A*, or *A with B* are input, the semantic interpreter looks for a role R from the class associated with A to the class associated with B. If no such role exists, the search is for a role relating the nearest ancestor of the class of A to any ancestor of the class of B. The implicit assumption is that items structured closely together in the domain model can be related with such vague words, and that items that can be related via such vague words will naturally have been organized closely together in the domain model.

While describing the procedure as a search, in fact, an explicit run-time search may not be necessary. All SUPERCs (ancestors) of a concept are compiled and stored when the taxonomy is loaded. All roles from one concept to another are also pre-compiled and stored, maintaining the distinction between roles that are explicit locally versus those that are compiled. Furthermore, the ancestors and role relations are indexed. One need only walk up the chain of ancestors if no locally defined role relates the two concepts, but some inherited (not locally defined) role does; then one walks up the ancestor chain(s) only to find the closest applicable role. Thus, in many cases, "semantic reasoning" is reduced to efficient table lookup.

4.1.3. Relation to Underlying System

Adopting WML offers the potential of simplifying the mapping from surface form to semantic representation, although it does increase the complexity of mapping from WML to executable code, such as SQL or expert system function calls. The mapping from intentional logic to executable code is beyond the scope of this paper; our first implementation was reported in [30]; the current implementation will be described elsewhere.

This process makes use of a model of underlying system capabilities in which each element relates a set of domain model constants to a method for accessing the related information in the database, expert system, simulation program, etc. For example, the constant HARPOON-CAPABLE, which defines a set of vessels equipped with harpoon missiles, is associated with an underlying system model element which states how to select the subset of exactly those vessels. In a Navy relational data base that we have dealt with, the relevant code selects just those records of a table of unit characteristics with a "Y" in the HARP field.

4.1.4. Knowledge Acquisition

We have developed two complementary tools to greatly increase our productivity in porting BBN's Janus NL understanding and generation system to new domains. IRACQ [3] supports learning lexical semantics from examples with only one unknown

word. IRACQ is used for acquiring the diverse, complex patterns of syntax and semantics arising from verbs, by providing examples of the verb's usage. Since IRACQ assumes that a large vocabulary is available for use in the training examples, a way to rapidly infer the knowledge bases for the overwhelming majority of words is an invaluable complement.

KNACQ [33] serves that purpose. The domain model is used to organize, guide, and assist in acquiring the syntax and semantics of domain-specific vocabulary. Using the browsing facilities, graphical views, and consistency checker of KREME [1] on NIKL taxonomies, one may select any concept or role for knowledge acquisition. KNACQ presents the user with a few questions and menus to elicit the English expressions used to refer to that concept or role.

To illustrate the kinds of information that must be acquired consider the examples in Figure 4.

The vessel speed of Vinson
The vessels with speed above 20 knots
The vessel's speed is 5 knots
Vinson has speed less than 20 knots
Its speed
Which vessels have a CROVL of C3?
Which vessels are deployed C3?

Figure 4: Examples for Knowledge Acquisition

To handle these one would have to acquire information on lexical syntax, lexical semantics, and mapping to expert system structure for all words not in the domain-independent dictionary. For purposes of this exposition, assume that the words, *vessel*, *speed*, *Vinson*, *CROVL*, *C3*, and *deploy* are to be defined. A *vessel has a speed of 20 knots* or *a vessel's speed is 20 knots* would be understood from domain-independent semantic rules regarding *have* and *be*, once lexical information for *vessel* and *speed* is acquired. In acquiring the definitions of *vessel* and *speed*, the system should infer interpretations for phrases such as *the speed of a vessel*, *the vessel's speed*, and *the vessel speed*.

Given the current implementation, the required knowledge for the words *vessel*, *speed*, and *CROVL* is most efficiently acquired using KNACQ; names of instances of classes, such as *Vinson* and *C3* are automatically inferred from instances; and knowledge about *deploy* and its derivatives would be acquired via IRACQ.

To illustrate this acquisition centered around the domain model, consider acquisition centered around roles. *Attributes* are binary relations on classes that can be phrased as *the <relation> of a <class>*. For instance, suppose CURRENT-SPEED is a binary relation relating vessels to SPEED, a subclass of ONE-D-MEASUREMENT. An attribute treatment is the most appropriate, for *the speed of a vessel* makes perfect sense. KNACQ asks the user for one or more English phrases associated with this functional role;

the user response in this case is *speed*. That answer is sufficient to enable the system to understand the kernel noun-phrases listed in Figure 5. Since ONE-D-MEASUREMENT is the range of the relation, the software knows that statistical operations such as average and maximum apply to *speed*. The lexical information inferred is used compositionally with the syntactic rules, domain independent semantic rules, and other lexical semantic rules. Therefore, the generative capacity of the lexical semantic and syntactic information is linguistically very great, as one would require. A small subset of the examples illustrating this without introducing new domain specific lexical items appears in Figure 5.

KERNEL NOUN PHRASES

the speed of a vessel
the vessel's speed
the vessel speed

RESULTS from COMPOSITIONALITY

The vessel speed of Vinson
Vinson has speed 1
The vessels with a speed of 20 knots
The vessel's speed is 5 knots
Vinson has speed less than 20 knots
Their greatest speed
Its speed
Which vessels have speed above 20 knots
Which vessels have speeds
Eisenhower has Vinson's speed
Carriers with speed 20 knots
Their average speeds

Figure 5: Attribute Examples

Some lexicalizations of roles do not fall within the attribute category. For these, a more general class of regularities is captured by the notion of caseframe rules. Suppose we have a role UNIT-OF, relating CASREP and MILITARY-UNIT. KNACQ asks the user which subset of the following six patterns in Figure 6 are appropriate plus the prepositions that are appropriate.

1. <CASREP> is <PREP> <MILITARY-UNIT>
2. <CASREP> <PREP> <MILITARY-UNIT>
3. <MILITARY-UNIT> <CASREP>
4. <MILITARY-UNIT> is <PREP> <CASREP>
5. <MILITARY-UNIT> <PREP> <CASREP>
6. <CASREP> <MILITARY-UNIT>

Figure 6: Patterns for the Caseframe Rules

For this example, the user would select patterns (1),

(2), and (3) and select *for*, *on*, and *of* as prepositions.⁷

The information acquired through KNACQ is used both by the understanding components and by BBN's Spokesman generation components for paraphrasing, for providing clarification responses, and for answers in English. Mapping from the WML structures to lexical items is accomplished using rules acquired with KNACQ, as well as handcrafted mapping rules for lexical items not directly associated with concepts or roles.

4.2. Where an Alternative Mechanism was Selected

Though the domain model is central to the semantic processing of Janus, we have not used it in all possible ways, but only where there seems to be clear benefit.

In telegraphic language, omitted prepositions, as in *List the creation date file B*, may arise. Alternatively, if the NLP system is part of a speech understanding system, prepositions are among the most difficult words to recognize reliably. Omitted prepositions could be treated with the same heuristic as implemented for interpreting the meaning of *have*, *with*, and *of*. However, we have chosen a different inference technique for omitted prepositions.

Though one could represent selection restrictions directly in a taxonomy (as reported in [7, 29]), selection restrictions in Janus are stored separately, indexed by the semantic class of the head word. We believe it more likely that Janus will have the selectional pattern involving the omitted preposition, than that the omitted preposition corresponds to a usage unknown to Janus and inferable from the domain model relations. Consequently, Janus applies the selection restrictions corresponding to all senses of the known head, to find what senses are consistent with the proposed phrase and with what prepositions. In practice, this gives rise to far fewer possibilities than considering all relations possible whether or not they can be expressed with a preposition.

4.3. Proposals not yet Implemented (Possible Future Directions)

In this section, we speculate regarding some possible future work based on further exploiting the domain model and hybrid representation system described in this paper.

⁷Normally, if pattern (1) is valid, pattern (2) will be as well and vice versa. Similarly, if pattern (4) is valid, pattern (5) will normally be also. As a result, the menu items are coupled by default (selecting (1) automatically selects (2) and vice versa), but this default may be simply overridden by selecting either and then deselecting the other. The most frequent examples where one does not have the coupling of those patterns is the preposition *of*.

4.3.1. An Approach to Bridging

It has long been observed [11] that mention of one class of entities in a communication can bring into the foreground other classes of entities which can be referred to though not explicitly introduced. The process of inferring the referent when such a reference occurs has been called *bridging* [12]. Some examples, taken from [12], appear below, where the reference requiring bridging is underlined.

1. *I looked into the room. The ceiling was very high.*
2. *I walked into the room. The chandeliers sparkled brightly.*
3. *I went shopping yesterday. The time I started was 3 PM.*

We believe a taxonomic domain model provides the basis for an efficient algorithm for a broad class of examples of bridging, though we do not believe that it will cover all cases. If A is the class of a discourse entity arising from previous utterances, then any entity of class B, such that the NIKL domain model has a role from A to B (or from B to A) can be referred to by a definite NP. This has not yet been integrated into the Janus model of reference processing [4].

4.3.2. Metonymy

Unstated relations in a communication must be inferred for full understanding of nominal compounds and metonymy. Those that can be anticipated can be built into the lexicon; the challenge is to deal with those that are novel to Janus. Finding the omitted relation in novel nominal compounds using a taxonomy has been explored and reported elsewhere [13].

We propose treating many novel cases of metonymy in the following way:

1. Where patterns of metonymy can be identified, such as using a description of a part to refer to the whole (and other patterns identified in [17]), pre-compile chains of relations between classes in the domain model, e.g., (PART-OF A B) where A and B are concepts.
2. In processing an input, when a selection restriction on an NP fails, record the failed restriction with the partial interpretation for possible future processing, after all attempts at a literal interpretation of the input have failed.
3. If no literal interpretation of the input can be found, look among the precompiled relations of step 1 above for any class that could be so related to the class of the NP that appears.
4. If a relation is applicable, attempt to resume interpretation assuming the referent of the NP is in the related class.

This has not been implemented, but offers an efficient

alternative to the abductive theorem-proving approach described in [16].

5. Top-Level Abstractions in the NIKL Taxonomy

WML and NIKL together provide a framework for representation. The highest concepts and relations in the NIKL network provide a representational style in which more concrete constants must fit. The first abstraction structure used in Janus was the USC/ISI "upper structure" [19]. Because it seemed tied to systemic linguistics in critical ways, rather than to a more general ontological style, we have replaced it with another domain-independent set of concepts and roles. For any application domain, all domain-dependent constants must fit underneath the domain-independent structure. The domain-independent taxonomy consists of 70 concepts and 24 roles currently, but certainly could be further expanded as one attempts to further axiomatize and model notions useful in a broad class of application domains.

During the evolution of Janus, we explored whether the domain-independent taxonomy could be greatly expanded by a broad set of primitives used in the Longman Dictionary of Contemporary English [18] (LDOCE) to define domain-independent constants. LDOCE defines approximately 56,000 words in terms of a base vocabulary of roughly 2,000 items.⁸ We estimate that about 20,000 concepts and roles should be defined corresponding to the 2,000 multi-way ambiguous words in the base vocabulary. The appeal, of course, is that if these basic notions were sufficient to define 56,000 words, they are generally applicable, providing a candidate for general-purpose primitives.

The course of action we followed was to build a taxonomy for all of the definitions of approximately 200 items from the base vocabulary *using the definitions of those vocabulary items themselves in the dictionary*. In this attempt, we encountered the following difficulties:

- Definitions of the base vocabulary often involved circularity.
- Definitions included assertional information and/or knowledge appropriate in defeasible reasoning, which are not fully supported by NIKL. For example, the first definition of *cat* is "a small four-legged animal with soft fur and sharp claws, often kept as a pet or for catching mice or rats."
- Multiple views and/or vague definitions and usage arose in LDOCE. For instance, the

⁸Though the authors of LDOCE definitions try to stay within the base vocabulary, exceptions do arise such as diagrams and proper nouns, e.g., *Catholic Church*.

second definition of *cat* (p. 150) is "an animal *related to this* such as the lion or tiger" (italics added). Such a vague definition helped us little in axiomatizing the notion.

Thus, we decided that hand-crafted abstractions would be needed to axiomatize by hand the LDOCE base vocabulary if general-purpose primitives were to result. On the other hand, concrete concepts corresponding to a lower level of abstraction seem obtainable from LDOCE. In particular the LDOCE definitions of units of measurement for the avoirdupois and metric systems were very useful. A more detailed analysis of our experience is presented in [23].

6. Related Work

Several hybrid representation schemes have been created, although only ours seems to have explored a hybrid of intensional logic with an axiomatizable frame system. The most directly related efforts are the following:

- KL-TWO [31], which marries a frame system (NIKL) with propositional logic (RUP [20]). Limited inference in propositional logic is the goal of KL-TWO. Limited aspects of universal quantification are achieved via allowing demons in the inference process. KL-TWO and its classification algorithm [27] are at the heart of the lexicalization process of the text generator Penman [28].
- KRYPTON [9], which marries a frame system with first-order logic. The frame system is designed to be less expressive than NIKL to allow rapid checking for disjointness of two class concepts in order to support efficient resolution theorem proving. KRYPTON has not as yet been used in any natural language processor.

7. Conclusions

Our conclusions regarding the hybrid representation approach of intensional logic plus NIKL-based axioms to define constants are based on three kinds of efforts:

- Bringing Janus up on two large expert system and data base applications within DARPA's Battle Management Programs. The combined lexicon in the effort is approximately 7,000 words (not counting morphological variations).
- The efforts synopsised in Section 5 towards general purpose domain notions.
- Experience in developing IRACQ and KNACQ, acquisition tools integrated with the domain model acquisition and maintenance facility KREME.

First, a taxonomic language with a formal semantics can supplement a higher order logic in support of efficient, limited inferences needed in a natural language processor. Based on our experience and that of others, the axioms and limited inference algorithms can be used for classes of anaphora resolution, interpretation of *have*, *with*, and *of*, finding omitted relations in novel nominal compounds, applying selection restrictions, and mapping from the semantic representation of the input to code to carry out the user's request.

Second, an intensional logic can supplement a taxonomic language in trying to define word senses formally. Our effort with LDOCE definitions showed how little support is provided for defining word senses in a taxonomic language. A positive contribution of intensional logic is the ability to distinguish universal statements from generic ones from existential ones; definite sets from unspecified ones; and necessary and sufficient information from assertional information, allowing for a representation closer to the semantics of English.

Third, the hybridization of axioms for taxonomic knowledge with an intensional logic does not allow us to represent all that we would like to, but does provide a very effective engineering approach. Out of 7,000 lexical entries (not counting morphological variations), only 0.1% represented concepts inappropriate for the formal semantics of NIKL.

The ability to pre-compile pre-specified, inferential chains, to index them via concept name and role name, and to employ taxonomic inheritance for organizing knowledge were critical in selecting taxonomic representation to supplement WML. These techniques of pre-compiling pre-specified inferential chains and of indexing them should also be applicable to other knowledge representations than taxonomies.

At a later date, we hope to quantify the effectiveness of the semantic heuristics described in this paper.

Acknowledgements

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contracts N00014-85-C-0079 and N00014-85-C-0016. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

This brief report represents a total team effort. Significant contributions were made by Damaris Ayuso, Rusty Bobrow, Ira Haimowitz, Erhard Hinrichs, Thomas Reinhardt, Remko Scha, David Stallard, and Cynthia Whipple. We also wish to acknowledge many

discussions with William Mann and Norman Sondheimer in the early phases of the project.

References

1. Abrett, G. and Burstein, M. "The KREME Knowledge Editing Environment". *Int. J. Man-Machine Studies* 27 (1987), 103-126.
2. Ayuso Planes, D. The Logical Interpretation of Noun Compounds. Master Th., Massachusetts Institute of Technology, June 1985.
3. Ayuso, D.M., Shaked, V., and Weischedel, R.M. An Environment for Acquiring Semantic Information. Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics, ACL, 1987, pp. 32-40.
4. Ayuso, Damaris. Discourse Entities in Janus. Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics, 1989.
5. BBN Systems and Technologies Corp. A Guide to IRUS-II Application Development in the FCCBMP. BBN Report 6859, BBN Systems and Technologies Corp., Cambridge, MA, 1988.
6. Bobrow, R. and Webber, B. PSI-KLONE: Parsing and Semantic Interpretation in the BBN Natural Language Understanding System. Proceedings of the 1980 Conference of the Canadian Society for Computational Studies of Intelligence, CSCSI/SCEIO, May, 1980.
7. Bobrow, R. and Webber, B. Knowledge Representation for Syntactic/Semantic Processing. Proceedings of the National Conference on Artificial Intelligence, AAAI, August, 1980.
8. Brachman, R.J. and Schmolze, J.G. "An Overview of the KL-ONE Knowledge Representation System". *Cognitive Science* 9, 2 (April 1985).
9. Brachman, R.J., Gilbert, V.P., and Levesque, H.J. An Essential Hybrid Reasoning System: Knowledge and Symbol Level Accounts of Krypton. Proceedings of IJCAI85, International Joint Conferences on Artificial Intelligence, Inc., Los Angeles, CA, August, 1985, pp. 532-539.
10. Carlson, G.. *Reference to Kinds in English*. Garland Press, New York, 1979.
11. Chafe, W. Discourse Structure and Human Knowledge. In *Language Comprehension and the Acquisition of Knowledge*, Winston and Sons, Washington, 1972.
12. Clark, H.H. Bridging. Theoretical Issues in Natural Language Processing, 1975, pp. 169-174.
13. Finin, T.W. The Semantic Interpretation of Nominal Compounds. Proceedings of The First Annual National Conference on Artificial Intelligence,

The American Association for Artificial Intelligence, August, 1980, pp. 310-312.

14. Freeman, M. The QUA Link. Proceedings of the 1981 KL-ONE Workshop, Bolt Beranek and Newman Inc., 1982, pp. 55-65.

15. Hinrichs, E.W., Ayuso, D.M., and Scha, R. The Syntax and Semantics of the JANUS Semantic Interpretation Language. In *Research and Development in Natural Language Understanding as Part of the Strategic Computing Program, Annual Technical Report December 1985 - December 1986*, BBN Laboratories, Report No. 6522, 1987, pp. 27-31.

16. Hobbs, et. al. Interpretation as Abduction. Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics, 1988, pp. 95-103.

17. Lakoff, G. and Johnson, M.. *Metaphors We Live By*. The University of Chicago Press, Chicago, 1980.

18. *Longman Dictionary of Contemporary English*. Essex, England, 1987.

19. Mann, W.C., Arens, Y., Matthiessen, C., Naberschnig, S., and Sondheimer, N.K. Janus Abstraction Structure -- Draft 2. USC/Information Sciences Institute, 1985.

20. David A. McAllester. Reasoning Utility Package User's Manual. AI Memo 667, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, April, 1982.

21. Montague, Richard. The Proper Treatment of Quantification in Ordinary English. In *Approaches to Natural Language*, J. Hintikka, J. Moravcsik and P. Suppes, Eds., Reidel, Dordrecht, 1973, pp. 221-242.

22. Moser, M.G. An Overview of NIKL, the New Implementation of KL-ONE. In *Research in Knowledge Representation for Natural Language Understanding - Annual Report, 1 September 1982 - 31 August 1983*, Sidner, C. L., et al., Eds., BBN Laboratories Report No. 5421, 1983, pp. 7-26.

23. Reinhardt, T. and Whipple, C. Summary of Conclusions from the Longman's Taxonomy Experiment. In Goodman, B., Ed., , BBN Systems and Technologies Corporation, Cambridge, MA, 1988, pp. .

24. Rich, C. Knowledge Representation languages and the Predicate Calculus: How to Have Your Cake and Eat It Too. Proceedings of the Second National Conference on Artificial Intelligence, AAAI, August, 1982, pp. 193-196.

25. Scha, R. and Stallard, D. Multi-level Plurals and Distributivity. 26th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, June, 1988, pp. 17-24.

26. Schmolze, J. G., and Israel, D. J. KL-ONE: Semantics and Classification. In *Research in Knowledge Representation for Natural Language Understanding - Annual Report, 1 September 1982 - 31 August 1983*, Sidner, C.L., et al., Eds., BBN Laboratories Report No. 5421, 1983, pp. 27-39.

27. Schmolze, J.G., Lipkis, T.A. Classification in the KL-ONE Knowledge Representation System. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 1983.

28. Sondheimer, N. K. and Nebel, B. A Logical-form and Knowledge-base Design for Natural Language Generation. Proceedings AAAI-86 Fifth National Conference on Artificial Intelligence, The American Association for Artificial Intelligence, Los Altos, CA, Aug, 1986, pp. 612-618.

29. Sondheimer, N.K., Weischedel, R.M., and Bobrow, R.J. Semantic Interpretation Using KL-ONE. Proceedings of COLING-84 and the 22nd Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Stanford, CA, July, 1984, pp. 101-107.

30. Stallard, David. Answering Questions Posed in an Intensional Logic: A Multilevel Semantics Approach. In *Research and Development in Natural Language Understanding as Part of the Strategic Computing Program*, R. Weischedel, D.Ayuso, A. Haas, E. Hinrichs, R. Scha, V. Shaked, D. Stallard, Eds., BBN Laboratories, Cambridge, Mass., 1987, ch. 4, pp. 35-47. Report No. 6522.

31. Vilain, M. The Restricted Language Architecture of a Hybrid Representation System. Proceedings of IJCAI85, International Joint Conferences on Artificial Intelligence, Inc., Los Angeles, CA, August, 1985, pp. 547-551.

32. Weischedel, R.M. "Knowledge Representation and Natural Language Processing". *Proceedings of the IEEE 74, 7* (July 1986), 905-920.

33. Weischedel, R.M., Bobrow, R., Ayuso, D.M., and Ramshaw, L. Portability in the Janus Natural Language Interface. Notebook of Speech and Natural Language Workshop, 1989. To be reprinted by Morgan Kaufmann Publishers.