THE INTERPRETATION OF RELATIONAL NOUNS

Jos de Bruin* and Remko Scha

BBN Laboratories
10 Moulton Street
Cambridge, MA 02238, USA

ABSTRACT

This paper[1] decribes a computational treatment of the semantics of *relational nouns.* It covers relational nouns such as *"sister"* and *"commander"*, and focuses especially on a particular subcategory of them, called *function nouns ("speed", "distance", "rating").* Relational nouns are usually viewed as either requiring non-compositional semantic interpretation, or causing an undesirable proliferation of syntactic rules. In contrast to this, we present a treatment which is both syntactically uniform and semantically compositional. The core ideas of this treatment are: (1) The recognition of different levels of semantic analysis; in particular, the distinction between an *English-oriented* and a *domain-oriented* level of meaning representation. (2) The analysis of relational nouns as denoting relation-extensions.

The paper shows how this approach handles a variety of linguistic constructions involving relational nouns. The treatment presented here has been implemented in BBN's Spoken Language System, an experimental spoken language interface to a database/graphics system.

1 RELATIONAL NOUNS AND THEIR DENOTATIONS

When Jean Piaget faced his nine year old subject Hal with the question *"What's a brother?"*, the answer was: *"When there's a boy and another boy, when there are two of them."* And, with a greater degree of formal precision, ten year old Bern replied to the same question: *"A brother is a relation, one brother to another."* [2] [8] What these children are beginning to be able to articulate is that there is something wrong with the experimenter's question as it is posed: it talks about "brothers" as if they constituted a natural kind, as if there is a way of looking at an individual to find out whether he is a brother. But *"brother"* is normally not used that way -- a property which it shares with words like *"co-author"*, *"commander"*, *"speed"*, *"distance"*, and *"rating"*.

Nouns of this sort are called *relational nouns.* As we shall see in a moment, their semantic properties differ significantly from those of other nouns, so that the standard treatments of nominal semantics don't apply to them. The problem of the semantic interpretation of relational nouns constitutes the topic of this paper. We shall argue that this problem is indeed a *semantic* one, and should preferably *not* be treated in the syntax. The semantic treatment that we present uses a *multilevel semantics* framework, and is based on the idea of assigning *relation extensions* as denotations to relational nouns.

Relational nouns are semantically unsaturated. They are normally used in combination with an implicit or explicit *argument: "John's brother".* The argument of a relational noun, if overtly realized in the sentence, is connected to the noun by means of a relation-denoting lexical element: the verb "have" or one of its semantic equivalents (the genitive and the prepositions *"of"* and *"with"*): *"John has a sister", "John's sister", "a sister of John's", "a boy with a sister".* It has been noted that these lexical items interact differently with relational nouns than they do with other nouns. [7] Compare, for instance, the noun *"car"* in (1)/(1abcd) with the relational noun *"brother"* in the parallel sentences (2)/(2abcd): (1) entails (1abcd), but the corresponding (2) does not entail (2abcd).[2]

(1) "John's cars are wrecks."
(1a) "Some wrecks of John's are cars."
(1b) "Some wrecks are John's."
(1c) "Some cars are John's."
(1d) "John has wrecks."

(2) "John's brothers are punks."
#(2a) "Some punks of John's are brothers."
#(2b) "Some punks are John's."
#(2c) "Some brothers are John's."
#(2d) "John has punks."

A particular subcategory of the relational nouns, that we shall consider in some detail, is constituted by the *function nouns*; they are semantically distinct in that for every argument they refer to exactly one entity, which is an element of a linear ordering: a num-

[1] This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. N00014-87-C-0085.

*Current address: Cartesian Products BV, WG Plein 316, 1054 SG Amsterdam, The Netherlands.

[2] We refrain from saying that (2abcd) are ungrammatical. Because of the semantic open-endedness of *"have"* and the genitive, these sentences can certainly be wellformed and meaningful, if uttered in an appropriate context. The issue at stake is that the interpretation which is the salient one for the genitive in (2) is not available for the corresponding elements in (2abcd). Sentences displaying this property have been marked with the #-sign (rather than the ungrammaticality-asterisk) in this paper.

ber, an amount, or a grade. Examples are "length", "speed", "distance", "rating". Function nouns can be used in constructions which exclude other nouns, relational as well as non-relational. Compare, for instance:

(3) "The USS Frederick has a speed of 15 knots."
#(3a) "John has a car of this wreck."
#(3b) "John has a brother of Peter."

The examples above show that there are significant semantic differences between phrases connecting relational nouns to their functions/values, and the corresponding, similarly structured phrases built around other nouns. This suggests that the standard treatment of ordinary nouns cannot be applied directly to relational nouns and yield correct results. To conclude this introductory section, we investigate this issue in a little more detail.

Assume a semantic framework with the following, not very unusual, features. Nouns are analyzed as set-denoting constants; concomitantly, adjectives are analyzed as one-place predicates, prepositions as two-place predicates, verbs as n-place predicates. Plural noun phrases with "the" or a possessive denote sets which have the same semantic type as the noun around which they are built: *"John's cars"* denotes a particular set of cars. In this approach, the representation of the noun phrase *"Peter's sisters"* would be:

$\{x \in SISTERS \mid POSSESS(PETER,x)\}$,

where *SISTERS* denotes the set of persons who are a sister, and *POSSESS* represents the possessive relation indicated by the genitive construction.

Now this expression does not have the right properties. It lacks necessary information: the predicate $(\lambda\ x: POSSESS(PETER,x))$ applies to elements of the *extension* of *SISTERS*; it cannot take into account *how* this extension was defined. For instance, if in a particular world the set of sisters is co-extensional with the set of co-authors, the approach just outlined would incorrectly assign to "*Peter's sisters*" the same denotation as to "*Peter's co-authors*".

It is clear what the source of the problem is: the semantic representations for relational nouns considered above denote simple sets of individuals, and do not contain any information about the *relation* involved. To salvage a uniform compositional treatment, a richer representation is needed. One might think of invoking Montague's *individual concepts* [3] [6], or enriching one's ontology with *qua*-individuals (distinguishing between Mary *qua* sister and Mary *qua* aunt) [4]. In section 4 we will present our solution to this problem. First, we discuss why we didn't choose for a more syntactically oriented approach.

2 AGAINST SYNTACTIC TREATMENTS

Often, the complexities mentioned above are taken to require a distinction between intransitive common nouns and transitive common nouns in the syntax, with a concommittant proliferation of syntactic rules. Instead, we have chosen to extend a treatment of "ordinary" nouns only at the semantic processing stage. We shall now indicate some of the reasons for this choice.

Relational nouns are semantically dependent on an argument. In this respect, they are more reminiscent of verbs than of standard nouns like *"boy"* or *"chair"*. Most verbs of English have one or more argument places that *must* be filled for the verb to be used in a syntactically/semantically felicitous way; this property of verbs is probably an important reason for the persisting tendency to analyze them as n-place predicates rather than sets of situations. The semantic similarity between relational nouns and verbs has given rise to treatments which model the syntactic treatment of nouns on the treatment of verbs: one introduces lexical subcategories for nouns which indicate how many arguments they take and how these arguments are marked; the syntactic rules combine N-bars or noun-phrases with genitive phrases and preposition-phrases, taking these subcategorizations into account. [15] We will now argue, however, that from a syntactic point of view such a move is unattractive.

Syntactically, relational nouns do not behave very differently from "ordinary" nouns. They combine with adjectives, determiners, preposition phrases and relative clauses to form noun phrases with a standard X-bar structure; and the noun phrases thus constituted can participate in all sentence-level structures that other noun phrases partake in.

Also, no nouns have syntactic properties that would be analogous to the sentence-level phenomenon of a verb *obligatorily* taking one or more arguments. The overt realization of the arguments of a "transitive noun" is always optional.

Finally, we may note that relational nouns can be connected to their arguments/values by a variety of verbs and prepositions, which constitute a semantic complex that is also used, with exactly the same structure but with a different meaning, to operate on non-relational nouns. Compare, for instance:

"The Chevrolet of Dr. Johnson"
/ *"The speed of Frederick"*
"Dr. Johnson's Chevrolet"
/ *"Frederick's speed"*
"The Chevrolet that Dr. Johnson has"
/ *"The speed that Frederick has"*
"Dr. Johnson acquired a rusty Chevrolet"
/ *"Frederick acquired a formidable speed"*
"A philosopher with a rusty Chevrolet"
/ *"A ship with a formidable speed"*

The same set of terms is used in English for the ownership relation, for the part-whole relation, and for the relation between a function and its argument. These terms (like *"of"*, *"have"* and *"with"*) are highly polysemous, and any language processing system must encompass mechanisms for disambiguating their intended meaning in any particular utterance.

To summarize: relational nouns do not distinguish themselves syntactically from other nouns, and they mark their function-argument structures by means of polysemous descriptive terms. We therefore conclude that it would be theoretically elegant as well as computationally effective to treat relational and non-relational nouns identically at the syntactic level, and to account for the semantics of relational noun constructions by exploiting independently motivated disambiguation mechanisms. The remainder of this paper describes such a treatment.

First, Section 3 discusses the *multilevel semantics* architecture which constitutes the framework for our approach. Section 4 presents our answer to a basic question about relational nouns: what should their denotations be? This section then goes on to describe the semantic transformations which derive the desired analyses of constructions involving relational nouns. Section 5 briefly discusses the interface with a Discourse Model, which is necessary to recover arguments of a relation that are left implicit in an utterance. Section 6 shows that our treatment is useful for the purpose of response-formulation in question-answering.

3 MULTILEVEL SEMANTICS.

Our approach to the problem of relational nouns is based on the idea of *multilevel semantics*: the distinction between different levels of semantic analysis. [1] [10] In this approach, interpreting a natural language sentence is a multi-stage process, which starts out with a high-level meaning representation which reflects the semantic structure of the English sentence rather directly, and then applies translation rules which specify how the English-oriented semantic primitives relate to the ones that are used at deeper levels of analysis.

At every level of analysis, the meaning of an input utterance is represented as an expression of a logical language.[3] The languages used at the various levels of analysis differ in that at every level the descriptive constants are chosen so as to correspond to the semantic primitives which are assumed at that level.

At the highest semantic level, the meaning of an input utterance is represented as an expression of the *English-oriented Formal Language* (EFL). The constants of EFL correspond to the descriptive terms of English. A feature of EFL which is both unusual and important, is the fact that descriptive constants are allowed to be *ambiguous*. Within each syntactic category, every word is represented in EFL by a single descriptive constant, no matter how many senses the word may have. An EFL expression can thus be seen as an expression *schema*, where every constant can be expanded out in a possibly large number of different ways. (See [5] for details on the model theory of such a logic.)

The ambiguity of EFL follows from its domain-independence. All descriptive words of a language are polysemous, and only when used in the context of a particular subject domain do they acquire a single precise meaning -- a meaning which cannot be articulated independently of that subject domain. Even within one subject domain, many words have a range of different meanings. Joint representations for such sets of possible expansions are computationally advantageous; and when the range of possibilities is defined in an open-ended way, they are even necessary. Such cases occur when we attempt to account for the interpretation of metonymy, metaphor and nominal compounds [12], or the interpretation of multilevel plural noun phrases [11].

The logical language used at the domain-dependent level of representation is called the *World Model Language* (WML). This is an unambiguous language, with an ordinary model-theoretic interpretation. Its constants are chosen to correspond to the concepts which constitute the domain of discourse.[4]

We can illustrate the distinction between EFL and WML by means of an example involving relational nouns. Compare (4) and (5) below. Sentence (4) will usually be translated into something like (4a):[5]

(4) "John has a house in Hawaii."

(4a) $\exists\ h \in \{h \in HOUSES \mid IN(h,HAWAII)\}:$
$$HAVE(JOHN,h)$$

Now consider (5) instead; a single-level architecture would have to analyse this sentence as (5b) rather than (5a), since (5b) is the representation one would prefer to end up with.

(5) "Frederick has a speed of 15 knots."

(5a) $\exists\ c \in \{c \in SPEEDS$
$$\mid OF(c, \mathrm{amount}(15, KNOTS))\}:$$
$$HAVE(FREDERICK,c)$$

[3] BBN's Spoken Language System uses a higher-order intensional logic based on Church's lambda-calculus, combining features from PHLIQA's logical language [5] with Montague's Intensional Logic [6].

[4] To provide a smooth interface with underlying application systems, there is a third level of semantic interpretation. The language used at this level is called the *Data Base Language* (DBL). Its constants stand for the files and attributes of the database to be accessed, and the available graphics system operations and their parameters.

[5] Accommodating discourse anaphora may motivate a different treatment of the indefinite noun phrase, representing its semantics by a Skolem-constant or a similar device, rather than by the traditional existential quantifier. For the present discussion we may ignore this issue.

(5b) F-SPEED(FREDERICK) = amount*(15, KNOTS)*

In a multilevel semantics architecture, however, one would prefer to maintain a completely uniform first stage in the semantic interpretation process, where (5) would be treated exactly as (4), and therefore be analyzed as (5a). By applying appropriate EFL-to-WML translation rules, the EFL expression (5a) would then be transformed into the WML expression (5b). Taking natural language at semantic face value thus simplifies the process of creating an initial meaning representation. The remaining question then is, whether one can in fact write EFL-to-WML translation rules which yield the desired results. This is the question we will come back to in section 4. In the remainder of the present section, we first give some more detail on the general properties of the translation rules and the logical languages.

The interpretive rules which map syntactic structures onto EFL expressions are *compositional*, i.e., they correspond in a direct way to the syntactic rules which define the legal input strings. There is a methodological reason for this emphasis on compositionality: it makes it possible to guarantee that all possible combinations between syntactic rules are in fact covered by the interpretive rules, and to minimize surprises about the way the rules interact. Similar considerations apply when we think about the definition of the EFL-to-WML translation: we wish to guarantee that the WML translations of every EFL expression are defined in an effectively computable way, and that the different rules which together specify the translation interact in a predictable fashion. This is achieved by specifying the EFL-to-WML translation using strictly *local* rules: rules operating only on *constants*, which specify for every EFL constant the WML expressions that it translates into.

Translation by means of local rules, which expand constants into complex expressions, tends to create fairly large and complicated formulas. The result of the EFL-to-WML translation is therefore processed by a logical simplification module; this keeps formulas from becoming too unwieldy to handle and impossible to evaluate.

Local rules by themselves do not specify what combinations between them will lead to legitimate results. Since the rules can be applied independently of each other, we need a separate mechanism for checking the meaningfulness of their combined operation. This mechanism is the *semantic type system*.

EFL, WML and DBL are *typed* languages. This means that for every expression of these languages, a semantic type is defined. The denotation of an expression is guaranteed to be a member of the set denoted by its type. In WML, for instance, *FREDERICK* has the type *SHIPS* which denotes the set of all ships; *GUAM* and *INDIAN-OCEAN* have the type *LOCATIONS* which denotes the set of all locations; *CARRIERS* and *SHIPS* both have the type *SETS(SHIPS)* which denotes the powerset of the set of all ships; *F-SPEED* has the type

FUNCTIONS(U(SHIPS,PLANES, LAND-VEHICLES), AMOUNTS(SPEED-UNITS)),

which denotes the set of functions whose domain is the union of the sets of ships, of planes and of land vehicles, and whose range is the set of amount-expressions whose units are members of the set of speed-units.

Given the types of the constants occurring in it, the type of a complex expression is determined by formal rules. For instance, the expression *F-SPEED(FREDERICK)* would have the type *AMOUNTS(SPEED-UNITS)*. The rules which define the types of complex expressions also define when an expression does *not* have a legitimate type, and is therefore not considered to be a bona fide member of the language. For instance, *F-SPEED(GUAM)* does not have a legitimate type, because the type-computation rule for function-application expressions requires that the type of the argument not be disjoint with the domain of the function.

The semantic type constraints make it possible to express the possible interpretations of ambiguous EFL constants by means of local translation rules, without running the danger of creating spurious nonsensical combinations. For instance, if "Guam" were the name of a ship as well as the name of a location, there could be one EFL constant *GUAM-EFL* with two WML-expansions: *GUAM-LOC* with type *LOCATIONS* and *GUAM-SHIP* with type *SHIPS*. Applying the EFL-to-WML rules to *F-SPEED(GUAM-EFL)* would nevertheless yield only *one* result, since the other combination would be deemed illegitimate.

In the next section we show how relational noun denotations and EFL-to-WML translations may be chosen in such a way that sentences involving relational nouns after an initially uniform treatment end up with plausible truth conditions -- so that, for instance, (5) above can be initially analyzed as (5a) and then translated into (5b) in a principled way.

4 MULTILEVEL SEMANTICS FOR RELATIONAL NOUNS

The treatment we propose is based on a simple, yet powerful idea: analyse a relational noun as denoting the *extension* of the corresponding relation *R* (i.e., the set of pairs *<x,y>* such that *R(x,y)*), and allow predicates to apply not only to individuals but also to such pairs.[6]

As an example, consider again the phrase *"Peter's sisters."* that we discussed in section 1 above. In the treatment we propose, this phrase would get the EFL analysis (6a).

[6] Terminology: We assume *directed relations*. If *<x,y>* is a pair in a relation-extension, we call *x* the *argument* and *y* the *value*.

(6) "Peter's sisters"

(6a) $\{x \in \textit{R-SISTER} \mid \textit{POSSESS}(\textit{PETER},x)\}$,

where *R-SISTER*, with the type[7]

$\mathrm{U}(\textit{MALES},\textit{FEMALES})\ \mathrm{X}\ \textit{FEMALES}$,

denotes the extension of the sister-relation, and where *POSSESS* has as one of its types:

$$\textit{FUNCTIONS}\ ((\mathrm{U}(\textit{MALES},\textit{FEMALES})\ \mathrm{X}\ \textit{FEMALES}),\ \textit{TRUTHVALUES}).$$

(6a) can be transformed into a plausible expression for (6) by applying the translation rule:

$$\textit{POSSESS} \Rightarrow (\lambda\, u,v{:}\ \ u = v[1])$$

where *u* has type *THINGS* and *v* has type *THINGS* X *THINGS*. Applying this rule to (6a) yields after β-reduction:

(6b) $\{x \in \textit{R-SISTER} \mid \textit{PETER} = x[1]\}$,

which is equivalent to:

(6c) $\{u,v \mid u = \textit{PETER}\ \ \&\ \textit{R-SISTER}(u,v)\}$

Thus, we see that by allowing the semantic translation of *"Peter's"* to select over pairs consisting of a person and the sister of that person, we can end up with a representation of *"Peter's sisters"* which comes close to having the right denotation: it denotes the correct set of persons, but they are still paired up with Peter. This "extra information" is of course a problem. For instance, *"Peter's sisters are Mary's aunts."* asserts the equality of two sets of persons, not two sets of pairs of persons.

It turns out that we have two distinct cases to deal with: to account for the interaction between a relational noun and the phrases which indicate its arguments and values, we would like to treat it as denoting a relation-extension; but to account for its interaction with everything else, we would like to treat it as denoting a set of individuals. In order to make the relational treatment yield the right results, we must assume that part of the meaning of ordinary descriptive predicates is an implicit projection-operator, which projects tuples onto their value-elements. This is the solution we adopt. We formalize it by means of a meaning-postulate schema which applies to every function *F* which is not among a small number of explicitly noted exceptions: $\forall\, x,y{:}\ F(x) = F(<y,x>)$

The copula *"be"* is not an exception to this meaning postulate schema: it operates on values rather than relation-elements. This is the reason why *"John"* is not available as an argument for *"brother"* in (2ac) above (*"Some punks of John's are brothers." "Some brothers are John's"*)

We shall now consider the actual EFL-to-WML translation rules which handle the relational nouns in a little more detail. The EFL relations have many different translations into WML; which ones are relevant in a given case, is decided by considering the semantic types of the arguments to which they are applied. Consider again, for example, the part of the EFL-to-WML translation rules that deals with the interpretation of the possessive relation as specifying a relational argument, as in *"Peter's sister", "Frederick's speed"*:

$$\textit{POSSESS} \Rightarrow (\lambda\, u,v{:}\ u = v[1])$$

where *u* has type *THINGS* and *v* has type *THINGS* X *THINGS*. Being a local translation rule, this rule could be applied to any occurrence of *POSSESS* in an EFL formula. However, many such applications would give rise to semantically anomalous WML formulas (with necessarily denotationless sub-expressions) which are filtered out if there are any other non-anomalous interpretations. For instance, the above rule for *POSSESS* would yield an anomalous expression if applied to the representation of *"Peter's cars"*, because the EFL constant *CARS* does not denote a set of pairs but a set of individual entities. It would also yield an anomalous expression if applied to *"The USS Frederick's sisters"*, because the type of the EFL constant *FREDERICK*, which is *SHIPS*, is disjoint with the argument type of *R-SISTER*, which is U*(MALES,FEMALES)*.

To avoid spurious generation of anomalous expressions, the semantic types of the arguments of an EFL function or EFL relation are checked before the EFL-to-WML rule for that function or relation is applied. For instance, the above rule for *POSSESS* will only be applied to an expression of the form *POSSESS(A,B)*, if *A* and *B* have types α and β such that

$$\exists\, P,Q{:}\ \ \beta = P\ \mathrm{X}\ Q\ \ \&\ \ \textit{NON-EMPTY}(\alpha \cap P).$$

As noted above, the interdefinability which exists between *"have", "of"*, the genitive, and *"with"*, when they are used, for instance, in reference to ownership, carries over to their use for indicating the relation between a relational noun and its argument. Thus, the EFL representations of *"of", "have"*, and *"with"* have WML translations which, modulo the order of their arguments, are all identical to the rule for *POSSESS* discussed above.

Function nouns, like *"speed"* and *"length"*, induce a special interpretation on preposition phrases with *"of"*. Such phrases can be used to connect the function noun with its *value*. The treatment of relational nouns sketched in the previous section can also accommodate this phenomenon, as we shall show now.

Consider example (7) below, which is identical to (5) above. It gets, in the treatment we propose, the EFL analysis (5a); this analysis is exactly analogous to the one that a syntactically similar sentence involving a non-relational noun would get. (Cf. (4) and (4a).)

[7] Notation: *A* X *B* denotes the set of pairs <*x*,*y*> such that *x* is in the denotation of *A* and *y* is in the denotation of *B*.

(7) *"Frederick has a speed of 15 knots."*

$$\text{(7a)}\ \exists\, s \in \{s \in \textit{F-SPEED} \mid \textit{OF}(s, \text{amount}(15, \textit{KNOTS}))\}: \textit{HAVE}(\textit{FREDERICK}, s)$$

It is the task of the EFL-to-WML translation rules to define a transformation on EFL expressions which would turn (5a) into (5b) or a logically equivalent formula.

$$\text{(7b)}\ \textit{F-SPEED}(\textit{FREDERICK}) = \text{amount}(15, \textit{KNOTS})$$

To achieve the desired result, we need a rule for *HAVE* which is precisely analogous to the rule for *POSSESS* above; and we need a rule for *OF* which is *not* analogous to the rule for *POSSESS* above: *"a speed of 15 knots"* is unlike *"the speed of the USS Frederick"* in that in the former case we must connect the relation with its *value* rather than its argument. The rule for *OF* that we need here is as follows:

$$\textit{OF} \Rightarrow (\lambda\, u,v{:}\ u[2] = v)$$

Note that different rules for one EFL constant can coexist without conflict, because of the assumption of lexical ambiguity in EFL. (In the general case, an EFL expression will have several WML expansions for this reason; often, many rule-applications will be blocked by semantic type-checking.)

This basic approach makes it possible to transform the EFL representation of any of the constructions shown in the examples in section 1 into reasonable World Model Language and Data Base Language formulations of the intended query. We shall illustrate the process of applying the EFL-to-WML translations and logical simplifications in a little more detail while showing how to extend this treatment to function nouns which can take more than one argument. Such nouns interact with specific kinds of preposition phrases to pick up their arguments. For instance: *"Frederick's distance to Hawaii"*, *"the distance from Hawaii to Guam"*. As an example, we will now discuss the noun *"readiness"* as used in the U.S. Navy, which designates a *two*-argument function.

"Readiness", as used in the Navy battle management domain, refers to the degree to which a vessel - to be more precise, a unit - is prepared for combat or for a specific mission. This degree is indicated on a five-point scale, using either c-codes (C1 to C5), if referring to combat readiness, or m-codes (M1 to M5), if referring to mission readiness. The readiness for combat can furthermore be the overall readiness (the default case) or the readiness with respect to one of the four resource readiness areas: personnel, training, equipment or supplies. Therefore, *READINESS-OF* is a function which maps two arguments, an element of *SHIPS* and an element of *READINESS-AREAS*, into *READINESS-VALUES*.

Consider as an example the noun phrase *"the readiness of Frederick"*. If we ignore for the moment the effect of the "singular the" operator (see section 5), its initial translation is:

$$\{x \in \textit{READINESS-OF} \mid \textit{OF}(x, \textit{FREDERICK})\}$$

The parts of this expression are translated as follows. A logical transformation translates the function-constant *READINESS-OF* into the following equivalent expression, which will be convenient for subsequent processing:

$$\{x \in \text{domain}(\textit{READINESS-OF}) \times \text{range}(\textit{READINESS-OF}) \mid \textit{READINESS-OF}(x[1]) = x[2]\}$$

which in its turn is equivalent to

$$\{x \in (\textit{SHIPS} \times \textit{READINESS-AREAS}) \times \textit{READINESS-VALUES} \mid \textit{READINESS-OF}(x[1]) = x[2]\}$$

The relation *OF* is eliminated in the EFL-to-WML transformation by a variant[8] of the translation rule mentioned above. It transforms

$\textit{OF}(x, \textit{FREDERICK})$ into $x[1][1] = \textit{FREDERICK}$

The net result of these logical and descriptive transformations is the following expression:

$$\{x \in \{z \in (\textit{SHIPS} \times \textit{READINESS-AREAS}) \times \textit{READINESS-VALUES} \mid \textit{READINESS-OF}(z[1]) = z[2]\} \mid x[1][1] = \textit{FREDERICK}\}$$

This expression is then simplified to:

$$\{z \in (\{\textit{FREDERICK}\} \times \textit{READINESS-AREAS}) \times \textit{READINESS-VALUES} \mid \textit{READINESS-OF}(z[1]) = z[2]\}$$

which in its turn can be transformed into a logically equivalent but more optimally evaluable expression[9]:

$$(\textbf{for}{:}\ \{\textit{FREDERICK}\} \times \textit{READINESS-AREAS},\ \textbf{apply}{:}\ (\lambda\, x{:} <x, \textit{READINESS-OF}(x)>))$$

(The actual system may apply further transformations (from WML into DBL), if it has to account for discrepancies between the database structure and the canonical domain model, possibly followed by further optimizations at the DBL level.)

Other restrictions on *"readiness"*, as in *"the readiness <u>on personnel</u>"*, *"the <u>personnel</u> readiness*, or *"a <u>c1</u> readiness"*, are handled in an analogous manner:

$$\textit{ON} \Rightarrow (\lambda\, u,v{:}\ u[1][2] = v)$$
$$\textit{PREMOD} \Rightarrow (\lambda\, u,v{:}\ u[1][2] = v)$$
$$\textit{PREMOD} \Rightarrow (\lambda\, u,v{:}\ u[2] = v)$$

where *PREMOD* is the EFL translation of the elided relation in a noun-noun compound. (Note that if the same preposition is used with different nouns to mark different argument places, we need a more elaborate notation which identifies the arguments of a function by labels rather than by position.)

[8] Multi-argument functions are viewed as functions on n-tuples. *OF* specifies, in this case, the first element of the argument-n-tuple.

[9] Notation: (**for**: A, **apply**: F) denotes the bag containing the results of all applications of the function F to elements of the set A.

Because of the essentially local character of the descriptive transformations on *HAVE, OF, ON, PREMOD*, etc., and the completely general character of the simplifications dealing with intersections of sets and tuples, a small number of transformations (a few for each EFL relation) covers a wide variety of expressions.

5 IMPLICIT ARGUMENTS.

One or more of the arguments of a relation may be unspecified in the input sentence, while the intent of the utterance is nevertheless that a particular argument should be filled in. The present section discusses briefly how this issue can be dealt with during a phase of semantic processing which follows the EFL-to-WML translation.

The most important case arises from the use of *definite descriptions* in the English input sentence. The phrase "the readiness of Frederick", for instance, leads to an expression which has the operator "the" wrapped around the expression which represents "readiness(es) of Frederick". "the" is a pragmatic operator, which selects the single most salient element out of the set that it operates on.

Where the expression representing "readiness of Frederick on personnel" would denote a set containing exactly one tuple, the expression representing "readiness of Frederick" denotes a set containing a number of different tuples: ones with *EQUIPMENT, PERSONNEL, OVERALL*, etc., filled in as the second argument. Eliminating the "the" operator consists in accessing a *Discourse Model* to find out which of the fillers of the second argument place is contextually most accessible. (We assume that available discourse referents are stored at every level of embedding in a recursive model of discourse surface structure, such as [9]). If none of the readiness areas were mentioned in an accessible discourse constituent, the system defaults to the "unmarked" readiness area, i.e., *OVERALL*.

Plural definite noun phrases are treated in a similar fashion. For instance, "the readinesses of Frederick" leads to an expression in which a pragmatic operator selects the contextually salient *multiple element subset* of the tuples in the extension of *READINESS-OF* which have *FREDERICK* as a first argument. In this case, if no particular subset of the readiness areas can be construed as a discourse referent, the system defaults to the assumption that here the overall readiness plus the four resource readinesses are intended. (Another possibility being the reference to the ship's readiness *history*: a sequence of past, current and projected future readinesses.)

6 RELATION EXTENSIONS AS ANSWERS.

The decision to treat relational nouns as denoting relation extensions has an immediate consequence, of some practical importance for question-answering systems, concerning the way in which wh-questions involving relational nouns are answered. For example, the request *"List the speeds of the ships in the Indian Ocean."* could be answered in three ways, of ascending informativeness: 1) with a set of speed values (possibly of smaller cardinality than the set of ships in the Indian Ocean) 2) with a bag of speed values (of the same cardinality as the set of ships) and 3) with a set of <ship, speed> ordered pairs, such that each ship is paired off with its speed.

Clearly, 3) is most likely to be the desired response (although it is possible to envision situations where reponses 1) and 2) are desired). One cannot obtain this response, however, if the semantic translation of the noun phrase *"the speeds of the ships in the Indian Ocean"* does not retain the information of which speed goes with which ship. An important advantage of our approach to the relational noun problem is that it preserves this information, making 3) the normal reponse and 1) and 2) derivable from it.

This may be compared to the "procedural semantics" approach to this same problem, as found in the work on LUNAR [14]. In this work, meaning is regarded as procedural in nature, and quantifications are represented in terms of nested iterations. The request *"List the speeds of the ships in the Indian Ocean"* would be represented as:

```
(FOR ALL X / SHIPS
        : (IN X INDIAN-OCEAN)
        ; (PRINT (SPEED X)))
```

where the action of this representation would be to iterate over the class *SHIPS*, for each member checking to see if it is *IN* the *INDIAN-OCEAN*, and if so, printing its speed. The PRINT operator is made "smart" enough to detect the occurrence of the free variable in its argument and to add in a printout the value of this variable for each iteration.

Note that while this representation provides for the tuple response (3), and perhaps, if the "smartness" is made optional, for the bag response (2), the set response (1) would seem out of reach. In contrast, the approach this paper presents allows for all three, by generating as a default response the tuple set, and then optionally "projecting" on its second column.

7 CONCLUSION

Relational nouns are of primary importance for natural language interfaces to databases and expert systems, since they are commonly used to refer to database relations and to arithmetical functions. This paper has presented a treatment of relational nouns which manages to maintain uniformity and generality at the level of syntactic analysis and initial semantic interpretation. This treatment has been incorporated into the semantic framework of BBN's Spoken Language System without writing additional LISP code. The semantic transformations necessary for the treatment are all carried out by general algorithms which were part of the pre-existing semantic framework. Implementing the treatment consisted in writing descriptive (EFL to WML) translation specifications for the EFL relations involved with function nouns, and a few dozen logical transformations to supplement the existing set of simplifications.

Further work on this topic should investigate how our perspective on relational nouns carries over to an account of the temporal and spatial modifiers that can be used with *any* noun. This will then make it possible to explore its connections with the work on the semantics of time-dependent nouns that has been done in the Montague-tradition. [3] [13]

ACKNOWLEDGMENTS

We thank David Stallard for important contributions to the ideas presented here; Jan Landsbergen for his share in the development of the conceptual framework that inspired this research; Damaris Ayuso and Sean Boisen for their assistance in applying our results to BBN's Spoken Language System.

REFERENCES

[1] Bronnenberg, W.J.H.J., H.C. Bunt, S.P.J. Landsbergen, R.J.H. Scha, W.J. Schoenmakers, E.P.C. van Utteren.
The Question Answering System PHLIQA1.
In L. Bolc (editor), *Natural Language Question Answering Systems*, pages 217-305. London: MacMillan, 1980.

[2] Clark, Eve V.
What's in a word? On the Child's Acquisition of Semantics in his First Language.
In *Cognitive Development and the Acquisition of Language*. Academic Press, New York, 1973.

[3] Janssen, Theo.
Individual Concepts are Useful.
In Fred Landman and Frank Veltman (editors), *Varieties of Formal Semantics*, pages 171-192. Foris Publications, Dordrecht, 1984.

[4] Landman, Fred.
Groups.
Technical Report, Un. of Massachusetts, Amherst, MA, 1987.

[5] Landsbergen, S.P.J. and R.J.H. Scha.
Formal Languages for Semantic Representation.
In S. Allen and J. Petofi (editors), *Aspects of Automatized Text Processing*. Buske, Hamburg, 1979.

[6] Montague, Richard.
The Proper Treatment of Quantification in Ordinary English.
In J. Hintikka, J. Moravcsik and P. Suppes (editors), *Approaches to Natural Language*. Reidel, Dordrecht, 1973.

[7] Partee, Barbara H.
Compositionality.
In Fred Landman and Frank Veltman (editors), *Varieties of Formal Semantics*. Foris Publications, Dordrecht, 1984.

[8] Piaget, Jean.
Judgment and Reasoning in the Child.
Humanities Press, New York, 1928.

[9] Polanyi, Livia and Remko Scha.
A Syntactic Approach to Discourse Semantics.
In *Proceedings of Coling 84*, pages 413-419. Stanford University, Stanford, CA, July, 1984.

[10] Scha, Remko J.H.
Logical Foundations for Question Answering.
Technical Report, Philips Research Labs, Eindhoven, M.S. 12.331, 1983.

[11] Scha, Remko and David Stallard.
Multi-Level Plurals and Distributivity.
In *Proceedings of the 26th Annual Meeting of the ACL*. Buffalo, NY, 1988.

[12] Stallard, David.
The Logical Analysis of Lexical Ambiguity.
In *Proceedings of the 25th Annual Meeting of the ACL*. Stanford University, Stanford, CA, July, 1987.

[13] Thomason, Richmond H.
Home is where the hearth is.
In *Contemporary Perspectives in the Philosophy of Language*. Un. of Minnesota Press, Minneapolis, 1979.

[14] Woods, William A.
Semantics and Quantification in Natural Language Question Answering.
In M. Yovits (editor), *Advances in Computers*, pages 1-87. Academic Press, 1978.

[15] Zoeppritz, Magdalena.
The meaning of 'of' and 'have' in the USL system.
AJCL 7(2):109-119, 1981.