

DISCOURSE ENTITIES IN JANUS

Damaris M. Ayuso
BBN Systems and Technologies Corporation
10 Moulton Street
Cambridge, Massachusetts 02138
dayuso@bbn.com

Abstract

This paper addresses issues that arose in applying the model for discourse entity (DE) generation in B. Webber's work (1978, 1983) to an interactive multi-modal interface. Her treatment was extended in 4 areas: (1) the notion of context dependence of DEs was formalized in an intensional logic, (2) the treatment of DEs for indefinite NPs was modified to use skolem functions, (3) the treatment of dependent quantifiers was generalized, and (4) DEs originating from non-linguistic sources, such as pointing actions, were taken into account. The discourse entities are used in intra- and extra-sentential pronoun resolution in BBN Janus.

1 Introduction

Discourse entities (DEs) are descriptions of objects, groups of objects, events, etc. from the real world or from hypothesized or possible worlds that are evoked in a discourse. Any communicative act, be it spoken, written, gestured, or system-initiated, can give rise to DEs. As a discourse progresses, an adequate discourse model must represent the relevant entities, and the relationships between them (Grosz and Sidner, 1986). A speaker may then felicitously refer anaphorically to an object (subject to focusing or centering constraints (Grosz et al., 1983, Sidner 1981, 1983, Brennan et al. 1987) if there is an existing DE representing it, or if a corresponding DE may be directly inferred from an existing DE. For example, the utterance "Every senior in Milford High School has a car" gives rise to at least 3 entities, describable in English as "the seniors in Milford High School", "Milford High School", and "the set of cars each of which is owned by some senior in Milford High School". These entities may then be accessed by the following next utterances, respectively:

- "They graduate in June."
- "It's a good school."
- "They completely fill the parking lot."

Webber (1978, 1983) addressed the question of determining what discourse entities are introduced by a text. She defined rules which produce "initial descriptions" (IDs) of new entities stemming from noun phrases, given a meaning representation of a text. An ID is a logical expression that denotes the corresponding object and uses only information from the text's meaning representation. The declarative nature of Webber's rules and the fact that they relied

solely on the structure of the meaning representation, made her approach well suited for implementation.

The present work recasts her rules in Janus's intensional logic framework (described in section 2). Two goals guided our approach: (1) that our DE representations be semantically clear and correct according to the formal definitions of our language, and (2) that these representations be amenable to the processing required in an interactive environment such as ours, where each reference needs to be fully resolved against the current context.

In the following sections, we first present the representational requirements for this approach, and introduce our logical language (section 2).

Then we discuss issues that arose in trying to formalize the logical representation of DEs with respect to (1) the context dependence of their denotations, and (2) the indeterminacy of denotation that arises with indefinite NPs. For context dependence, we use an intensional logic expression indexed by time and world indices (discussed in section 3). This required us to extend Webber's rules to detect modal and other index-binding contexts. In representing DEs for indefinites (appearing as existential formulae in our meaning representation), we replaced Webber's EVOKE predicate with skolem constants for the independent case, where it does not contain a variable bound by a higher FORALL quantifier (section 4), and do not use EVOKE at all in the dependent case.

In section 5 we introduce a generalized version of the rules for generating DEs for dependent quantifiers stemming from indefinite and definite NPs which overcomes some difficulties in capturing dependencies between discourse entities.

In our multi-modal interface environment, it is important to represent the information on the computer screen as part of the discourse context, and allow references to screen entities that are not explicitly introduced via the text input. Section 6 briefly discusses some of these issues and shows how pointing actions are handled in Janus by generating appropriate discourse entities that are then used like other DEs.

Finally, section 7 concludes and presents plans for future work.

This is, to our knowledge, the first implementation of Webber's DE generation ideas. We designed the

algorithms and structures necessary to generate discourse entities from our logical representation of the meaning of utterances, and from pointing gestures, and currently use them in Janus's (Weischedel et al., 1987, BBN, 1988) pronoun resolution component, which applies centering techniques (Grosz et al., 1983, Sidner 1981, 1983, Brennan et al. 1987) to track and constrain references. Janus has been demonstrated in the Navy domain for DARPA's Fleet Command Center Battle Management Program (FCCBMP), and in the Army domain for the Air Land Battle Management Program (ALBM).

2 Meaning Representation for DE Generation

Webber found that appropriate discourse entities could be generated from the meaning representation of a sentence by applying rules to the representation that are strictly structural in nature, as long as the representation reflects certain crucial aspects of the sentence. This has the attractive feature that any syntactic formalism may be used if an appropriate semantic representation is produced. Some of the requirements (described in (Webber 1978, 1983)) on the representation are: (1) it must distinguish between definite and indefinite NPs and between singular and plural NPs, (2) it must specify quantifier scope, (3) it must distinguish between distributive and collective readings, (4) it must have resolved elided verb phrases, and (5) it must reflect the modifier structure of the NPs (e.g., via restricted quantification). An important implied constraint is that the representation must show one recognizable construct (a quantifier, for example) per DE-invoking noun phrase. These constructs are what trigger the DE generation rules.

Insofar as a semantic representation reflects all of the above in its structure, structural rules will suffice for generating appropriate DEs, but otherwise information from syntax or other sources may be necessary. There is a trade-off between using a level of representation that shows the required distinctions, and the need to stay relatively close to the English structure in order to only generate DEs that are justified by the text. For example, in Janus, in addition to quantifiers from NPs, the semantic representation has quantifiers for verbs (events), and possibly extra quantifiers introduced in representing deeper meaning or by the collective/distributive processing. Therefore, we check the syntactic source of the quantifiers to ensure that we only generate entities for quantifiers that arose from NPs (using the bound variable as an index into the parse tree).

Other than the caveat just discussed, the Janus meaning representation language WML (for World Model Language) (Hinrichs et al., 1987) meets all the other constraints for DE generation. WML is a higher-order intensional language that is based on a synthesis between the kind of language used in PHLIQA (Scha, 1976) and Montague's Intensional Logic

(Montague, 1973). A newer version of WML (Stallard, 1988) is used in the BBN Spoken Language System (Boisen et al., 1989). The intensionality of WML makes it more powerful than the sample language Webber used in developing her structural rules.

The scoping expressions in WML have a sort field (which restricts the range of the variable) and have the form:

$$(B \ x \ S \ (P \ x))$$

where B is a quantifier such as FORALL or EXISTS, a term-forming operator like IOTA or SET, or the lambda abstraction operator LAMBDA. S is the sort, a set-denoting expression of arbitrary complexity specifying the range of x, and (P x) is a predication in terms of x. The formal semantics of WML assigns a type to each well-formed expression which is a function of the types of its parts. If expression E has type T, the denotation of E, given a model M and a time t and world w, is a member of the set which is T's domain. One use of types in our system is for enforcing selectional restrictions. The formation rules of WML, its type system, and its recursive denotation definition provide a formal syntax and semantics for WML.

3 Context Dependence of Discourse Entities

A formal semantics was assumed though not given for the sample logical language used by Webber. The initial descriptions (IDs) of DEs produced by her rules were stated in this language too, and thus are meant to denote the object the DE represents. For example, the rule which applies to the representation for independent definite NPs assigns to the resulting DE an ID which is the representation itself:

$$(\iota \ x \ S \ (P \ x)) \Rightarrow ID: (\iota \ x \ S \ (P \ x))$$

where ι is Russell's iota operator. Thus, the ID for "the cat" in "I saw the cat" is $(\iota \ x \ cats \ T)$. (Since the body of the ι in this example has no additional predication on x, it is merely T, for TRUE.) However, because IDs are solely drawn from the meaning representation of the isolated text, they may not suffice to denote a unique object. Connection to prior discourse knowledge or information from further discourse may be necessary to establish a unique referent, or determining the referent may not even be necessary. For example, the ID for "the cat" would need to be evaluated in a context where there is only one salient cat in order to obtain a denotation.

Our system's representation of a DE is a structure containing several fields. The "logical-form" field contains a WML expression which denotes the object the DE describes (this corresponds roughly to Webber's ID). Given that WML is intensional, we are able to explicitly represent context dependence by having the logical form include an intensional core, plus tense, time, and world information (which includes discourse context) that grounds the intension so that it may be

evaluated. For example, the logical form for the DE corresponding to "the cat" in our system is

```
((INTENSION (IOTA x cats T))
 time world)
```

where *time*, if unfilled, defaults to the present, and *world* defaults to the real world and current discourse state. The semantics of our IOTA operator makes it denotationless if there is not exactly one salient object that fits the description in the context, else its denotation is that unique object. In our interactive system each reference needs to be fully resolved to be used successfully. If unknown information is necessary to obtain a *unique* denotation for a IOTA term, a simple clarification dialogue should ensue. (Clarification is not implemented yet, currently the set of all values fitting the IOTA is used.)

An example using the time index is the noun phrase "the ships that were combat ready on 12/1/88", which would generate a DE with logical form:

```
((INTENSION
 (PAST (INTENSION
 (IOTA x (SETS ships)
 (COMBAT-READY x)))) )
 12/1/88 world)
```

Representing this time index in the logical form is crucial, since a later reference to it, made in a different time context must still denote the original object. For example, "Are they deployed?" must have "they" refer to the ships that were combat ready on 12/1/88, not at the time of the latter utterance.

In order to derive the proper time and world context for the discourse entities, we added structural rules that recognize intensional and index-binding logical contexts. Our DE generation algorithm uses these rules to gather the necessary information as it recurses into the logical representation (applying rules as it goes) so that when a regular rule fires on a language construct, the appropriate outer-scoping time/world bindings will get used for the generated DEs.

It should be noted that, as the discussion above suggests, a definite NP always gives rise to a new discourse entity in our system. If it is determined to be anaphoric, then a pointer to the DE it co-refers with (when found) will be added to its "refers-to" field, indicating they both denote the same object.

4 DEs for Independent Indefinite NPs

In Webber's work, the initial description (ID) for a DE stemming from an independent existential (i.e., with no dependencies on an outer FORALL quantifier), contained an EVOKE predicate. "I saw a cat":

```
(EXISTS x cats (saw I x))
```

would generate a DE with ID:

```
(I x cats
 (& (saw I x) (EVOKE Sent x)))
```

"The cat I saw that was evoked by sentence Sent", where Sent is the parsed clause for "I saw a cat". The purpose of EVOKE was to make clear that although more than one cat may have been seen, the "a" picks out one in particular (which one we do not know except that it is the one mentioned in the utterance), and this is the cat which makes the EVOKE true. Any subsequent reference then picks out the same cat because it will access this DE. The semantics of the EVOKE predicate and the type of the S argument (which is syntactic in nature) were unclear, so we looked for a different formulation with better understood semantics.

Predicate logic already provides us with a mechanism for selecting arbitrary individuals from the domain via skolem functions (used as a mechanism for removing existentials from a formula while preserving satisfiability). Skolem functions have been used in computational linguistics to indicate quantifier scope, for example (VanLehn, 1978). Following a suggestion by R. Scha, we use skolem functions in the logical form of the DE for the "indefinite individuals" introduced by independent existentials (Scha et al., 1987). For clarity and consistency with the rest of the language, we use a sorted skolem form, where the range of the function is specified. Since we use this for representing existentials that are independent, the function has no arguments and is thus equivalent to a sorted constant whose denotation is undetermined when introduced. (In this sense it is consistent with Karttunen's (1976) and Kamp's (1984) view of the indefinite's role as a referential constant, but unlike Kamp, here the sentence's meaning representation is separate from the representation of the evoked entity.)

Thus we introduced a new operator to WML named SKOLEM, for expressions of the form (SKOLEM *n* <sort>), where *n* is an integer that gets incremented for each new skolem created, as a way of naming the skolem function. For the example above, the core logical form (stripping the outer intension and indices) for the DE of "a cat" would be:

```
(SKOLEM 1 (SET x cats (saw I x)))
```

denoting a particular cat from the set of all the cats I saw. The type of a SKOLEM expression is well-defined and is given by the following type rule:

```
TYPEOF (SKOLEM INTEGERS (SETS a))
 = a
```

where INTEGERS is the type for integers, and (SETS *a*) is the type of sets whose members have type *a*. This type rule says that when the first argument of SKOLEM is of type INTEGER, and the second is a set with elements of type *a*, then the type of the SKOLEM expression is *a*. Therefore, the type of the above example is cats. The explicit connection to the originating sentence which the EVOKE predicate provided is found in our scheme outside of the logical

representation by having a pointer in the DE's structure to the parse tree NP constituent, and to the structure representing the communicative act performed by the utterance (in the fields "corresponding-constituent" and "originating-communicative-act", respectively). These connections are used by the pronoun resolution algorithms which make use of syntactic information.

Does the denotation of a skolem constant ever get determined? In narrative, and even in conversation, identifying the individual referred to by the indefinite NP frequently doesn't occur. However, in our interactive system, each reference must be fully resolved. When the evaluation component of Janus determines a successful value to use for the existential in the text's logical form, the appropriate function denotation for SKOLEM *n* gets defined, and the "extension" field is set for the discourse entity.

Note that many interesting issues come up in the treatment of reference to these indefinite entities in a real system. For example, cooperative responses by the system introduce new entities that must be taken into account. If the user asks "Is there a carrier within 50 miles of Hawaii?", a cooperative "There are two: Constellation and Kennedy" (as opposed to just "Yes") must add those two carriers as entities, which now overshadow the singular skolem entity for "a carrier within 50 miles of Hawaii". On the other hand, a "No" answer should block any further reference to the carrier skolem, since its denotation is null, while still allowing a reference to a class entity derived from it, as in "Is there one near San Diego?" where *one* refers to the class *carriers*.

The treatment presented works for straightforward cases of independent indefinites. Trickier cases like donkey sentences (Kamp, 1984, Webber, 1981) and interactions with negation have not yet been addressed.

5 Dependent NPs

5.1 Dependent Indefinite NPs

Our work uncovered a need for modifications in Webber's structural rules for quantifiers from indefinite and definite NPs which have dependencies on variables bound directly or indirectly by an outer FORALL quantifier. In this section we address the case of dependent existentials arising from indefinite NPs. We first argue that the predicate EVOKE is not needed in this context. Then we point out the need for generalizing the rule to take into account not just FORALL, but all scoping operators that intervene between the outer FORALL and the inner EXISTS. Finally, we show that the dependencies between discourse entities must be explicitly maintained in the logical forms of newly created DEs that depend on them.

Webber's rules are designed to apply from the outermost quantifier in; each time a rule is applied the remaining logical form is modified to be in terms of the just created DE. For example, "Every boy saw a girl he knows" has logical form (for the bound pronoun reading):

```
(FORALL x boys
  (EXISTS y (SET y' girls
    (knows x y'))
    (saw x y)))
```

The first step is to apply the rule for an independent universal quantifier:

```
R0: (FORALL x S (P x)) => de: S
```

This application yields the entity for "the set of all boys"

```
DE1: boys
```

and we rewrite the logical form to be:

```
(FORALL x DE1
  (EXISTS y (SET y' girls
    (knows x y'))
    (saw x y)))
```

The steps shown so far are consistent with both Webber's and our approach. Now we want to apply the general rule for existentials within the body of a distributive, in order to generate an entity for the relevant set of girls. Webber uses Rule 3 in (Webber, 1983) (here corrected to position the existential's sort S inside the scope of the outer quantifiers in the generated DE):

```
R3: (FORALL y1...yk
  (EXISTS x S (P x))) =>
  de: (SET x things
    (EXISTS y1...yk
      (& (member x S) (P x)
        (EVOKE Sent x))))
```

where FORALL y₁...y_k is shorthand for FORALL y₁ de₁ ... (FORALL y_k de_k), analogously for EXISTS, and S or P depends directly or indirectly on y₁...y_k.

Now the first DE we want to generate with this rule is for "the set of girls, each of which is known by some boy in DE₁, and was seen by him". Does each girl in the set also have to satisfy an EVOKE predicate? It seems that any future reference back to the set formed by the existential seeks to obtain all items fitting the description, not some subset constrained by EVOKE. For example, if the example above is followed by "the girls tried to hide", taking "the girls" anaphorically, one wants all the girls seen by some boy in DE₁, that knows them, no less. Our core logical representation for the set of girls is thus:

```
DE2: (SET y girls
  (EXISTS x DE1
    (& (knows x y) (saw x y))))
```

So the modified rule used in producing DE₂ is:

```

R3': (FORALL y1...yk
      (EXISTS x S (P x))) =>
de: (SET x St
      (EXISTS y1...yk
      (& (member x S) (P x))))

```

where EVOKE has been removed, and the DE's sort field is S_t for the "root type" of S , which is the type of the members of S , in order to appropriately constrain the DE's sort (instead of leaving it as the unconstrained "things").

A second change that needs to be made is to generalize the left hand side of the rule so that the scoping expressions outscoping the inner EXISTS in the pattern also be allowed to include other scoping operators, such as EXISTS and IOTA. As long as the outermost quantifier is a FORALL, any other dependent scoping expression within it will generate a set-denoting DE and will behave as a distributive environment as far as any more deeply embedded expressions are concerned. In other words, the distributiveness chains along the dependent quantifiers. To see this, consider the more embedded example "Every boy gave a girl he knew a peach she wanted", where there is an intervening existential between the outer FORALL and innermost EXISTS. The core logical form for this sentence is:

```

(FORALL x boys
  (EXISTS y (SET y' girls
    (knows x y'))
  (EXISTS z (SET z' peaches
    (wants y z'))
  (gave x y z)))

```

DE_1 would be as above. Using rule R3' DE_2 becomes:

```

DE2:
(SET y girls
  (EXISTS x DE1
    (& (knows x y)
    (EXISTS z (SET z' peaches
      (wants y z'))
    (gave x y z))))

```

"The set of girls, each of which is known by some boy in DE_1 , and got a peach she wanted from that boy." Now the peach quantifier should generate a set DE in terms of DE_1 and DE_2 . Applying R3' gives us:

```

DE3: (SET z peaches
  (EXISTS x DE1
    (EXISTS y DE2
      (& (wants y z)
      (gave x y z))))

```

"The set of peaches z such that there is a girl in DE_2 (who is known by some boy in DE_1 , and who got some peach she wanted from the boy), who wants z , and who got it from some boy in DE_1 ".

Now a third and final problem becomes apparent: for the general case of arbitrary embedding of dependent quantifiers we generate a DE (e.g., DE_3) de-

pendent on other DEs from the outer quantifiers, but the dependencies between those DEs (e.g., DE_1 and DE_2) are not maintained. This is counter-intuitive, and also leads to an under-specified set DE. In the peaches example above, envision the situation where a boy b_1 gave out two peaches p_1 and p_2 : one to a girl g_1 he knew, and one to a girl g_2 he didn't know, who also got a peach p_3 from another boy b_2 who did know her. These are the facts of interest in this scenario:

1. (& (gave b₁ g₁ p₁) (know b₁ g₁)
 (wants g₁ p₁))
2. (& (gave b₁ g₂ p₂)
 (NOT (know b₁ g₂))
 (wants g₂ p₂))
3. (& (gave b₂ g₂ p₃) (know b₂ g₂)
 (wants g₂ p₃))

Since b_1 and b_2 are in DE_1 (due to facts 1 and 3), and g_2 is in DE_2 (due to fact 3), then p_2 is in DE_3 (due to fact 2 and according to the DE_3 logical form above). But p_2 should not be in DE_3 , since p_2 was NOT given to a girl by a boy *she knew*. The set of peaches obtained for DE_3 is too large. The problem would not arise if in the DE_3 logical form, the variables ranging over DE_2 were appropriately connected to DE_1 using the dependent restriction present in the original formula (*knows x y*). A correct DE_3 is:

```

DE3:
(SET z peaches
  (EXISTS x DE1
    (EXISTS y (SET y' DE2
      (knows x y'))
      (& (wants y z)
      (gave x y z))))

```

To be able to do this, the rule-application algorithm must be modified to include the restriction information (for dependent restrictions) when the formula gets rewritten in terms of a newly created DE. Therefore the final generalized rule, which includes other scoping operators and works on properly connected DEs is as follows:

```

R3'': (FORALL v1 S1
      Q2 v2 S2 ... Qn vn Sn
      (EXISTS x S (P x))) =>
de: (SET x St
      (EXISTS v1 S1 ... vn Sn
      (& (member x S) (P x))))

```

where S or P depend directly or indirectly on $v_1 \dots v_n$. Q_i may be FORALL, EXISTS, or IOTA, and the scoping operators outside the inner EXISTS have already been processed by any appropriate rules that have replaced their original sorts by the S_i s, which are in terms of generated DEs and explicitly show any DE dependencies. The right hand side is as before, with existentials picking out elements from each outer quantifier.

act. Since "them" and "it" have different number requirements, there is no ambiguity and the anaphor resolution module resolves "them" to the DE corresponding to "the C1 carriers in the Indian Ocean" and "it" to the DE for Kennedy. We are currently working on having system-initiated actions also generate entities.

7 Conclusions and Further Work

Webber's general approach to discourse entity generation from a logical representation proved very useful in our efforts. We were able to recast her basic ideas in our logical framework, and currently use the generated DEs extensively.

The fact that the generation of DEs is done via structural rules operating on a semantic representation provided a degree of modularity that allowed our pronoun resolution component to work automatically when we combined a new syntactic component with our semantic and discourse component (replacing an ATN by a unification grammar, in an independently motivated experiment). We are currently starting to port the DE generation component to the BBN Spoken Language System (Boisen et al., 1989), and plan to integrate it with the intra-sentential mechanisms in (Ingria and Stallard, 1989). The fact that entity representations are mostly semantic in nature, not syntactic, also facilitated the addition and use of non-linguistic entities in a uniform way.

There are several areas that we would like to study to extend our current treatment. We want to address the interactions between centering phenomena and non-linguistic events that affect discourse focus, such as changing contexts via a menu selection in an expert system.

Our paraphrasing component (Meteer and Shaked, 1988) already uses the discourse entities to a limited extent. One area of future work is to have the language generator make more extensive use of them, so it can smoothly refer to focused objects.

Finally, although quantified expressions are already generated in Janus for events implicit in many verbs, they are not being used for DEs. We would like to address the problem of event reference and its interaction with temporal information, using ideas such as those in (Webber, 1988) and in the special issue of *Computational Linguistics* on tense and aspect (Vol. 14, Number 2 June 1988).

8 Acknowledgments

The work presented here was supported under DARPA contract #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 of the United States Government. The author would like to thank Dave Stallard for invaluable discussions during the writing of this paper. Thanks also to Remko Scha, Lance Ramshaw, Ralph Weischedel, and Candy Sidner.

References

- BBN Systems and Technologies Corp. (1988). *A Guide to IRUS-II Application Development in the FCCBMP* (BBN Report 6859). Cambridge, MA: Bolt Beranek and Newman Inc.
- Boisen, S., Chow Y., Haas, A., Ingria, R., Roucos, S., Scha, R., Stallard, D., and Vilain, M. (1989). *Integration of Speech and Natural Language: Final Report* (BBN Report 6991). BBN Systems and Technologies Corp.
- Brennan, Susan E., Friedman, Marilyn W., and Pollard, Carl J. (1987). A Centering Approach to Pronouns. *Proceedings of the 25th Annual Meeting of the ACL*. ACL.
- Grosz, Barbara J., and Sidner, Candace L. (1986). Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12(3), 175-204.
- Grosz, Barbara J., Joshi, Aravind K., Weinstein, Scott. (1983). Providing a Unified Account of Definite Noun Phrases in Discourse. *Proceedings of the 21st Annual Meeting of the ACL*. Cambridge, MA: ACL.
- Hinrichs, E.W., Ayuso, D.M., and Scha, R. (1987). 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.
- Ingria, Robert J.P., and Stallard, David. (1989). A Computational Mechanism for Pronominal Reference. *Proceedings of the 27th Annual Meeting of the ACL*. ACL.
- Kamp, Hans. (1984). A Theory of Truth and Semantic Representation. In J. Groenendijk, T.M.V. Janssen, and M. Stokhof (Eds.), *Truth, Interpretation and Information, Selected Papers from the Third Amsterdam Colloquium*. Dordrecht: Foris Publications.
- Karttunen, Lauri. (1976). Discourse Referents. In J. D. McCawley (Ed.), *Syntax and Semantics, Volume 7*. New York: Academic Press.
- Meteer, Marie and Shaked, Varda. (1988). Strategies for Effective Paraphrasing. *Proceedings of COLING-88, Budapest, Hungary, August 22-27*. COLING.

5.2 Dependent Definite NPs

Some of the problems described in the previous section also arise for the rule to handle dependent definite NPs. Definite NPs are treated as IOTA terms in WML. (Webber's logical language in (Webber, 1978) used a similar ι . The treatment was later changed (Webber, 1983) to use the definite existential quantifier "Exists!", but this difference is not relevant for the following.) Replacing IOTA for ι in Webber's (1978) rule 5:

```
R5: (FORALL y1...yk
      (P (IOTA x S (R x)))) =>
  de: (SET z things
       (EXISTS y1...yk
          (= z (IOTA x S (R x)))))
```

where $y_1 \dots y_k$ are universal quantifiers over DEs as in R3 above, and S or R depend directly or indirectly on $y_1 \dots y_k$.

The second and third extensions discussed in the previous section are needed here too: generalizing the quantifiers that outscope the inner existential, and keeping the dependencies among the DEs explicit to avoid under-specified sets. An example of an under-specified set arises when the dependent IOTA depends jointly on more than one outer variable; for example, in "Every boy gave a girl he knew *the* peach *they selected*", each peach depends on the selection by a boy and a girl together. Take a scenario analogous to that in the previous section, with the facts now as follows (replacing "selected" for "wants"):

1. (ϵ (gave $b_1 g_1 p_1$) (know $b_1 g_1$)
 (selected (SETOF $b_1 g_1$ p_1))
2. (ϵ (gave $b_1 g_2 p_2$)
 (NOT (know $b_1 g_2$))
 (selected (SETOF $b_1 g_2$ p_2))
3. (ϵ (gave $b_2 g_2 p_3$) (know $b_2 g_2$)
 (selected (SETOF $b_2 g_2$ p_3))

By an analogous argument as before, using R5, the set of peaches will incorrectly contain p_2 , given by a boy to a girl who selected it with him, but whom he did not know. The modified rule is analogous to R3" in the previous section:

```
R5': (FORALL v1 S1
        (Q2 v2 S2 ... Qn vn Sn
         (P (IOTA x S (R x)))) =>
  de: (SET z St
       (EXISTS v1 S1 ... vn Sn
          (= z (IOTA x S (R x)))))
```

Note that this problem of under-specified sets does not arise when the dependency inside the IOTA is on one variable, because the definite "the" forces a one-to-one mapping from the possible assignments of the single outer variable represented in the IOTA to the IOTA denotations. If we use the example, "Every boy gave a girl he knew *the* peach she wanted", with logical form:

```
(FORALL x boys
  EXISTS y (SET y' girls
    (knows x y'))
    (gave x y (IOTA z peaches
      (wants y z))))
```

there is such a mapping between the set of girls in the appropriate DE_2 (those who got the peach they wanted from a boy they knew) and the peaches in DE_3 obtained via R5' (the peaches that some girl in DE_2 wanted). Each girl wants exactly one peach, so facts 2 and 3, where the same girl receives two different peaches, cannot occur. So the definite ensures that no scenario can be constructed containing extra items, as long as there is only one outer variable in the inner iota. However in the joint dependency example above using "selected", the one-to-one mapping is between boy-girl pairs and peaches, so the relationship between the boys and the girls becomes an integral part of determining the correct DE_3 .

6 Non-Linguistic Discourse Entities

In a dialogue between persons, references can be made not only to linguistically-introduced objects, but also to objects (or events, etc.) that become salient in the environment through some non-linguistic means. For example, a loud noise may prompt a question "What was *that*?", or one may look at or point to an object and refer to it, "What's wrong with *it*?". It seems an attention-drawing event normally precedes such a reference.

In the Janus human-computer environment, non-linguistic attention-drawing mechanisms that we have identified so far include pointing actions by the user, and highlighting (by the system) of changes on the screen as a response to a request (or for other reasons). The appearance of answers to questions also draws the user's attention. We incorporated these into generalized notion of a "communicative act" which may be linguistic in nature (English input or generated English output), a pointing gesture by the user, or some other system-initiated action. Any communicative act may give rise to DEs and affect the focused entities in the discourse.

We have implemented procedures to handle pointing actions by generating discourse entities which are then used in the pronoun resolution component uniformly with the others. For example, after the request "Show me the C1 carriers in the Indian Ocean" the system will display icons on the color monitor representing the carriers. The user can then say "Which of them are within 200 miles of it? <point with mouse to Kennedy>". Before the sentence gets processed, a discourse entity with the logical form (IOTA x carriers (nameof x "Kennedy")) will be created and added to the list of entities currently in focus (the "forward looking centers" of the last linguistic act); the DE's "originating-communicative-act" field will point to a newly created "pointing" communicative

- Montague, Richard. (1973). The Proper Treatment of Quantification in Ordinary English. In J. Hintikka, J. Moravcsik and P. Suppes (Eds.), *Approaches to Natural Language*. Dordrecht: Reidel.
- Scha, Remko J.H. (1976). Semantic Types in PHLIQA1. *Coling 76 Preprints*. Ottawa, Canada.
- Scha, Remko J.H., Bruce, Bertram C., and Polanyi, Livia. (1987). Discourse Understanding. In *Encyclopedia of Artificial Intelligence*. John Wiley & Sons, Inc.
- Sidner, Candace L. (1981). Focusing for the Interpretation of Pronouns. *American Journal of Computational Linguistics*, 7(4), 217-231.
- Sidner, Candace L. (1983). Focusing in the Comprehension of Definite Anaphora. In M. Brady and R. C. Berwick (Eds.), *Computational Models of Discourse*. Cambridge, MA: MIT Press.
- Stallard, David G. (1988). A Manual for the Logical Language of the BBN Spoken Language System. Unpublished.
- Kurt VanLehn. (1978). *Determining the Scope of English Quantifiers* (Tech. Rep. 483). MIT Artificial Intelligence Laboratory.
- Webber, Bonnie L. (1978). *A Formal Approach to Discourse Anaphora* (BBN Report 3761). Cambridge, MA: Bolt Beranek and Newman.
- Webber, Bonnie L. (1981). Discourse Model Synthesis: Preliminaries to Reference. In Joshi, Webber, and Sag (Eds.), *Elements of Discourse Understanding*. Cambridge University Press.
- Webber, Bonnie L. (1983). So What Can We Talk About Now? In Brady and Berwick (Eds.), *Computational Models of Discourse*. MIT Press.
- Webber, Bonnie L. (1988). Discourse Deixis: Reference to Discourse Segments. *Proceedings of the 26th Annual Meeting of the ACL*. ACL.
- Weischedel, R., Ayuso, D., Haas, A., Hinrichs, E., Scha, R., Shaked, V., and Stallard, D. (1987). *Research and Development in Natural Language Understanding as Part of the Strategic Computing Program, Annual Technical Report December 1985 - December 1986* (BBN Report 6522). Cambridge, MA: Bolt Beranek and Newman.