RC23396 (W0411-005) November 1, 2004
Computer Science

IBM Research Report

Euphoria Semantic Analysis

Arendse Bernth
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Euphoria Semantic Analysis

Arendse Bernth
IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights
NY 10598
arendse@us.ibm.com

Abstract

The semantic analysis produced by the discourse understanding system Euphoria is described. Euphoria produces a semantic analysis spanning several sentences (a *discourse*) with coreference resolved and implicit arguments made explicit. The semantic analysis uses entity-oriented logical forms, built around a notion of extended entities that use generalized event variables. We describe the treatment of implicit arguments, time expressions, appositives, locational phrases; and higher-order entities ranging from entities derived from adverbs to compound events and paragraph topics.

1 Introduction

This report describes the semantic analysis produced by the discourse understanding system Euphoria, which is is built on top of the English Slot Grammar (ESG) [McCord, 1980, McCord, 1990, McCord, 1993]. Whereas ESG delivers a syntactic analysis on a sentence level, Euphoria produces a semantic analysis spanning several sentences (a *discourse*) with coreference resolved and implicit arguments made explicit. The coreference component used is an enhanced version of the system described in [Bernth, 2002].

The semantic analysis uses entity-oriented logical forms (EOLFs), which make use of *extended entities* (*EE*s). EEs include not only entities in the conventional sense (including named entities), but also events and relations. They are basically anything that can be referred to. One of the major foci of Situation Semantics [Barwise and Perry, 1983] was that most classes of words are referential, a point we agree with. The consequence of this view is a need to make all types of entities referrable. The solution in Situation Semantics was to utilize the notion of realism, a computationally somewhat vague idea. The same objective can be accomplished through the idea of indexing, first proposed by [Davidson, 1967]. Davidson's original idea covered indexing of verbs by so-called event variables, an idea that can be generalized to other entity types. Our "events" are indeed very general, along the lines described in [Hobbs, 1985] and [McCord and Bernth, 2004]. This notion of such generalized "events" furthermore has the advantage of allowing a "flat" semantic structure, a property that makes automatic reasoning easier. Both Situation Semantics and Discourse Representation Theory (DRT) [Kamp, 1981] point out the necessity of interpreting a discourse in context. Like DRT, Euphoria builds up the discourse structure within the context of the preceding discourse, but also takes some later discourse into consideration for certain types of cataphora. Other divergences from a DRT-like representation include the generalization of events as mentioned above, and the use of a type-free semantic representation (see [Menzel, 1986] for the advantages of a type-free semantic representation for natural language).

Section 2 describes the entity-oriented logical forms and their components. In Section 3 the treatment of implicit arguments is described, and Section 4 is devoted to time expressions. Disambiguation of the so-called "comma coordinations" for appositives, certain locational phrases, and true coordination is described in Section 5. Section 6 discusses several types of higher-order entities, ranging from entities derived from adverbs to compound events and paragraph topics. Finally, in Section 7 we indicate some topics for future work.
This report contains a great deal of examples of actual EOLFs, all produced by Euphoria.

2 Entity-oriented logical forms

In this section we describe the entity-oriented logical forms and their components.
An *entity-oriented logical form* (EOLF) consists of an extended entity E (called the *index* of the EOLF), together with a set S of *predications* that are "about" E, in the sense that E appears in each member of S.[1]
Currently, the EOLFs look like what is described in the following, but in order to handle word senses we plan to extend them as described in Section 7.
Each predication in an EOLF is of the form

(1) $(e\ arg_1\ arg_2\ \ldots\ arg_n)$

where both the predicate e and the arguments $arg_1, \ldots, arg_n$ are extended entities (EEs) (n may be 0). For example, if e is the usual sense of the verb "see", then e would be an event of seeing, with arg_1 as the subject, arg_2 as the object, and $n = 2$. Arguments generally follow the order of ESG slot filler arguments.
So we are using the main entity like the event arguments of Davidson, but they are not restricted to events. They can name/index any entity in the ontology, where **entity** is the unique top node of the ontology.
The semantic analysis of a document (or collection of documents) produced by Euphoria is a list of EOLFs that express the semantic content of the document. The list is *entity-oriented* in two senses:

- Every entity E mentioned in the document has associated with it an EOLF for which it is the index: All the things that are said about E are listed with E and are accessible efficiently from E.
- For the predicates that appear in the EOLFs, both the predicates and their arguments are entities, except for certain special predicates described in Section 2.3.

Entity types are described in Section 2.1, and special entities and special predicates are listed in Section 2.2 and Section 2.3, respectively.

2.1 Entity Types

In this section we describe the various entity types. Even though no type conformance is required for entities, it is useful to retain a trace of the *kind* of entity, as indicated by the part of speech of the word that gives rise to the entity. This is useful for both text generation from the EOLF and for constraining inference.

[1] In the examples, we will display the EOLFs as follows: `Index` $<$ `(Predication`$_1$ `...` `Predication`$_n$`)`.

Generally, entities are given reference identifications (*refID*s, for short) of the form *word#NT*, e.g. *see#33V*, where *word* is derived from the first mention in the text of the entity, *N* is a unique number, and *T* is one or more characters indicating the part of speech of the first mention.
The following types are currently produced:

A Adjective. Example: *small#2A*.

Adv Adverb. Example: *quickly#5Adv*.

G Generic. Example: *elephant#1G*.

P Preposition. Example: *above#15P*.

V Verb. Example: *see#3V*. Verbs occurring within intensional contexts, for instance *reads* in *John believes that Mary reads*, may further be marked with *i* as in *read#33Vi*. Currently only the head of the embedded clause is marked with *i*, and from this it is possible for the user to infer that all the embedded entities are within the intensional context. In future versions of Euphoria we may mark *all* embedded entities, as a service to the user. We do not currently have any plans to disambiguate the intensional context.

If no type is given, the type defaults to noun. Example: *house#120*. Note that currently generics are assumed to be derived from nouns only.[2]

2.2 Special Entities

In addition to the entities directly reflecting mentions in the text, there are the following special entities:

u An unfilled argument slot. Example: *(eat#5 u mango#4)*. Here the first argument, the deep subject, is unspecified. This could stem from a passive construction like *The mango was eaten*.

you-imp The implicit subject of an imperative verb. For example, *Read the book!* is represented as *(read#2V you-imp book#1)*. See Section 3.1.

year#n Entity for time expressions involving a year. *n* is (a string representing) an integer indicating the year. For example, *year#1998* means the year *1998*. See Section 4 for further treatment of time expressions.

month#n Entity for time expressions involving a month. *n* is an integer indicating the month. For example, *month#12* means the month *December*. See Section 4 for further treatment of time expressions.

date#N Entity for the time expressions *today*, *yesterday* and *tomorrow*. *N* is a term indicating the date. Following ESG conventions about naming of dates, dates are given as *(WD MD M Y)* where *WD* is weekday, *MD* is day of the month, *M* is month, and *Y* is year. Values are integers. *WD* starts with 1 for Monday and ends with 7 for Sunday; month and year as above. If a field is not available it is set to 0 (zero). An example of a fully resolved date is *date#(4 12 8 2004)*, which means *Thursday, August 12, 2004*. See Section 4 for further treatment of time expressions.

[2]Later, in Section 6.1.1, we shall encounter some examples, examples (19) and (21), that could be candidates for generics derived from verbs.

2.3 Special Predicates

Entities, which may function as predicates, are derived from specific mentions in the text. However, there are also a number of special predicates, which are not entities, and which do not *directly* reflect any mentions, but rather derived attributes. These are as follows:

card This predicate indicates the *cardinality* of an entity. For example, *(card pilot#2 sing)* states that the cardinality of the entity *pilot#2* is singular. Other values for *card* may be *plur*, a specific number, or a generalized quantifier such as *many*. *Negation* is considered a special case where *card* is zero; hence the cardinalty of the seeing event *see#3V* in example (2) is zero. For verbs, the cardinality is only given in case of negation, but not for positive statements. Note also that for negation, the cardinality is always attached to the verb, but this leaves undecided what the scope of the negation is.

```
(2) a. John did not see Mary.
    b. John#1 < ((see#3V John#1 Mary#2  u) (card John#1 sing))
       Mary#2 < ((see#3V John#1 Mary#2  u) (card Mary#2 sing))
       see#3V < ((see#3V John#1 Mary#2  u) (card see#3V 0))
```

compound_event This predicate indicates that a single word in the text covers more than one event. For example, *(compound_event cross#17V cross#18V cross#19V)* means that *cross#17V* consists of the two events *cross#18V* and *cross#19V*. Example (42) illustrates this in more detail.

loc This predicate indicates the *location*. For example, *(loc Alaska#1 snow#4V)* means that the location of the entity *snow#4V* is *Alaska#1*.

is_in This predicate specifies a geographical location within another geographical location. For example, *(is_in Marrakech#8 Morocco#9)* means that *Marrakech#8* is in *Morocco#9*. See Section 5.3.

time This predicate indicates the *time*. For example, *(time year#1991 cross#19V)* means that the time of the entity *cross#19V* is *1991*.

poss This predicate indicates *possession*. It may reflect an *'s*-possessive in the text, a possessive pronoun, or the verb *have* with an object. For example, *John has a house*, *John's house*, and *His house* (assuming that *His* is coreferential with *John*)[3] will all produce the predication *(poss John#1 house#2 u)*.

3 Implicit Arguments

The EOLFs include a number of arguments that are implicit in the surface structure. Some of these arguments are derived directly from the deep level of the ESG parses; others are decided on by Euphoria.
Several parts of speech can have implicit arguments; here we focus on implicit subjects of verbs. These occur with the nonfinite forms of verbs and the imperative mood. We shall here consider implicit subjects of imperatives, infinitives, and present participles, as well as implicit deep objects of past participles, which appear as subjects on the surface level.

[3] Also assuming that this is an attributive statement; see Section 5.2.

The rest of this section is organized as follows. Section 3.1 describes implicit subject of imperatives; Section 3.2 describes infinitives, including verb complements and adjectival complements; Section 3.3 present participles; and Section 3.4 past participles.

3.1 Imperatives

The implicit subject of an imperative is whoever the imperative is addressed to. In dialogue, this may be "the other party". As we are currenly only concerned with narrative text, we will take the "addressee" to be the reader, and assign a special entity *you-imp* to designate such an implicit subject, as illustrated by example (3). Furthermore we will assume that this implicit subject (the reader) is of semantic type *human*.

```
(3) a. Inflate the balloon!
    b. balloon#1  < ((inflate#2V you-imp balloon#1  u) (card balloon#1 sing))
       inflate#2V < ((inflate#2V you-imp balloon#1  u))
```

3.2 Infinitives

Implicit subjects of infinitives occur with infinitive complements of verbs, nouns, adjectives etc.

3.2.1 Verb Complements

Maybe the simplest example of an implicit subject for an infinitive, syntactically speaking, is a sentence like *John can swim*, where the bare infinitive *swim* is a complement of the modal verb *can*. Euphoria is not currently handling modals, though, so we will go on to a more complicated example in (4), where the main verb has both an object, *Mary*, and an infinitive complement *swim*. Since *swim* is a complement of *see*, it shows up in the argument list of that word. And *Mary* is identified as the implicit subject of *swim*.[4]

```
(4) a. John sees Mary swim.
    b. John#1  < ((see#3V John#1 Mary#2  swim#4V))
       Mary#2  < ((see#3V John#1 Mary#2  swim#4V) (swim#4V Mary#2 u))
       see#3V  < ((see#3V John#1 Mary#2  swim#4V))
       swim#4V < ((see#3V John#1 Mary#2  swim#4V) (swim#4V Mary#2 u))
```

Infinitives can also be complements of verbs which require the infinitive marker *to*.
Example (5) shows a simple example of this. Here *John* is identified as the subject of *swim*. And since *want* creates an intensional context, the refID for *swim* is marked with an *i*, as described in Section 2.1.

```
(5) a. John wants to swim.
    b. John#1   < ((want#2V John#1 swim#3Vi ) (swim#3Vi John#1 u u))
       swim#3Vi < ((want#2V John#1 swim#3Vi ) (swim#3Vi John#1 u u))
       want#2V  < ((want#2V John#1 swim#3Vi ))
```

Example (6) combines the cases illustrated in example (4) and example (5). The main verb has both an object and an infinitive complement, and the context is intensional. *Mary* is identified as the implicit subject of *swim*.

[4] We shall not go further into the semantics of perception verbs in this report.

```
(6) a. John wants Mary to swim.
    b. John#1   < ((want#3V John#1 Mary#2  swim#4Vi))
       Mary#2   < ((want#3V John#1 Mary#2  swim#4Vi) (swim#4Vi Mary#2 u u))
       swim#4Vi < ((want#3V John#1 Mary#2  swim#4Vi) (swim#4Vi Mary#2 u u))
       want#3V  < ((want#3V John#1 Mary#2  swim#4Vi))
```

3.2.2 Adjective Complements

Infinitive complements of adjectives pose special problems. Example (7) illustrates the simplest case where the implicit subject of *go* clearly is *John*.

```
(7) a. John was clever to go.
    b. John#2    < ((clever#1A John#2 go#3V) (go#3V John#2 u u))
       clever#1A < ((clever#1A John#2 go#3V))
       go#3V < ((clever#1A John#2 go#3V) (go#3V John#2 u u))
```

In example (7) we can justifiably conclude that John actually goes. However, this is by no means *always* the case. Some adjectives clearly do not allow us to conclude that the activity described by the infinitive complement actually took place. If we substitute *afraid* for *clever* in example (7) we get *John was afraid to go*. Depending on context, John may or may not have gone. According to our scheme of marking verbs within intensional contexts with an *i*, we should mark the refID for *go* with an *i* in this case, but that is an issue that we will disregard for now.
In the above examples involving adjective complements, the overt subject of the sentence provided the implicit subject of the infinitive. This may not always be the case. Consider the sentence *The book was enjoyable to read*. Here the overt subject *the book* supplies not the implict subject of *read*, but rather the object. Distinquishing these cases is by no means trivial and we shall disregard this ambiguity for now.

3.3 Present Participles

Present participles following an object pose an interesting ambiguity in that they can be attached either to the subject or the object. The implicit subject of the participle depends on the attachment. The controlled-language checker EEA [Bernth, 1997] identifies this type of ambiguity, and offers disambiguated rewriting suggestions reflecting the different attachment possibilities, but does not make a decision on which attachment is correct. Using techniques similar to those described in [Bernth, 1998] for EEA to *identify* the ambiguity, and the lexicon of selectional preferences described in [Bernth and McCord, 2003] to *make a choice*, we can resolve the implicit subject. The lexicon provides us with preferences for the semantic types of the complements of a verb. For example, the verb *eat* strongly prefers an object of type *food* and a subject of type *animate*, even though there are exceptions, of course.
An example of resolving an implicit subject of a present participle is given in (8).
This sentence is structurally ambiguous. Who is *wearing old shoes and rubber gloves*? There are two possibilities here: either the subject in the main clause or the object. A further complication in this example is that the subject in the main clause is implicit, due to the verb being an imperative. According to the common parsing philosophy of preferring close attachment, the *walnuts* are thus accoutred with shoes and gloves, as evidenced by the ESG parse shown in (9), but real-world knowledge tells us that humans are much more likely to wear gloves than walnuts are.

```
(8) a. Harvest the walnuts wearing old shoes and rubber gloves.
    b. and#4            < ((wear#7V you-imp and#4  u)
                           (and#4 shoe#3G rubber gloves#5G))
       harvest#6V       < ((harvest#6V you-imp walnut#2  u))
       old#1A           < ((old#1A shoe#3G))
       rubber gloves#5G < ((and#4 shoe#3G rubber gloves#5G))
       shoe#3G          < ((old#1A shoe#3G) (and#4 shoe#3G rubber gloves#5G))
       walnut#2         < ((harvest#6V you-imp walnut#2  u) (card walnut#2 plur))
       wear#7V          < ((wear#7V you-imp and#4  u))
```

(9) "Harvest the walnuts wearing old shoes and rubber gloves."

```
o--------- top    harvest1(1,u,3) verb vimpr human_agent nhuman_object (harvest#6V)
| .------- ndet   the1(2)         det pl def the ingdet
‘-+------- obj(n) walnut1(3)      noun cn pl st_tree st_nut (walnut#2)
  ‘------- nnfvp  wear1(4,3,7,u)  verb ving (wear#7V)
    |   .- nadj   old1(5)         adj erest adjnoun (old#1A)
    | .--- lconj  shoe1(6)        noun cn pl st_shoe (shoe#3G)
    ‘-+--- obj(n) and0(7)         noun cn pl cord st_shoe st_clothes (and#4)
      | .- nnoun  rubber1(8)      noun cn sg massn
      ‘--- rconj  glove1(9)       noun cn pl st_clothes (rubber gloves#5G)
```

The confidence score given by the selectional constraints lexion for humans wearing gloves is 0.101124 whereas the confidence score for walnuts is only 0.001873; hence the attachment of *wearing* to the implicit subject of *Harvest* is preferred, and the implicit subject of *wearing* is determined to be *you-imp*. The implicit subject of the present participle *flying* in example (10) is also resolved using this technique.
A similiar technique is applied to prepositional phrase attachment.

3.4 Past Participles

Whereas present participles are inherently active in nature, past participles are inherently passive. This is reflected in example (10) by the fact that the deep subject position for *dip* is unfilled, as indicated by the presence of *u*, whereas the deep object position is filled by *wing#9*.

```
(10) a. The plane hit the mountain flying with its right wing dipped downwards.
     b. dip#7V         < ((dip#7V u wing#9  u) (downwards#1Adv dip#7V))
        downwards#1Adv < ((downwards#1Adv dip#7V))
        fly#6V         < ((fly#6V plane#3 u u) (with#8P wing#9 fly#6V))
        hit#5V         < ((hit#5V plane#3 mountain#4  u))
        mountain#4     < ((hit#5V plane#3 mountain#4  u) (card mountain#4 sing))
        plane#3        < ((hit#5V plane#3 mountain#4  u) (card plane#3 sing)
                          (fly#6V plane#3 u u) (poss plane#3 wing#9))
        right#2A       < ((right#2A wing#9))
        wing#9         < ((poss plane#3 wing#9) (with#8P wing#9 fly#6V)
                          (right#2A wing#9) (dip#7V u wing#9  u)
                          (card wing#9 sing))
        with#8P        < ((with#8P wing#9 fly#6V))
```

4 Time Expressions

Many documents such as newspaper articles include a publication date near the top. We try to identify the presence of such a date and resolve temporal references within the text relative to this date. For example *today* would resolve to the document date and *yesterday* to the day before, and so forth.
Example (11) is simplified from a newspaper article for illustrative purposes. Using the document date, *07/19/98*, we can resolve *last year* to *1997*.

```
(11) a. 07/19/98.
        Atlantic City had fewer applicants last year.
     b. Atlantic City#2 < ((have#4V Atlantic City#2 applicant#3G  u)
                           (card Atlantic City#2 sing))
        applicant#3G    < ((have#4V Atlantic City#2 applicant#3G  u)
                           (card applicant#3G plur) (fewer#1A applicant#3G))
        fewer#1A        < ((fewer#1A applicant#3G))
        have#4V         < ((have#4V Atlantic City#2 applicant#3G  u)
                           (time year#97 have#4V))
        year#97         < ((time year#97 have#4V))
```

The date could alternatively have been specified as *July 19, 98*. In case of date expressions given with slashes as in example (11), the current interpretation is American style, i.e. the month is assumed to appear before the day of the month.
Example (12) illustrates resolution of *yesterday*. The full date *date#(0 30 6 98)* is given, with correct wrap-around for the month and day.

```
(12) a. 07/01/98.
        Atlantic City had fewer applicants yesterday.
        b. Atlantic City#2  < ((have#4V Atlantic City#2 applicant#3G  u)
                              (card Atlantic City#2 sing))
           applicant#3G     < ((have#4V Atlantic City#2 applicant#3G  u)
                              (card applicant#3G plur) (fewer#1A applicant#3G))
           date#(0 30 6 98) < ((time date#(0 30 6 98) have#4V))
           fewer#1A         < ((fewer#1A applicant#3G))
           have#4V          < ((have#4V Atlantic City#2 applicant#3G  u)
                              (time date#(0 30 6 98) have#4V))
```

Currently handled are years and months (*this year/month*, *last year/month*, and *next year/month*), and *today*, *yesterday* and *tomorrow*. Further work will address e.g. weekdays.

5 Coordination

ESG coordinated phrases are underspecified semantically. Different meanings have the same surface syntactic structure. This is particularly the case for the so-called "comma coordinations": coordination involving one or more commas. Here we shall look at some cases of comma coordination: true coordination, described in Section 5.1; appositives, described in Section 5.2; and comma-separated locational phrases, described in Section 5.3.
A special case of coordination, not involving comma coordination, warrants a separate treatment, which will be given in Section 6.4.

5.1 True coordination

Example (13) shows the parse of a sentence that contains what one could call "true coordination", viz. list items conjoined by commas and *and*.[5]

(13) "Hydrogen, helium, and air lift balloons."

```
  .--- lconj   hydrogen1(1)     noun cn sg (hydrogen#1)
.-+--- subj(n) ,(101)           noun cn pl cord massn
| | .- lconj   helium1(2)       noun cn sg massn (helium#2)
| `-+- rconj   and0(3)          noun cn pl cord massn (and#3)
|   `- rconj   air2(4)          noun cn sg massn (air#4)
o----- top     lift2(5,101,6,u) verb vfin vpres pl vsubj (lift#6V)
`----- obj(n)  balloon1(6)      noun cn pl st_artifact (balloon#5G)
```

The corresponding Euphoria output is shown in (14).

```
(14) a. Hydrogen, helium, and air lift balloons.
     b. air#4      < ((and#3 hydrogen#1 helium#2 air#4))
        and#3      < ((and#3 hydrogen#1 helium#2 air#4)
                      (lift#6V and#3 balloon#5G  u) (card and#3 plur))
        balloon#5G < ((lift#6V and#3 balloon#5G  u) (card balloon#5G plur))
        helium#2   < ((and#3 hydrogen#1 helium#2 air#4))
        hydrogen#1 < ((and#3 hydrogen#1 helium#2 air#4))
        lift#6V    < ((lift#6V and#3 balloon#5G  u))
```

Scope ambiguities could be introduced by premodifiers of the conjuncts; that is an issue that we are not currently addressing.

5.2 Appositives

A defining characteristic of appositives is that they are coreferent with the noun they modify. However, treating them as purely referential loses important meaning. More useful is to consider them *attributive* in the sense of [Donnellan, 1966]. Donnellan defines referential and attributive uses in the context of definite descriptions:

> A speaker who uses a definite description attributively in an assertion states something about whoever or whatever is the so-and-so. A speaker who uses a definite description referentially in an assertion, on the other hand, uses the description to enable his audience to pick out whom or what he is talking about and states something about that person or thing. In the first case, the definite description may be said to occur essentially, for the speaker wishes to assert something about whatever or whoever fits that description; but in the referential use the definite description is merely one tool for doing a certain job–calling attention to a person or thing–and in general any other device for doing the same job, another description or a name, would do as well. In the attributive use, the attribute of being the so-and-so is all important, while it is not in the referential use.

[5] If a comma coordination cannot be resolved to an appositive or a locational coordination, the analysis defaults to true coordination.

Appositives are clearly attributive in this sense.
Example (15) contains two appositives. One is *chairman of the Virgin Group*, which attributes chairmanship of the Virgin Group to *Branson*; the other is *Fossett*, which gives the name of *his competitor*. Because appositives are attributive, i.e. predicate something about the entity in question, it makes sense to represent them as predicates in the EOLF.

```
(15) a. Branson, chairman of the Virgin Group, believes that his competitor,
        Fossett, faces grave risks.
     b. Branson#2      < ((chairman#8 Virgin Group#3 Branson#2)
                          (believe#6V Branson#2 face#7Vi  u) (card Branson#2 sing)
                          (competitor#9 Branson#2 Fossett#4))
        Fossett#4      < ((chairman#8 Virgin Group#3 Branson#2)
                          (believe#6V Branson#2 face#7Vi  u)
                          (card Branson#2 sing) (competitor#9 Branson#2 Fossett#4)
                          (face#7Vi Fossett#4 risk#5G  u) (card Fossett#4 sing))
        Virgin Group#3 < ((chairman#8 Virgin Group#3 Branson#2))
        believe#6V     < ((believe#6V Branson#2 face#7Vi  u))
        chairman#8     < ((chairman#8 Virgin Group#3 Branson#2))
        competitor#9   < ((competitor#9 Branson#2 Fossett#4))
        face#7Vi       < ((believe#6V Branson#2 face#7Vi  u)
                          (face#7Vi Fossett#4 risk#5G  u))
        grave#1A       < ((grave#1A risk#5G))
        risk#5G        < ((face#7Vi Fossett#4 risk#5G  u) (card risk#5G plur)
                          (grave#1A risk#5G))
```

The ESG parse in (16) shows that ESG treats the first appositive as a "comma coordination" similar to (13).[6]

(16) "Branson, chairman of the Virgin Group, believes that his competitor, Fossett, faces grave risks."

```
  .------- lconj          Branson(1)        noun propn sg notfnd (Branson#2)
.-+------- subj(n)        ,(101)            noun cn sg h cord notfnd humind
| `------- rconj          chairman1(2,3)    noun cn sg h humind (Branson#2)
|   `----- nobj(n)        of1(3,6)          prep pprefn nonlocp
|     | .- ndet           the1(4)           det sg def the ingdet
|     `--- objprep(n)     virgin1 Group1(6) noun propn sg h ctitle (Virgin Group#3)
o--------- top            believe1(7,101,8) verb vfin vpres sg vsubj (believe#6V)
`--------- obj(thatc)     that1(8,12)       thatconj
  |   .--- ndet           his1(9)           det sg possdet notfnd propn masc (Branson#2)
  | .-+--- subj(n)        competitor1(10,u) noun cn sg h humind (Fossett#4)
  | | `--- nprop          Fossett(11)       noun propn sg notfnd (Fossett#4)
  `-+----- thatcomp(bfin) face2(12,10,14)   verb vfin vpres sg vsubj (face#7Vi)
    | .--- nadj           grave1(13)        adj erest (grave#1A)
    `----- obj(n)         risk2(14,u)       noun cn pl (risk#5G)
```

[6]Different versions of the lexicons used with ESG give different results for the treatment of the second appositive; the standard lexicon, not used in this application, gives a comma coordination for the second appositive as well.

5.3 Comma-separated locational phrases

Example (17) illustrates the case of a comma-separated locational phrase. Only the relevant parts of the EOLFs are given. The locational phrase gives rise to the special predicate *is_in*.

```
(17) a. Lindstrand said that because of unfavorable weather patterns over England
        he and his colleagues had decided to launch their Virgin Global Challenger
        from a military airfield at Marrakech, Morocco.
     b. Marrakech#8 < ((is_in Marrakech#8 Morocco#9) (loc Marrakech#8 launch#12Vi))
        Morocco#9   < ((is_in Marrakech#8 Morocco#9))
```

The parse of an abbreviated version, shown in (18), makes it clear that ESG treats the locational phrase *Marrakech, Morocco* as a comma coordination. Euphoria utilizes semantic type checking on the left and right conjuncts, as well as the presence of a locational preposition to identify this case and produce the locational relation.

(18) "Lindstrand decided to launch the Virgin Global Challenger at Marrakech, Morocco."

```
.--------- subj(n)      Lindstrand(1)                noun propn sg notfnd (Lindstrand#1)
o--------- top          decide1(2,1,3)               verb vfin vpast sg vsubj (decide#5V)
‘--------- obj(inf)     infto(3,4)                   infto
  ‘------- tocomp(binf) launch2(4,1,8,u,u)           verb vinf (launch#6V)
    | .--- ndet         the1(5)                      det sg pl def the ingdet
    ‘----- obj(n)       Virgin Global Challenger(8)  propn sgpl (Virgin Global Challenger#2)
    ‘----- vprep        at1(9,110)                   prep pprefv staticp
      | .- lconj        Marrakech(10)                noun propn sg notfnd (Marrakech#3)
      ‘-+- objprep(n)   ,(110)                       noun propn sg h cord notfnd st_country
        ‘- rconj        Morocco1(11)                 noun propn sg h st_country (Morocco#4)
```

6 Higher-order Entities

Higher-order entities are reifications of predications. Being entities themselves, such reifications become objects of a higher order in the logic, and may themselves be arguments of predicates. Higher-order entities of this sort are typically derived from mentions in the text, and include such parts of speech as verbs, adverbs, and adjectives.

But there are also higher-order entities that have no one-to-one correspondence with mentions. These are for example events described by a paragraph of text. The text may not explicitly mention the event of ballooning, but that's what the text is really about. This concept of "higher-order" is somewhat different from the one mentioned above.

Considering these notions of higher-order entities, we see that some higher-order entities are on a higher level than others; e.g. the event that is described by a whole paragraph is on a higher order than an adverb that describes the manner in which an action is performed. In the following, we will classify some higher-order entities and describe how Euphoria treats them.

In Section 6.1 we describe the lower-most higher-order entities such as nominals, adverbs and subordinate conjunctions.

Slightly higher-level entities such as demonstrative pronouns are discussed in Section 6.2.

Paragraph topics and document summaries are entities of the highest order, and Section 6.3 is devoted to a proposed treatment of these.

6.1 "Lower" higher-order entitities

On the lower-most end of the scale we have nominal clauses; constructions involving extraposed clausal subjects; adverbs; subordinate conjunctions; and verbs taking *that*-complements.

6.1.1 Nominal clauses

Nominal clauses are clauses having a function approximating that of a noun phrase; they may function as subject, object, complement, appositive, and object of preposition. [Quirk *et al.*, 1972] defines the following categories of nominal clauses: The *that*-clause, the dependent interrogative clause, the nominal relative clause, the *to*-infinitive clause and the *ing*-clause. Here we shall consider the *to*-infinitive nominal clause, the *-ing* nominal clause, and the nominal *that*-clause, including extraposition of clausal subject. These refer to events and are thus inherently higher-order.

Nominal *to*-infinitive clauses Example (19) illustrates the case of an infinitive in the role of subject of the verb *be*. The infinitive *swim* does not have a subject.

```
(19) a. To swim is easy.
     b. easy#1A < ((easy#1A swim#2V))
        swim#2V < ((swim#2V  u) (easy#1A swim#2V))
```

In this type of construction, if a subject of the infinitive is called for, it will usually be preceded by a *for*. We can modify example (19) in this manner as shown in (20).

```
(20) a. For John to swim is easy.
     b. John#2  < ((swim#3V John#2 u) (card John#2 sing))
        easy#1A < ((easy#1A swim#3V))
        swim#3V < ((swim#3V John#2 u) (easy#1A swim#3V))
```

However, if the clause is the direct object, then the *for* is omitted: *John wants Mary to leave* is dealt with in Section 3.2.1 on implicit subjects.
Example (21) illustrates subject complement.

```
(21) a. To swim is to move.
     b. move#2V < ((move#2V swim#1V u) (move#2V  u))
        swim#1V < ((swim#1V  u) (move#2V swim#1V u))
```

In example (22) the nominal clause is an adjectival complement.

```
(22) a.John is happy to help Mary.
     b. John#2   < ((happy#1A John#2 help#4V) (help#4V John#2 Mary#3  u)
                    (card John#2 sing))
        Mary#3   < ((help#4V John#2 Mary#3  u) (card Mary#3 sing))
        happy#1A < ((happy#1A John#2 help#4V))
        help#4V  < ((happy#1A John#2 help#4V) (help#4V John#2 Mary#3  u))
```

Nominal *-ing* clauses Example (23) illustrates the case of an *-ing* nominal clause in the subject position. Variation (23a) shows the subject of the nominal clause as a genitive, and in (23b) the subject is in the objective case.[7] These two variations are given the same analysis.

```
(23) a. John's going was easy.
     b. John going was easy.
     c. John#2  < ((go#3V John#2))
        easy#1A < ((easy#1A go#3V))
        go#3V   < ((go#3V John#2) (easy#1A go#3V))
```

A more complex example of a nominal *-ing* clause is given in (24a). Here the subject of the *-ing* verb is indicated by a prepositional phrase *by John*, and the object by still another prepositional phrase (*of the meeting*). Euphoria unwraps and normalizes this construction as shown in (24b).

```
(24) a. The joining of the meeting by John was a surprise.
     b. John#2     < ((join#4V  John#2 meeting#1))
        join#4V    < ((join#4V  John#2 meeting#1) (surprise#3 join#4V))
        meeting#1  < ((join#4V  John#2 meeting#1))
        surprise#3 < ((surprise#3 join#4V))
```

In example (25), the *-ing* clause is the object.

```
(25) a. John enjoyed going.
     b. John#1   < ((enjoy#2V John#1 go#3V  u) (card John#1 sing) (go#3V John#1 u))
        enjoy#2V < ((enjoy#2V John#1 go#3V  u))
        go#3V    < ((enjoy#2V John#1 go#3V  u) (go#3V John#1 u))
```

Nominal *that*-clauses In example (26) the *that*-clause is the subject.

```
(26) a. That John went was a problem.
     b. John#1    < ((go#3V John#1 u) (card John#1 sing))
        go#3V     < ((go#3V John#1 u) (problem#2 go#3V u))
        problem#2 < ((problem#2 go#3V u) (card problem#2 sing))
```

Example (27) shows the *that*-clause as object.

```
(27) a. John told Mary that Bill left.
     b. Bill#3   < ((leave#5V Bill#3 u) (card Bill#3 sing))
        John#1   < ((tell#4V John#1 leave#5V Mary#2 u) (card John#1 sing))
        Mary#2   < ((tell#4V John#1 leave#5V Mary#2 u) (card Mary#2 sing))
        leave#5V < ((tell#4V John#1 leave#5V Mary#2 u) (leave#5V Bill#3 u))
        tell#4V  < ((tell#4V John#1 leave#5V Mary#2 u))
```

[7] The case can be seen by substituting a pronoun for *John*. This construction is described by [Quirk *et al.*, 1972] (see p. 741) as possible, albeit very informal.

Extraposition of clausal subject A clausal subject may also be extraposed to the end of the sentence. This gives rise to constructions with an anticipatory (or pleonastic) *it*. An example is given in (28).

```
(28) a. It was a problem that John went.
     b. John#2    < ((go#3V John#2 u) (card John#2 sing))
        go#3V     < ((problem#1 go#3V u) (go#3V John#2 u))
        problem#1 < ((problem#1 go#3V u) (card problem#1 sing))
```

The clausal (and semantic) subject *that John went* is placed at the end of the sentence, whereas the syntactic subject is the anticipatory *it*.
Generally, sentences involving this type of extraposition begin with the anticipatory *it* followed by a VP followed by the clausal subject. Even though the anticipatory *it* traditionally is considered redundant or empty (*pleonastic*), closer analysis reveals that this pronoun does indeed have a referent.[8] The example in (28) is equivalent to *That John went was a problem*. In other words, the thing that was a problem, was that John went, and hence the referent of *it* is the event of John going, which in our setup is represented by the refID for *go*. Therefore we predicate *problem#1* of *go#3V*, making this a higher-order EOLF. Furthermore, it is necessary to properly identify the subject of the embedded verb, if any. In this case, the subject is *John*.
In (29) we show a more complicated example. Here we note that the deep object of *recommend* as given by the ESG parse is the pleonastic pronoun, which is obviously not suitable for our purposes. Rather, the deep object should be *obtain*, and we mark this in the parse tree for *it* and also mark this node in the semantic features as *pleonastic*.

(29) "It is recommended that appropriate client consent be obtained."

```
.------- subj(n)        it(1)               noun pron sg pleonastic (obtain#4V)
o------- top            be1(2,1,3)          verb vfin vpres sg vsubj stative
‘------- pred(en)       recommend1(3,u,1,u) verb ven vpass (recommend#3V)
‘------- vextra         that1(4,8)          thatconj
|     .- nadj           appropriate1(5,u)   adj (appropriate#1A)
|     .- nnoun          client1(6,u)        noun cn sg h humind
|   .--- subj(n)        consent1(7,u)       noun cn sg st_agreement (client consent#2G)
  ‘-+--- thatcomp(bfin) be1(8,7,9)          verb vfin vpres pl vsubj stative
    ‘--- pred(en)       obtain1(9,u,7)      verb ven vpass (obtain#4V)
```

The resulting semantic analysis of (29) is shown in (30). Here we note the proper object of *recommend*.

```
(30) a. It is recommended that appropriate client consent be obtained.
     b. appropriate#1A    < ((appropriate#1A client consent#2G))
        client consent#2G < ((appropriate#1A client consent#2G)
                             (obtain#4V u client consent#2G  u)
                             (card client consent#2G sing))
        obtain#4V         < ((recommend#3V u obtain#4V  u)
                             (obtain#4V u client consent#2G  u))
        recommend#3V      < ((recommend#3V u obtain#4V  u))
```

[8] This is one reason we chose to follow [Quirk *et al.*, 1972]'s terminology and use *anticipatory* rather than *pleonastic* for the syntactic subject *it*.

Example (28) had a noun complement of *be*. In (31) it is adjectival. Variation (31a) shows an extraposed *that*-clause, whereas variation (31b) shows an extraposed infinitival clause.

```
(31) a. It was necessary that John cleaned the house.
     b. It was necessary for John to clean the house.
     c. John#2       < ((clean#4V John#2 house#3  u) (card John#2 sing))
        clean#4V     < ((necessary#1A  clean#4V) (clean#4V John#2 house#3  u))
        house#3      < ((clean#4V John#2 house#3  u) (card house#3 sing))
        necessary#1A < ((necessary#1A  clean#4V))
```

Example (32) differs from the preceding ones in that the pleonastic *it* is followed by a verb not *be*.

```
(32) a. It appears that John cleaned the house.
     b. John#1    < ((clean#4V John#1 house#2  u) (card John#1 sing))
        appear#3V < ((appear#3V clean#4V u))
        clean#4V  < ((appear#3V clean#4V u) (clean#4V John#1 house#2  u))
        house#2   < ((clean#4V John#1 house#2  u) (card house#2 sing))
```

Some other extraposed cases are shown in examples (33), (34), and (35).

```
(33) a. It is rare when John cleans the house.
     b. John#2   < ((clean#4V John#2 house#3  u) (card John#2 sing))
        clean#4V < ((rare#1A clean#4V ) (clean#4V John#2 house#3  u))
        house#3  < ((clean#4V John#2 house#3  u) (card house#3 sing))
        rare#1A  < ((rare#1A clean#4V ))

(34) a. It is the case that John cleaned the house.
     b. John#2   < ((clean#4V John#2 house#3  u) (card John#2 sing))
        case#1   < ((case#1 clean#4V u) (card case#1 sing))
        clean#4V < ((case#1 clean#4V u) (clean#4V John#2 house#3  u))
        house#3  < ((clean#4V John#2 house#3  u) (card house#3 sing))

(35) a. It was a mistake for John to go.
     b. John#2    < ((go#3V John#2 u) (card John#2 sing))
        go#3V     < ((mistake#1 go#3V u) (go#3V John#2 u))
        mistake#1 < ((mistake#1 go#3V u) (card mistake#1 sing))
```

6.1.2 Adverbs

Adverbs are very heterogeneous. Even classifying adverbs is controversial. For example, while ESG considers words like *here* and *there* pronouns, many would consider them adverbs. The view of the adverb class as a hodge-podge sort of class is supported by [Quirk *et al.*, 1972] (see p. 267):

> Because of its great heterogeneity, the adverb class is the least satisfactory of the traditional parts of speech. Indeed, it is tempting to say simply that the adverb is an item that does not fit the definitions for other parts of speech.

A consequence of this diversity is that adverbs generally can modify mosts parts of speech.[9]

[9] Adverbs used as particles, as in *John gave* UP *reading*, are in a class by themselves.

In this section we shall look at adverbs modifying verb phrases and adjective phrases; these are obviously higher-order entities. Time adverbs, such as *today* were handled in Section 4.
In example (36) we show a "manner adverb". The manner in which John swims, viz. *well* is predicated of *swim*.

```
(36) a. John swims well.
     b. John#2 < ((swim#3V John#2 u u) (card John#2 sing))
        swim#3V < ((swim#3V John#2 u u) (well#1Adv swim#3V))
        well#1Adv < ((well#1Adv swim#3V))
```

Example (37) shows a sentential adverb. We follow the conventions of ESG in letting a sentential adverb predicate on the entity that represents the head of the clause. Hence *probably* predicates on *take place*.

```
(37) a. The balloon launching will probably take place.
     b. balloon launching#2 < ((take place#3V balloon launching#2 u u)
                               (card balloon launching#2 sing))
        probably#1Adv       < ((probably#1Adv take place#3V))
        take place#3V       < ((probably#1Adv take place#3V)
                               (take place#3V balloon launching#2 u u))
```

In example (38) the adverb *definitely* predicates on an adjective, *appropriate*.

```
(38) a. John provided a definitely appropriate solution.
     b. John#3               < ((provide#5V John#3 solution#4  u)
                                (card John#3 sing))
        appropriate#2A       < ((definitely#1Adv appropriate#2A)
                                (appropriate#2A solution#4))
        definitely#1Adv      < ((definitely#1Adv appropriate#2A))
        provide#5V           < ((provide#5V John#3 solution#4  u))
        solution#4           < ((provide#5V John#3 solution#4  u)
                                (card solution#4 sing)
                                (appropriate#2A solution#4))
```

In this treatment we have totally ignored the fact that adverbs (and adjectives) are best treated as focalizers [McCord, 1987, McCord and Bernth, 2004]; this would also address issues of scope.

6.1.3 Subordinate conjunctions

Subordinate conjunctions conjoin two clauses, and are hence intrinsically higher-order.
In example (39), *although* form a relation between *travel* and *navigate*.

```
(39) a. Although balloons travel with the wind, a pilot can navigate.
     b. although#7  < ((although#7 travel#4V navigate#5V))
        balloon#1G  < ((travel#4V balloon#1G u u) (card balloon#1G plur))
        navigate#5V < ((although#7 travel#4V navigate#5V)
                       (navigate#5V pilot#3G u u))
        pilot#3G    < ((navigate#5V pilot#3G u u) (card pilot#3G sing))
        travel#4V   < ((although#7 travel#4V navigate#5V) (travel#4V balloon#1G u u)
                       (with#6P wind#2 travel#4V))
        wind#2      < ((with#6P wind#2 travel#4V))
        with#6P     < ((with#6P wind#2 travel#4V))
```

6.1.4 Verbs with *that*-complements

Verbs with *that*-complements such as *say*, *believe*, and *hope* take a whole *that*-clause as a complement; this makes them higher-order. A special problem for many of these verbs is that they may introduce intensional contexts, an issue that we are not currently attempting to give a real treatment of.
Example (15), given earlier in Section 5.2, illustrates the treatment of a verb with a *that*-complement, *believe*.

6.2 "Lower/medium" higher-order entitities

On a somewhat higher level than the entities described in Section 6.1, we have the demonstrative pronouns *this* and *that*, which are most commonly used to refer to higher-order entitities.[10]
In example (40) *This* in the second sentence has as its antecedent *the lower part can be filled with hot air from a gas burner*. The information in the second sentence, combined with reasoning, makes this clear.

> (40) A modern Rozier is kept aloft by helium gas, but to add extra lift, the balloon is compartmentalized and the lower part can be filled with hot air from a gas burner. *This* enables the balloon to maintain altitude at night, when cold air and the lack of sunlight cools the helium and lessens its lifting power.

In example (41), *That* in the last sentence has as antecedent at least *Steve will have to stay at around 20,000 feet*. This is followed by an explanation of why staying at this altitude would cause Steve to slow down, and that part of the sentence is probably not included in the antecedent. But, obviously, these cases are difficult and care must be taken.

> (41) ''Because his capsule is not pressurized," Branson said in an interview, "Steve will have to stay at around 20,000 feet, which is usually well below the core of the jet stream. *That* could slow him down a lot.

6.3 "Higher" higher-order entitities

Paragraph topics and document summaries are entities of the highest order. Paragraph topics can be considered "mini summaries" and hence a kind of summarization. Document summarization has two major approaches; one consists of trying to identify important sentences and piecing together a coherent summary from those. The important sentences are often identified by statistical means. The other major approach is to convert the input text to a semantic representation and generate a summary from that.
Euphoria allows for a combination of these two approaches. Obviously, there is a semantic representation available. Furthermore, a statistical component is available in the frequency count of the entities.
Hence, a first approach could be to identify the most frequent entity and produce a summary from the EOLFs associated with it. But a better idea might be not just to pick the entity that has the highest *mention* frequency, but rather the entity (or entities) that have the most things said about them, as indicated by the EOLFs. We are currently experimenting with all these ideas, as well as synonym and other semantic relations of higher-order entities.

[10] Probably particularly *that* and probably not plural (*these* and *those*).

6.4 Compound Events

Several different verb mentions may refer to the same actual event, but the converse can also be true. This is the case in syntactic constructions involving gapping, where a single verb mention actually covers two (or more) distinct events. The sentence in (42) is an example of this.
The single verb mention *cross* covers two distinct events, namely one that takes place in 1987, and another that takes place in 1991. In order to reflect this in the EOLFs, we introduce a special predicate *compound_event* to hook the refID for the relevant node in the parse tree to two other refIDs that are not directly reflected in the parse tree, but which we need in order to capture the idea of two events. In the example, the mention *cross* hence introduces the compound event *cross#17V* consisting of the simple events *cross#18V* and *cross#19V* as indicated by *(compound_event cross#17V cross#18V cross#19V)*. The two simple events correctly distinguish what is being crossed and when, as shown in the abbreviated version in (43).

```
(42) a. Branson and Lindstrand, who have set several ballooning records, were
        the first pilots of hot-air balloons to cross both the Atlantic Ocean,
        in 1987, and the Pacific, in 1991.
     b. Atlantic Ocean#11 < ((cross#18V pilot#9 Atlantic Ocean#11)
                             (and#12 Atlantic Ocean#11 Pacific#13))
        Branson#5         < ((and#6 Branson#5 Lindstrand#7))
        Lindstrand#7      < ((and#6 Branson#5 Lindstrand#7))
        Pacific#13        < ((cross#19V pilot#9 Pacific#13)
                             (and#12 Atlantic Ocean#11 Pacific#13)
        and#12            < ((and#12 Atlantic Ocean#11 Pacific#13))
        and#6             < ((and#6 Branson#5 Lindstrand#7)
                             (set#14V and#6 several  u) (card and#6 plur)
                             (pilot#9 and#6))
        balloon#10G       < ((of#16P balloon#10G pilot#9) (hot-air#3A balloon#10G))
        balloon#15V       < ((balloon#15V  record#8G  u))
        both#1Adv         < ()
        cross#17V         < ((compound_event cross#17V cross#18V cross#19V))
        cross#18V         < ((cross#18V pilot#9 Atlantic Ocean#11)
                             (time year#1987 cross#18V)
                             (compound_event cross#17V cross#18V cross#19V))
        cross#19V         < ((cross#19V pilot#9 Pacific#13)
                             (time year#1991 cross#19V)
                             (compound_event cross#17V cross#18V cross#19V))
        first#2A          < ((first#2A pilot#9))
        hot-air#3A        < ((hot-air#3A balloon#10G))
        of#16P            < ((of#16P balloon#10G pilot#9))
        pilot#9           < ((of#16P balloon#10G pilot#9) (pilot#9 and#6)
                             (first#2A pilot#9)
                             (cross#18V pilot#9 Atlantic Ocean#11)
                             (cross#19V pilot#9 Pacific#13))
        record#8G         < ((balloon#15V  record#8G  u) (card record#8G plur))
        set#14V           < ((set#14V and#6 several  u))
        year#1987         < ((time year#1987 cross#18V))
        year#1991         < ((time year#1991 cross#19V))
```

```
(43) (cross#18V pilot#9 Atlantic Ocean#11)
     (cross#19V pilot#9 Pacific#13)
     (time year#1987 cross#18V)
     (time year#1991 cross#19V)
```

7 Future Work

There are obviously many other areas that could be addressed in the semantic analysis of a discourse. Some of these we have already mentioned in the text, but we would like to mention a few additional important areas in this section. The topics are by no means to be considered exhaustive.
Section 7.1 discusses the issue of word sense disambiguation, which is high on the list of further disambiguation. In Section 7.2 we touch on further work on temporal issues. The concept of *cohesion* also seems worth exploring, as mentioned in Section 7.3. Finally, how to evaluate the correctness of the semantic analysis that Euphoria produces is also an important issue.

7.1 Word sense disambiguation

The EOLFs described above do not take *word senses* into account. Disambiguating word senses is an important aspect of creating a disambiguated semantic analysis.
ESG is being enhanced with word senses [McCord, 2004], and, given that Euphoria is built on top of ESG, it seems natural to take advantage of the word senses produced by ESG, and incorporate them into the EOLFs. To accommodate the word senses, the EOLFs will be expanded as explained in this section.
Each predication in an EOLF is of the form

(44) $(P\ e\ arg_1\ arg_2\ \ldots arg_n)$

where P is the predicate of the predication, and where e, arg_1, ..., arg_n, the arguments of P, are all EEs. The predicate P will normally be a word sense (possibly of a multiword), but might more generally be a predicate concept in the ontology that does not correspond to any word sense. The first argument e of P, which we call the main entity argument of P, is the central entity that P predicates about, and arg_1, ..., arg_n are additional arguments of P (n may be 0). For example, if P is the usual sense of the verb "see", then e would be an event of seeing, with arg_1 as the subject, arg_2 as the object, and $n = 2$.
Example (45) illustrates how the word senses will be marked in the EOLFs. Each word sense predicate is represented by a citation form suffixed with an appropriate sense number.[11]

[11] The sense numbers in the example are just made up for illustration.

```
(45) a. The plane was flown by a new pilot.  The aviator navigated it carefully
        through the clouds.
     b. cloud#4     < ((through1 through#6 cloud#4 plane#1) (cloud1 cloud#4))
        carefully#7 < ((carefully1 carefully#7 navigate#5))
        fly#3       < ((fly2 fly#3 pilot#2 plane#1 u u))
        navigate#5  < ((navigate1 navigate#5 pilot#2 plane#1 u through#6)
                       (carefully1 carefully#7 navigate#5))
        new#8       < ((new3 new#8 pilot#2))
        pilot#2     < ((fly2 fly#3 pilot#2 plane#1 u u) (card pilot#2 sing)
                       (navigate1 navigate#5 pilot#2 plane#1 u through#6)
                       (new3 new#8 pilot#2) (pilot1 pilot#2) (aviator1 pilot#2))
        plane#1     < ((fly2 fly#3 pilot#2 plane#1 u u) (card plane#1 sing)
                       (navigate1 navigate#5V pilot#2 plane#1 u through#6)
                       (through1 through#6 cloud#4 plane#1) (plane1 plane#1))
        through#6   < ((navigate1 navigate#5 pilot#2 plane#1 u through#6)
                       (through1 through#6 cloud#4 plane#1))
```

7.2 Temporal relations

More work on temporal issues is clearly needed. This includes not only identification and resolution of further time expressions, but also work on taking the verb tenses and temporal subordinating conjunctions into consideration.

7.3 Cohesion

For identifying higher-order relations, the concept of *cohesion* [Halliday and Hasan, 1976] should be explored. This includes the importance of *how* a statement is made. The same "basic statement" can be expressed in a multitude of ways. Whereas this has been largely ignored in this report, it is the author's firm belief that there is a reason the speaker or author chose a specific way of saying what is said. Probably it is not viable in a practical system intended for applications such as inference to provide different EOLFs for different variations of the same basic statement–this would be more or less equivalent to inference on the parse structure[12]–but the *how* of the natural language may well feed into the semantic analysis on a higher level, e.g. focus could be helpful in determining paragraph topic etc.

7.4 Evaluation schemes

There are obviously many dimensions along which one could score the adequacy of the produced semantic analysis, and these should be explored.

[12]This idea has actually been explored in a Slot Grammar framework; see [Bernth *et al.*, 1992].

References

[Barwise and Perry, 1983] Jon Barwise and John Perry. *Situations and Attitudes*. MIT Press, New York, NY, 1983.

[Bernth and McCord, 2003] Arendse Bernth and Michael C. McCord. A hybrid approach to deriving selectional preferences. In *Proceedings of MT Summit IX*, pages 9–15, New Orleans, 2003.

[Bernth *et al.*, 1992] Arendse Bernth, Shalom Lappin, Michael C. McCord, and Wlodek Zadrozny. Natural language processing in a Slot Grammar framework. *International Journal on Artificial Intelligence Tools*, 1(2):229–277, 1992.

[Bernth, 1997] Arendse Bernth. EasyEnglish: A tool for improving document quality. In *Fifth Conference on Applied Natural Language Processing*, pages 159–165, Washington, DC, USA, 1997. Association for Computational Linguistics.

[Bernth, 1998] Arendse Bernth. EasyEnglish: Addressing structural ambiguity. In David Farwell, Laurie Gerber, and Eduard Hovy, editors, *Machine Translation and the Information Soup, Third Conference of the Association for Machine Translation in the Americas*, number 1529 in Lecture Notes in Artificial Intelligence, pages 164–173, Langhorne, PA, USA, 1998. Association for Machine Translation in the Americas, Springer.

[Bernth, 2002] Arendse Bernth. Euphoria – A reference resolution system for machine translation. Technical Report RC22627, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, November 2002.

[Davidson, 1967] Donald Davidson. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press, Pittsburgh, Pa., 1967.

[Donnellan, 1966] Keith Donnellan. Reference and definite descriptions. *Philosophical Review*, 75:281–304, 1966.

[Halliday and Hasan, 1976] M. A. K. Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman Group, 1976.

[Hobbs, 1985] Jerry R. Hobbs. Ontological promiscuity. In *Proc. of the 23rd ACL*, pages 61–69, Chicago, IL, 1985.

[Kamp, 1981] Hans Kamp. A theory of truth and semantic representation. In Groenendijk et al., editor, *Formal Methods In the Study of Language*, Mathematical Centre Tracts 135, pages 277–322. Amsterdam, 1981.

[McCord and Bernth, 2004] Michael C. McCord and Arendse Bernth. A metalogical theory of natural language semantics. Technical Report RC23252, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, January 2004. Revised version to appear in Linguistics and Philosophy.

[McCord, 1980] Michael C. McCord. Slot Grammars. *Computational Linguistics*, 6:31–43, 1980.

[McCord, 1987] Michael C. McCord. Natural language processing in prolog. In Adrian Walker, editor, *Knowledge Systems and Prolog*, pages 291–402. Addison-Wesley, Reading, MA, 1987.

[McCord, 1990] Michael C. McCord. Slot Grammar: A system for simpler construction of practical natural language grammars. In R. Studer, editor, *Natural Language and Logic: International Scientific Symposium, Lecture Notes in Computer Science*, pages 118–145. Springer Verlag, Berlin, 1990.

[McCord, 1993] Michael C. McCord. Heuristics for broad-coverage natural language parsing. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 127–132, Princeton, New Jersey, 1993. Morgan-Kaufmann.

[McCord, 2004] Michael C. McCord. Word sense disambiguation in a slot grammar framework. Technical Report RC 23397, IBM T.J. Watson Research Center, Yorktown Heights, NY, 2004.

[Menzel, 1986] C. Menzel. A complete, type-free "second-order" logic and its philosophical foundations. Technical Report CSLI Report 86-40, Stanford University, 1986.

[Quirk *et al.*, 1972] Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. *A Grammar of Contemporary English.* Longman, London, 1972.