Lexical and Conceptual Structure for Knowledge-Based Machine Translation

Sergei Nirenburg

Center for Machine Translation
Carnegie Mellon University

Christine Defrise

IRIDIA
Universite Libre de Bruxelles

The most salient feature of a knowledge-based machine translation system is its reliance on understanding the semantics (and, if possible, pragmatics) of the input text as the prerequisite for successful generation of its target language correlate. Meaning extraction is understood as representation in a specially designed artificial meaning representation language, *interlingua*. In the knowledge-based machine translation project at Carnegie Mellon University the result of meaning analysis is called *interlingua text* or ILT. An ILT has a propositional-semantic and a pragmatic component. The former is produced by instantiating tokens of concepts from an underlying ontological and domain model (called *concept lexicon* in the KBMT-89 machine translation system, an earlier system we developed, see Goodman, 1989). The latter is a set of structures determined through various lexical and syntactic clues in the input, largely independently from the domain model. The ILTs in Dionysus are represented using the knowledge representation language TAMERLAN (Nirenburg and Defrise, 1989). Dionysus is a combination of two research projects — the Diogenes natural language generation project and the Diana natural language understanding project.

In a nutshell, processing in Dionysus is as follows. An input text first undergoes morphological and syntactic analysis as a result of which a set of structures similar to f-structures of lexical-functional grammar is produced. Based on these structures, the semantic analyzer produces elements of text meaning, represented in TAMERLAN. A special interactive program (the *augmentor* can be (optionally) used to check the the completeness of the resulting meaning representation. The next step is text planning for realizing the text meaning obtained at previous stages in the target language. The text plan serves as input to the semantics-to-syntac mapping process whose result is a set of f-structures in the target language. The syntactic generator for the target language is next used to produce the final output, a text in the target language.

In KBMT-89, our previous system, lexicons that supported analysis and generation were different. The central static knowledge source responsible for producing ILTs was the *analysis lexicon*. The lexical-semantic needs of target language generation are served by the generation lexicon. The structures of the entry in these lexicons is to a large degree similar. In our current systems we decided, therefore, to merge the two knowledge sources into a single lexicon. Some of the information in the entries of this new lexicon is used during analysis or generation only. But a large portion is used in both types of processing. The lexicon contains orthographic, morphological, syntactic, semantic and pragmatic information. It also includes information about the mapping of syntactic dependencies and features into their semantic counterparts. In their semantics zone, the lexicon entries provide a) mappings between word senses and elements of the ontology and domain model (for open-class lexical items) and b) mappings between word senses and operations on the emerging text meaning representation in TAMERLAN (for closed-class items). These mappings are not always simple and direct. In a large number of cases a word sense is mapped into an ontological concept which represents a meaning that is somewhat distinct from that of a word sense. Therefore, the lexicon entries contain "meaning patterns" that specify on which property values this word sense is different from the description of the word sense into which it is mapped. This flexibility is essential for making the knowledge acquisition task feasible.

In this paper we illustrate the conceptual and lexical structures used in Dionysus.

1. Ontology.

A theoretically sound model of the world, an *ontology*, provides uniform definitions of basic semantic categories (such as objects, event-types, relations, properties, episodes, and many more) that become the building blocks for descriptions of particular domains and the creation of machine-tractable lexicons for comprehensive natural language processing. We believe that an optimum way of organizing this world model is as a huge multiply interconnected network of ontological units, on which theoretically sound storage, access and update procedures will be developed.

The requirements of knowledge representation extend above and beyond providing an adequate formalism in which to record knowledge, and adequate means of storage and retrieval. One has to specify the *contents*, the semantics of the knowledge units. And, if the goal is to build a semantic theory for natural language processing, an actual large model of the world must be produced (whereas for a general semantic theory a general mechanism for creating such a world view is usually deemed sufficient, especially if supported by several examples). This task is less formalizable than the syntactic aspects of knowledge representation, and this is one reason why relatively little progress has been made in AI with respect to ontological world modelling. Even today this area of scientific research remains, as it has been for over 2,500 years, within the purview of philosophy. While a number of important theories have been propounded in philosophical ontology, we believe that it is necessary to reformulate the goals and methodology of this inquiry, similarly to the way the finite time/space constraints of practical computation changed the style and attitudes of certain areas of discrete mathematics, which gave birth to the theory of computation.

An ontological model must define a large set of generally applicable categories for world description. Among the types of such categories are:

- Perceptual and common sense categories necessary for an intelligent agent to interact with, manipulate and refer to states of the outside world.
- Categories for encoding interagent knowledge which includes one's own as well as other agents' intentions, plans, actions and beliefs.
- Categories that help describe metaknowledge (i.e., knowledge about knowledge and its manipulation, including rules of behavior and heuristics for constraining search spaces in various processor components).
- Means of encoding categories generated through the application of the above inference knowledge to the contents of an agent's world model (see articles in Brachman and Levesque, 1985).

The choice of categories is not a straightforward task, as anyone who has tried realistic-scale world description knows all too well. Here are some examples of the issues encountered in such an undertaking (taken from Gates et al., 1989):

Which of the set of attributes pertinent to a certain concept should be singled out as *concept-forming* and thus have named nodes in the conceptual network corresponding to them, and which other ones should be accessible only through the concept of which they are properties? As an example consider whether you would further subdivide the class *vehicle* into *water-vehicle, land-vehicle, air-vehicle* or rather into *engine-vehicle, animal-propelled-vehicle, gravity-propelled-vehicle*; or maybe into *cargo-vehicle, passenger-vehicle, toy-vehicle, mixed-cargo-and-passenger-vehicle*? Or maybe it is preferable to have a large number of small classes, such as, for instance, *water-passenger-animal-propelled-vehicle*, of which, for instance, a rowboat will be a member?

Which entities should be considered *objects* and which ones, *relations*? Should we interpret a cable connecting a computer and a terminal as a *relation*? Or should we rather define it as a *physical-object* and then specify its typical *role* in the static episode or 'scene' involving the above three objects? Should one differentiate between *relations* (links between ontological concepts) and *attributes* (mappings from ontological concepts into symbolic or numerical value sets)? Or rather define attributes as one-place relations? Is it a good idea to introduce the ontological category of *attribute value set* with its members being primitive unstructured meanings (such as the various scalars and other, unordered, sets of properties)? Or is it better to define them as full-fledged ontological concepts, even though a vast majority of relations defined in the ontology would not be applicable to them (such a list will include case relations, partonomy, ownership, causals, etc.)?

As an example, should we represent colors *symbolically*, as, say *red, blue*, etc. or should we rather define them through their spectrum wavelengths, position on the white/black scale and brightness? How should we treat *sets* of values? Should we represent *The three musketeers* as one concept or a set of three? What about *The Wolverhampton Wanderers*? What's an acceptable way of representing *complex causal chains*? How does one represent a concept like *toy gun*? Is it a gun? Or a toy? Or none of the above? Or is it maybe the influence of natural language and a peculiar choice of meaning realization on the part of the producer that poses this problem – maybe we don't *need* to represent this concept at all?

In designing and implementing an actual model of the intelligent agent, we must, for any given level of detail, provide concrete answers to these questions. We have used the knowledge acquisition and maintenance system ONTOS (see Nirenburg et al., 1988) to produce several prototype ontological models. Figures 1 and 2 show several subnetworks the ontology developed for and used in the KBMT-89 machine translation project. These displays already illustrate answers to some of the above questions. The graphics browser of ONTOS facilitates fast overview and navigation in the ontological model. But this model is, in fact, much more than a set of symbols (frame names) connected through *is-a* and *part-of* links.

Figure 3 illustrates the actual content of some of the nodes in this network.

The knowledge required in a model of an intelligent agent includes not only an ontological world model, as sketched above, but also records of past experiences, both actually perceived and reported. The *lingua mentalis* equivalent of a text is an *episode*, a unit of knowledge that encapsulates a particular experience of an intelligent agent, and which is typically represented as a temporally and causally ordered network of object and event instances.

The ontology and the episodes are sometimes discussed in terms of the contents of two different types of memory: semantic and episodic (e.g., Tulving, 1985). This distinction seems useful in computational modeling as well. In our knowledge base we represent, with a varying degree of specificity, both ontological

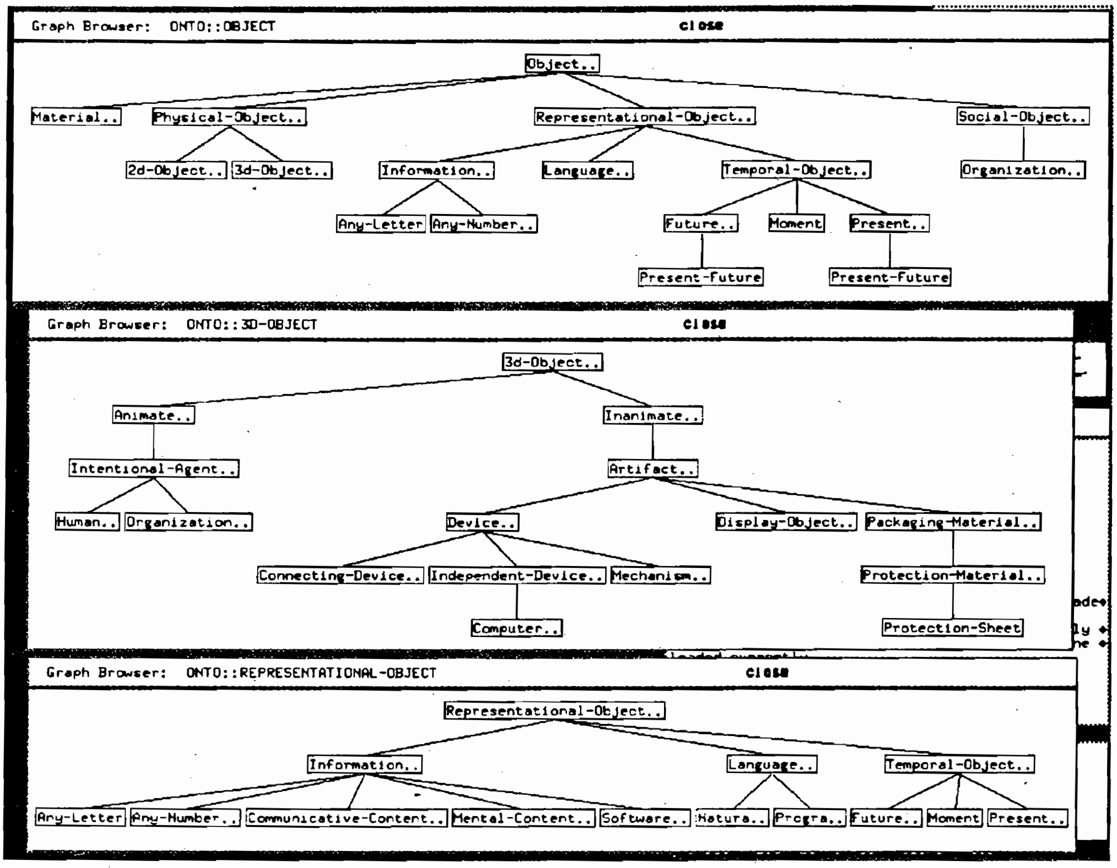


Figure 1: **Top-level Ontological Concepts.**

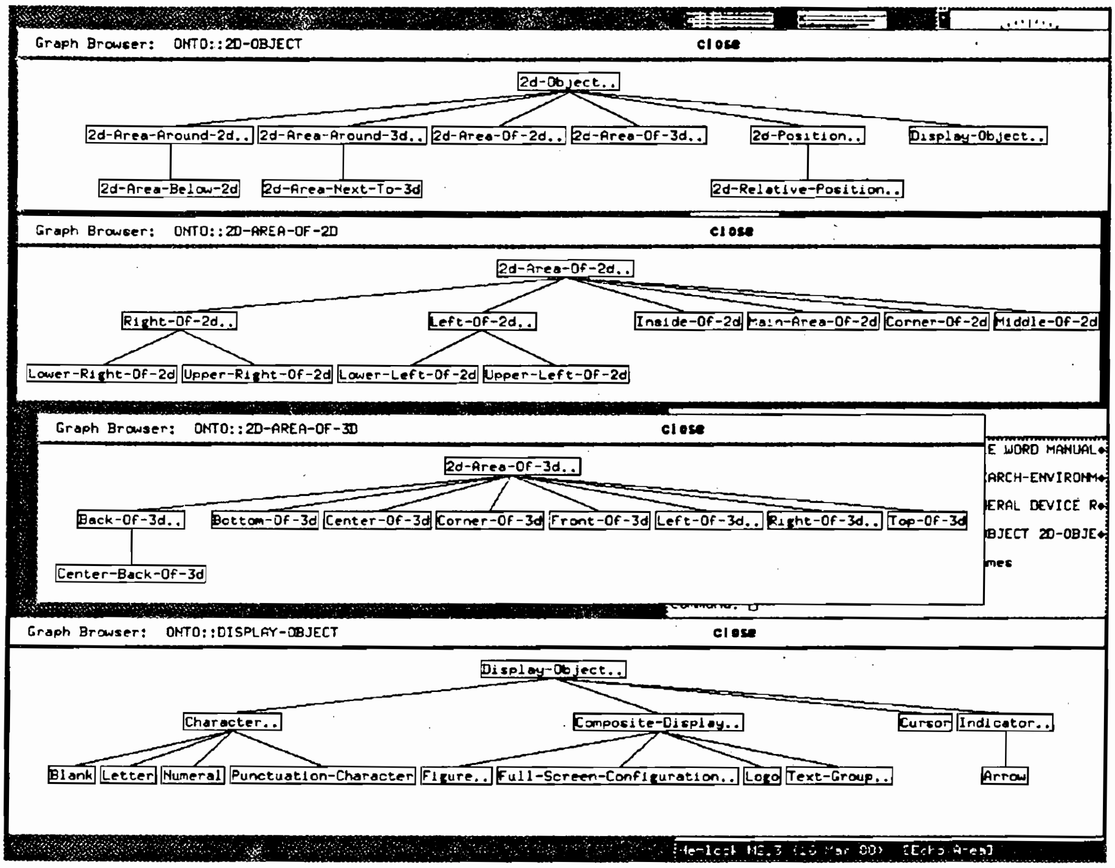


Figure 2: The `2d-object` hierarchy.

concepts and remembered instances of events and objects, which comprise the episodic memory.

Episodes are indexed through the type they correspond to and can be interrelated on temporal, causal and other links. The participant roles in the episodes can be either instantiations of object and event types in the semantic memory or references to existing named instances, stored outside semantic memory, but having links to their corresponding types (see Figure 4). The figure illustrates the typology of structures comprising the world model of an intelligent agent. The basic ontological world model is augmented (for the purposes of specific processing types, such as analogical reasoning) with a repository of the intelligent system's experiential knowledge. Our system must satisfy the knowledge representation needs of such a repository and abundantly cross-index it with the resident ontology. The presence of a systematic representation and indexing method for episodic knowledge is not only necessary for processing natural language but is also an enablement condition for case-based reasoning (see, e.g., the contributions in Kolodner and Riesbeck, 1986) and analogical inference (Carbonell, 1983).

2. Representing Text Meaning

A general description of TAMERLAN see in Nirenburg and Defrise, 1989 and *forthcoming*. In order to use TAMERLAN in an actual application, a further formalization is needed, in terms of a particular knowledge representation system. At the Center for Machine Translation of Carnegie Mellon University we used the FrameKit system (Nyberg, 1988).

We will illustrate the actual format of TAMERLAN text through a sample representation. The natural language text we will use in this illustration is a fragment of an advertisement published in *The Daily Hampshire Gazette*, Northampton, MA on April 26, 1985.

> Drop by your old favorite Dunkin' Donuts shop and you'll not only find fresh donuts made by hand, fresh Munchkins donut hole treats, the delicious smell of fresh-brewed coffee, and more. You'll also find a fresh new Dunkin' Donuts shop.

In the FrameKit intepretation, a TAMERLAN text is a directed graph rooted at the *text* frame, whose nodes are frame identifiers or terminal symbols (slot values), and whose arcs are slot names.

Prefixes on symbols in the TAMERLAN representation have the following meanings:

& A symbolic constant, a member of a value set defined in the ontology as the range of an attribute.

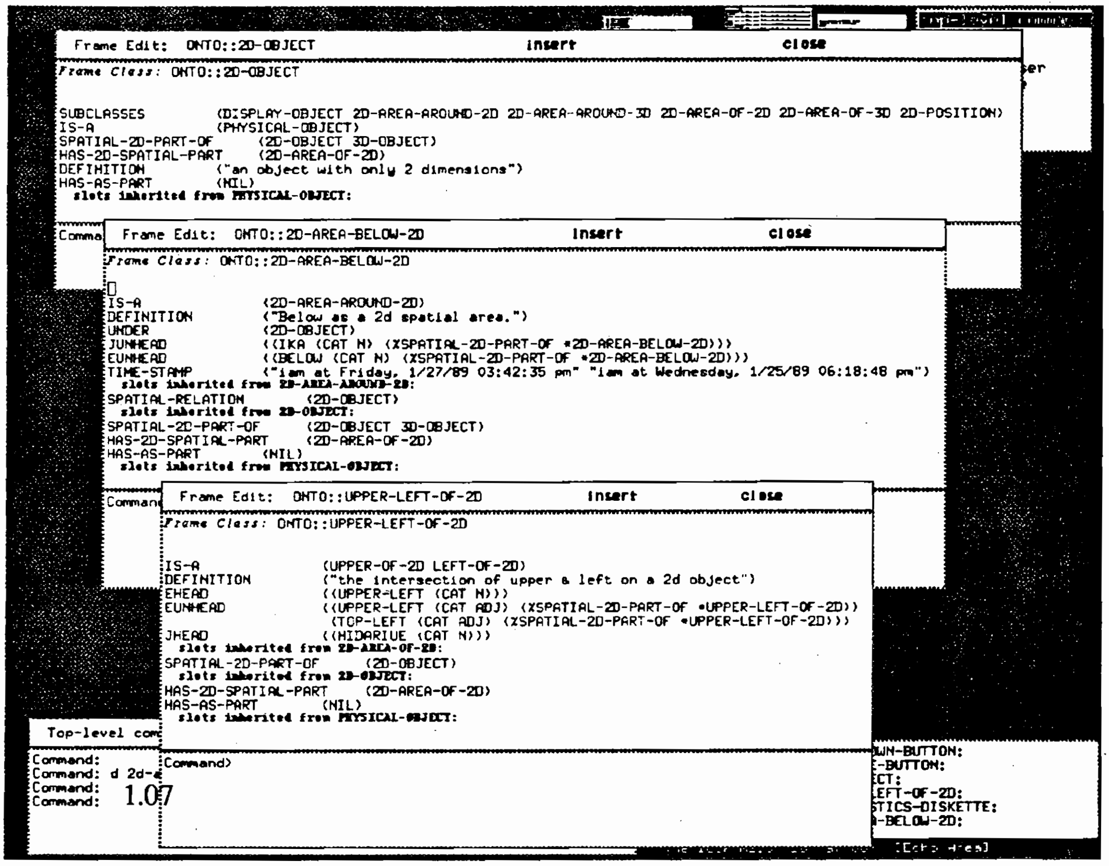
% An instantiated ontological concept. Note that the TAMERLAN syntactic structure identifier tokens (*text, clause, relation, attitude*) are not prefixed with "%" since they are not part of the ontology.

%% A *generic* instance of an ontological concept, used to represent set elements and other similar entities to which one doesn't individually refer.

$ A "remembered" instance, e.g., "John Kennedy."

* A concept from the ontology.

<symbol> A special variable.



```
Frame Edit:  ONTO::2D-OBJECT                    insert                  close
Frame Class: ONTO::2D-OBJECT

SUBCLASSES          (DISPLAY-OBJECT 2D-AREA-AROUND-2D 2D-AREA-AROUND-3D 2D-AREA-OF-2D 2D-AREA-OF-3D 2D-POSITION)
IS-A                (PHYSICAL-OBJECT)
SPATIAL-2D-PART-OF      (2D-OBJECT 3D-OBJECT)
HAS-2D-SPATIAL-PART     (2D-AREA-OF-2D)
DEFINITION          ("an object with only 2 dimensions")
HAS-AS-PART         (NIL)
  slots inherited from PHYSICAL-OBJECT:
```

```
Frame Edit:  ONTO::2D-AREA-BELOW-2D             insert                  close
Frame Class: ONTO::2D-AREA-BELOW-2D

IS-A                (2D-AREA-AROUND-2D)
DEFINITION          ("Below as a 2d spatial area.")
UNDER               (2D-OBJECT)
JUNHEAD             ((IKA (CAT N) (XSPATIAL-2D-PART-OF *2D-AREA-BELOW-2D)))
EUNHEAD             ((BELOW (CAT N) (XSPATIAL-2D-PART-OF *2D-AREA-BELOW-2D)))
TIME-STAMP          ("iam at Friday, 1/27/89 03:42:35 pm" "iam at Wednesday, 1/25/89 06:18:48 pm")
  slots inherited from 2D-AREA-AROUND-2D:
SPATIAL-RELATION        (2D-OBJECT)
  slots inherited from 2D-OBJECT:
SPATIAL-2D-PART-OF      (2D-OBJECT 3D-OBJECT)
HAS-2D-SPATIAL-PART     (2D-AREA-OF-2D)
HAS-AS-PART         (NIL)
  slots inherited from PHYSICAL-OBJECT:
```

```
Frame Edit:  ONTO::UPPER-LEFT-OF-2D             insert                  close
Frame Class: ONTO::UPPER-LEFT-OF-2D

IS-A                (UPPER-OF-2D LEFT-OF-2D)
DEFINITION          ("the intersection of upper & left on a 2d object")
EHEAD               ((UPPER-LEFT (CAT N)))
EUNHEAD             ((UPPER-LEFT (CAT ADJ) (XSPATIAL-2D-PART-OF *UPPER-LEFT-OF-2D))
                     (TOP-LEFT (CAT ADJ) (XSPATIAL-2D-PART-OF *UPPER-LEFT-OF-2D)))
JHEAD               ((HIDARIUE (CAT N)))
  slots inherited from 2D-AREA-OF-2D:
SPATIAL-2D-PART-OF      (2D-OBJECT)
  slots inherited from 2D-OBJECT:
HAS-2D-SPATIAL-PART     (2D-AREA-OF-2D)
HAS-AS-PART         (NIL)
  slots inherited from PHYSICAL-OBJECT:

Command>
```

```
Top-level com
Command:
Command: d 2d-a
Command:
Command:
```

```
WN-BUTTON:
-BUTTON:
CT:
EFT-OF-2D:
TICS-DISKETTE:
-BELOW-2D:
[Echo Area]
```

Figure 3: The `2d-object` frames.

```
(make-frame text_1
        (clauses (value clause_1 clause_2
                            clause_3 clause_4
                            clause_5 clause_6 clause_7 ))
        (relations (value relation_1 relation_2
                              relation_3 relation_4
                              relation_5 relation_6
                              relation_7 relation_8))
         (attitudes (value attitude_1 attitude_2 attitude_3))
         (producer-intentions (value producer-intention_1)))
```

The text frame serves as the index for all the clauses and relations in it. This particular text has seven clauses and thirteen relations.

Clause_1 represents the meaning of "Visit your favorite Dunkin' Donuts shop!"

```
(make-frame clause_1
        (head (value %visit_1))
        (aspect (duration prolonged)
                (phase begin)
                (iteration  1))
        (time (value time_2)))

(make-frame %visit_1
        (is-token-of (value *visit))
        (agent (value *consumer*))
        (destination (value %shop_1))
```

The special variables *producer* and *consumer* represent models of the speaker/writer/author and hearer/reader, respectively. Aspectual properties are represented as values on three properties — *duration, phase* (that is, beginning, continuation or end) and *iteration*. A detailed description of our microtheory of aspect see Nirenburg and Pustejovsky, 1988. In this example, we stress the fact that a visit is not an instantaneous event and that the phasal meaning is inchoative. The properties *agent, experiencer, destination* and *theme* are case roles.

```
(make-frame %shop_1
            (is-token-of (value *shop))
            (part-of (value $dunkin-donuts)))

(make-frame %shop_1_1
    (time-token-of (value %shop_1))      ; I seem to remember that we decided to
                                         ; call this a "time-token", and to
                                         ; give the "time slot" two facets
                                         ; "since" and "until".
     (time (value (until time_4))))

(make-frame %shop_1_2
     (time-token-of (value %shop_1))
     (time (value (since time_4))))
```

Object instances in TAMERLAN are represented as trees of time-stamped frames. This is required to be able to refer to 1) an object instance when some of its properties change with time and 2) to previously held beliefs about some properties of this instance. In our example, *shop_1* is the root of the tree, the timeless reference to a particular object instance. Its timed subinstances, *shop_1_1* and *shop_1_2* represent the shop before and after remodeling. However, if after another visit it will appear that the shop was, in fact, *not* remodeled after all, then the representation of the instance will consists of the following subinstances:

shop_1_1_1 — the consumer's remembered belief, after the first visit, about the state of the shop before the first visit;

shop_1_2_1 — the remembered belief, after the first visit, about the state of the shop after the first visit;

shop_1_1_2 — the belief, after the second visit, about the state of the shop

In this example, the subinstance *shop_1_1_1* is identical to *shop_1_1_2*. At the moment, for the sake of simplicity, we disregard the representations of producer beliefs about object instance property changes. In future implementations, however, we expect to introduce not only producer beliefs but also an indication of the strength of these beliefs, which would help in processing heuristic preference rules in both analysis and generation.

In the current example, both the time-stamped instances (shop_1_1, see relation_7, and shop_1_2, see %involuntary-visual-event _1 and relation_8) have a *relative* time constraint, specified in relation_7 and relation_8.

The meaning of *find* will be understood as a perceptual-event. Since the consumer will not deliberately look for the things that he will perceive, it will be classified as an involuntary-perceptual-event. In fact, in the underlying ontology, the perceptual action subnetwork has the following form:

```
perceptual-event
   voluntary-perceptual-event
         voluntary-visual-event      (look)
         voluntary-auditory-event    (listen)
         voluntary-tactile-event     (touch-1,  run fingers across)
         voluntary-gustatory-event   (taste-1)
         voluntary-olfactory-event   (sniff,  smell-1)
   involuntary-perceptual-event
         involuntary-visual-event     (see)
         involuntary-auditory-event    (hear)
         involuntary-tactile-event     (touch-2)
         involuntary-gustatory-event   (taste-2)
         involuntary-olfactory-event   (smell-2)
```

The meaning of "the consumer will perceive a) donuts, b) Munchkins, c) the smell of coffee, d) a new shop and e) additional things" is represented in TAMERLAN using as many clauses as there are instances of perception involved. Thus, the doughnuts, the Munchkins and the new shop are understood as having been involuntary perceived *visually* (this is, in fact, the default mode of perception), coffee as involuntary perceived *olfactorily*, and the meaning of "other things," which is a gloss of the meaning of *more* in the input, is realized as an instance of involuntary-perceptual-event because it is not specified what type of perception may be involved.

Clause_2 represents the meaning of "the consumer will perceive donuts."

```
(make-frame clause_2
      (head (value %involuntary-perceptual-event_1))
      (aspect
        (phase begin)
        (duration prolonged)
        (iteration 1))
        (time (value time_3)))

   (make-frame %involuntary-perceptual-event_1
      (is-token-of (value *involuntary-perceptual-event))
      (experiencer (value *consumer*))
      (theme (value %set_1)))

(make-frame %set_1
      (is-token-of (value *set))
      (type (value conjunctive))
      (element (value %%doughnut-1)))

(make-frame %%doughnut-1
      (is-token-of (value *doughnut))
      (age (value (< 0.1))))
```

```
The age of doughnuts is a range of values on a scale.  The ''age <
0.1'' slot expresses the fact that the doughnuts are fresh. Note that
it is necessary to mark an instance of {\tt \%\%doughnut}, {\tt
\%\%doughnut-1}, because of the constraint (the age) which is true
only of this group of doughnuts. (See below the similar treatment of
{\tt \%\%munchkin}.)
```

Multiple fillers of the `value` facet of a FrameKit frame are interpreted as conjoined elements. Sets in TAMERLAN are of two kinds — single element-type sets, as in the text about doughnuts, or enumerated sets, in which elements are overtly listed, as in the following example.

```
(make-frame %set_x
     (is-token-of (value *enumerated-set))
     (elements (value %element_1 %element_2 ...)))
```

Note that elements in the representation above can, naturally, be sets in their own right.

In this example, one representational property of `*set`, its `cardinality`, is not shown. The reason for this is that the cardinality of none of the sets used in our example is known.

Clause_3 represents the meaning of "the consumer will perceive munchkins."

```
(make-frame clause_3
      (head (value %involuntary-perceptual-event_2))
      (aspect
        (phase begin)
        (duration prolonged)
        (iteration 1))
      (time (value time_3)))

 (make-frame %involuntary-perceptual-event_2
      (is-token-of (value *involuntary-perceptual-event))
      (experiencer (value *consumer*))
```

```
     (theme (value %set_2 )))

(make-frame %set_2
     (is-token-of (value *set))
     (type (value conjunctive))
     (element (value %%munchkin-1)))

(make-frame %%munchkin-1
     (is-token-of (value *munchkin))
     (age (value (< 0.1))))
```

Clause_4 represents the meaning of "The consumer will find a new shop".

```
(make-frame clause_4
     (head (value %involuntary-perceptual-event_3))
     (aspect
      (phase begin)
      (duration prolonged)
      (iteration 1))
      (time (value time_3)))

(make-frame %involuntary-perceptual-event_3
     (is-token-of (value *involuntary-perceptual-event))
     (experiencer (value *consumer*))
     (theme (value %shop_1_2 )))
```

Clause_5 represents the meaning of "the consumer will perceive smell of coffee."

```
(make-frame clause_5
     (head (value %involuntary-olfactory-event_1))
     (aspect
      (phase begin)
      (duration prolonged)
      (iteration 1))
       (time (value time_3)))

(make-frame %involuntary-olfactory-event_1
     (is-token-of (value *involuntary-olfactory-event))
     (experiencer (value *consumer*))
     (theme (value %coffee_1)))

(make-frame %coffee_1
     (is-token-of (value *coffee))
     (age (value (< 0.1))))
```

Clause_6 represents the meaning of "the consumer will perceive things." At this point the representation does not specify that these "things" do not include those mentioned earlier in the text. This information may not be needed in some applications, such as, for instance, machine translation, where no reasoning is expected that would involve the determination of what these "things" actually are. An extension of the TAMERLAN text will be needed for such applications where this information may be essential, such as, for instance, question answering systems.

```
(make-frame clause_6
     (head (value %involuntary-perceptual-event_4))
     (aspect
      (phase begin)
      (duration prolonged)
      (iteration 1))
      (time (value time_3)))

(make-frame %involuntary-perceptual-event_4
     (is-token-of (value *involuntary-perceptual-event))
     (experiencer (value *consumer*))
     (theme (value %set_3 )))

(make-frame %set_3
     (is-token-of (value *set))
     (type (value conjunctive))
     (element (value (set-difference ontosubtree(physical-object)
                                     (*doughnut *munchkin *coffee *shop)))))
```

`ontosubtree` is a function that returns a list of all concepts in the subtree(s) of its argument(s), which should be ontological concepts. The operator `set-difference` is defined in the usual way. The elements of the set %set_3, thus, are all ontological descendents of `physical-object` with the exception of the concepts `*doughnut`, `*munchkin`, `*coffee` and `*shop`. Intuitively, this means that additional things that one can see in the shop are all kinds of objects other than those mentioned in the text.

Clause_7 represents the meaning of "Doughnuts at Dunkin' Donuts are made by hand."

```
(make-frame clause_7
    (head (value %produce_1))
     (aspect
      (phase continue)
      (duration prolonged)
      (iteration 1))
    (time (value *always*)))

(make-frame %produce_1
    (is-token-of (value *produce))
    (theme (value %set_1))
    (production-mode (value &manual)))
```

Clause_8 represents "The shop has been recently remodeled."

```
(make-frame clause_8
    (head (value %remodel_1))
    (aspect
       (phase end)
       (duration prolonged)
       (iteration 1))
    (time (value time_4)))

(make-frame %remodel_1
     (is-token-of (value *remodel))
     (theme (value %shop_1_1)))


(make-frame relation_1
     (type (value domain-conditional))
     (first (value %visit_1))
     (second (value  %involuntary-perceptual-event_1)))

(make-frame relation_2
     (type (value domain-conditional))
     (first (value  %visit_1))
     (second (value  %involuntary-perceptual-event_2)))

(make-frame relation_3
    (type (value domain-conditional))
    (first (value %visit_1))
    (second  (value %involuntary-perceptual-event_3)))

(make-frame relation_4
     (type (value domain-conditional))
     (first (value  %visit_1))
     (second (value %involuntary-olfactory-event_1)))

(make-frame relation_5
     (type (value domain-conditional))
     (first (value  %visit_1))
     (second (value %involuntary-perceptual-event_4)))
```

The above five relations represent the idea that it is only possible to perceive all the things in the new shop (including the new shop itself!) if one visits it.

```
(make-frame relation_6
     (type (value domain-temporal-during))
     (first (value  time_3))
     (second (value  time_2)))
```

The perception of Doughnuts, Munchkins, coffee, "other things" and the remodelled shop (time_3) occurs during the visit (time_2).

```
(make-frame relation_7
     (type (value domain-temporal-after))
     (relation-value (value 0.8))
     (first (value time_2))
     (second (value time_5)))
```

The positive attitude toward the shop existed long before the visit was made. (This is the realization of "*old* favorite.") The relation value is an estimate of the distance of the two events (making the visit and holding the attitude). The value 0.8 corresponds roughly to "large." (The value 1 means "infinite" distance.)

```
(make-frame relation_8
     (type (value intention-domain-temporal-after))
     (relation-value (value 0.2))
     (first (value time_1))
     (second (value time_4)))
```

The shop was remodeled not long before the statement was made. "Not long ago" is realized through `relation-value`.

```
(make-frame relation_9
     (type (value domain-conjunction-enumeration))
     (arguments  (value %involuntary-perceptual-event_1
                        %involuntary-perceptual-event_2
                        %involuntary-perceptual-event_3
                        %involuntary-olfactory-event_1
                        %involuntary-perceptual-event_4))
```

```
(make-frame relation_8
   (type (value intention-domain-temporal-after))
   (relation-value  (value 0.2))
   (first (value  time_2))
   (second (value time_3)))
```

Relation_10 represents the fact that visiting should start before perceiving

```
(make-frame attitude_1
    (type (value &evaluative))
    (attitude-value  (value 0.9))
    (scope (value @%visit_1.theme))
    (attributed-to  (value *consumer*))
            (time (value (since time_5))))
```

The presence of attitude_1 is the TAMERLAN way of realizing the meaning of "favorite." The value of the attributed-to slot realizes the meaning of "your." Epistemic attitudes to events, objects or properties are not overtly listed if their values are 1.

```
make-frame attitude_2
     (type (value &saliency))
     (attitude-value  (value 0.7))
     (scope (value %involuntary-perceptual-event_1
                   %involuntary-perceptual-event_2
                   %involuntary-perceptual-event_4
                   %involuntary-olfactory-event_1))
     (attributed-to  (value *producer*)))

(make-frame attitude_3
     (type (value &saliency))
     (attitude-value  (value 1))
     (scope (value %involuntary-perceptual-event_3  ))
     (attributed-to  (value *producer*)))
```

The hearer will expect to find fresh douhgnuts, fresh munchkins and fresh coffee in a Dunkin' Donuts shop. A redecorated shop will be unexpected.

```
(make-frame producer-intention_1
     (is-token-of (value *commissive-act))
     (scope (value relation_1 relation_2 relation_3 relation_4)))
```

The speech act performed by uttering the above text is a conditional request.

2.1. Selected Representation Decisions.

2.1.1. Representation of Modifiers.

If in the ontology the characteristic properties of an individual or a proposition type include the property that is expressed by a modifier, the meaning of the modifier will be expressed as a value of that property. Thus, if *color* is listed in the ontology as a characteristic property of the concept *car* (either directly or through inheritance — in this example, from *physical-object*) "a blue car" will be represented as

```
(make-frame %car5
  (is-token-of (value *car))
  (color (value &blue)))
```

If a modifier does not express a property defined for the ontological concept corresponding to its head, then it has to be represented in one of the following ways:

- as an attitude value: the meaning of *favorite* in "your favorite Dunkin' Donuts shop" is expressed through a consumer attitude towards the head of the phrase;
- as a separate clause: the meaning of "made by hand" is expressed through the entire `clause_3`;
- as a relation: all the relative temporal modifiers are expressed through temporal relations.

2.1.2. What Can Become the Head of a TAMERLAN Clause?

Heads of clauses in TAMERLAN can be a) event tokens or b) object tokens.

The examples above illustrate the former case. To illustrate the latter possibility, consider the following two examples. "The car is blue" will be represented in TAMERLAN as

```
(make-frame clause2
  (head (value %car6))
  (topic (new @%car6.color)))

(make-frame %car6
   (is-token-of (value *car))
   (color (value &blue)))
```

Here the meaning is pointing out a property of an object instance. There is no event involved. We realize this situation through assigning the object as the head of the clause and the property, as the contents of the "new" facet of the "topic" slot. Thus, if the object is listed with more properties (as in "The big car is blue") we will know which one is stressed.

The input "My neighbor John is a math teacher" will be represented as

```
(make-frame clause4
  (head (value %human8))
  (topic (new %teacher2)))

(make-frame %human8
   (is-token-of (value *human))
   (neighbor-of (value *producer*)) ;this property is the
   (name (value *John)))            ;only place in the ontology where
                                    ;the idea of ''being a neighbor''
                                    ;is covered
(make-frame %teacher2
   (is-token-of (value *teacher))
   (subject (value *mathematics)))

(make-frame relation1
   (type (value coreference))
   (from (value %human8))
   (to (value %teacher2)))
```

Examples such as the above are sometimes treated as special meanings of the verb *to be*., see Hirst, 1987, pp. 62ff for a discussion of representing the concept of "being." The representation of "My brother John is a teacher" in his semantic interpreter, Absity, will include three different concept instances for "John," "teacher" and "brother,"[1] and the statements (same-instance John teacher11) and (same-instance John brother2). Hirst, 1989, contains a more detailed analysis of the tratment of of being, predication and identity.

On a more general note with respect to reference, the sentence "The Iliad was written not by Homer but by another man with the same name" will be represented as

```
(make-frame clause14
  (head (value relation5))
  (topic (given (value relation5))
         (new (value attitude3.value))))

(make-frame $Homer                 ;a remembered instance
   (is-token-of (value *human)))   ;indexed by name

(make-frame %human4                ;representation of
   (is-token-of (value *human))    ;''another man'' --
   (name (value Homer)))           ;his name is Homer

(make-frame $Iliad                 ;The Iliad
   (is-token-of (value *book))     ;was written
   (author (value %human4)))       ;by this person;
                                   ;this is new information, and
                                   ;the format of the remembered instance
                                   ;$Iliad will be modified accordingly

(make-frame relation6              ;the remembered Homer
   (type (value coreference))      ;is coreferential with
   (from (value %human4))          ;the newly created Homer
   (to (value $Homer)))

(make-frame attitude3              ;but the producer attitude to
   (type (value epistemic))        ;the above relation is
```

[1] unless "brother" is defined as a property of "human"

```
    (value (value 0))              ;that it doesn't really hold
    (scope (value relation6)))
```

2.1.3. Representing Questions.

To represent special questions ("What is in the corner?") we need

```
(make-frame clause6
  (head (value %physical-object5))
  (producer-intention (value request-info))
  (topic (new %physical-object5.location)))

(make-frame %physical-object5
  (is-token-of (value *physical-object))
  (location (value *corner2)))
```

A problem: what if the next sentence is "A broom is in the corner" or "It is a broom" or "A broom?" This means that has to change, because now we know that it is not just any physical object but, indeed, a broom. Our solution is to introduce the new instance, broom as well as a new relation which records the coreferentiality of the two objects.

```
(make-frame %broom1
   (is-token-of (value *broom))
   (location (value %corner2)))

(make-frame relation9
   (type (value cofererence))
   (from (value %physical-object5))
   (to   (value %broom1)))
```

Now, we, just as unification-based grammarians, use paths and cross-references in order not to repeat information. Therefore, we would have to use only the more concrete (the latter) version of `%broom`. But then in generation we will have a difficult time distinguishing between generating the question "What is in the corner?" and "Is the thing in the corner a broom?" One way of avoiding this difficulty is time-stamping the individuals, as suggested above and illustrated in the example.

Yes-no questions ("Is the broom in the corner?") are represented as

```
(make-frame clause9
    (head (value role15))
    (producer-intention (value request-info))
    (topic (new role15.location)))

(make-frame role15
    (is-token-of (value broom))
    (location (value corner2)))
```

2.1.4. Problems with Choosing What to Instantiate.

One of the well-known difficulties in representation is the task of representing the meaning of the so-called intensional adjectives in phrases like "a fake gun," "an alleged criminal," "a toy tank," etc. The problem is that in a fake gun is not, in fact, a gun; however, it retains some of the properties of guns (e.g., shape). An alleged criminal may not be a criminal at all. Instead of instantiating a `*physical-object`, TAMERLAN instantiates a gun, but attaches an epistemic attitude that says that the producer is not sure about how "gun-like" it actually is, only that it is not a real gun. Therefore,

```
(make-frame %criminall                  ;there is a criminal
   (is-token-of (value *criminal)))

(make-frame %human31
   (is-token-of (value *human)))        ;and there is a human

(make-frame relation7                   ;who stand in a
   (type (value coreference))           ;coreference relation
   (from (value %criminall))
   (to (value %human31)))

(make-frame attitude3                   ;but producer is not
   (type (value epistemic))             ;confident that the
   (value (value < 1))                  ;coreference relation
   (scope (value relation7)))           ;is epistemically valid
```

2.1.5. Representing Time.

References to absolute times are listed in the "time" slots defined for event- and object-tokens. References to relative times are represented as temporal relations. The representation of time in DIANA / DIOGENES conforms to the following rules.

```
texpr ::= (C quant time)
| (U {time}+)
| time

time ::= (boolean {time}+)
| t-atom

t-atom  ::= (temp-op (t-arg | t-arg t-arg))

temp-op ::= during | at | after | before | while | between

t-arg ::= <an absolute time> | path

quant ::= some | all | ...

boolean ::= and | or
```

Temporal operators can take either point or interval arguments. The `at` operator with a point argument indicates a point in time; with an interval argument, it means that the event lasts the entire length of the interval. The `during` operator with an interval argument indicates that the time of the event- or object-instance is a proper subset of the argument interval, possibly having one of the same endpoints as the argument interval. It thereby subsumes Allen's (1983) during, starts, and finishes operators. The while operator is a notational shorthand: (while x) ≡ (or (during x) (at x)).

The and and or operators are used to combine constraints on defining *one* interval, not for combining multiple intervals. Thus "time" in the definition above necessarily represents the time interval over which one state/process/action occurred. So an or indicates that an event occurs during (time X or time Y). The and indicates that an event occurs during an interval of time which satisfies constraint X and constraint Y, and not something like "the event occurs at time X and then again at time Y").

In order to represent the latter situation, we introduce the union operator. The collection operator is like union in that it identifies multiple occurrences of the event in question, but it differs from the latter in that it does not enumerate the multiple occurrences, usually because information is lacking. For instance, for the sentence "In 1989 I often exercised on fridays and mondays" the time will be represented as (C many (and (during 1989) (or (during friday) (during monday)))).[2]

2.1.6. Representing Generic Senses.

The TAMERLAN convention for representing generic senses of concepts (as in, e.g.,

Running is moving on foot in such a way that at no time both feet touch the ground, or

The tiger lives in Asia),

is to use the 'instance' marker % appended to the concept name, but without an instance number. The properties of a generic instance are assumed to be exactly the same as those of the type concept.

3. Boundaries of Description.

In our work, we adopt the methodological attitude of developing the natural language processing functionalities in a breadth-first fashion. That is to say that, unlike many other projects, we do not tend to describe exhaustively a specific linguistic phenomenon (e.g., negation, anaphora, aspect, scope of quantifiers) or type of processing (e.g., text planning, lexical selection, syntactic realization) before proceeding to the next one (this approach can be considered depth-first). We prefer to go for a complete functioning system which contains all (or, in practice, most) of the above components and covers all (or most) of the above phenomena. It is clear that, at the beginning, the treatment of each (or most) of these components is incomplete, and not every phenomenon is described in sufficient detail. However, this methodology allows us to benefit from a complete experimentation environment and an open-ended architecture that facilitates the addition of knowledge to the system and its testing and debugging. At present we have a working prototype understanding, text planning and generation system with narrow coverage. Our current work is devoted to expanding the knowledge needed for achieving a deeper level of analysis of each of the linguistic phenomena covered in the system.

[2] Note that union is in fact a shorthand notational convenience for an equivalent collection expression, (U x y) ≡ (C all (or x y)).

4. The Lexicon

In this section we will illustrate our lexicon structure through a set of annotated examples.[3] The examples show only those senses of the corresponding lexemes that are used in the advertisement text used above.

```
(donut
    (make-frame
+donut-n1
(CAT (value n))
(STUFF
      (DEFN "pastry cooked in fat, usually in the
      shape of a ring or ball")
      (EXAMPLES "Dunkin’ Donuts produces more donuts
      than all other fast food outlets put together"))
(ORTH (VARIANTS doughnut))
(SYN
    (count +) (proper -))
(SEM
    (LEX-MAP
    (%doughnut)))))
```

The `stuff` zone in the definition contains human-oriented information and is not used by the system itself. The `orph` zone lists spelling variants. The `syn` zone lists paradigmatic syntactic features. The `sem` zone in the above example justs lists a simple lexical mapping of the meaning of this sense of *donut* into a corresponding ontological concept.

```
(find
 (make-frame
  +find-v1
  (CAT (value v))
  (STUFF
   (DEFN "to discover by chance, to come across")
   (EXAMPLES "drop by your old favorite Dunkin Donuts
    shop and you’ll not only find fresh donuts made by hand"
     "when I arrived home last night, I found a drunk sleeping
    on the porch/that a drunk was sleeping on the porch"))
  (MORPH
   (IRREG (*v+past* found) (*v+past-part* found)))
  (SYN-STRUC
    (*OR* ((root $var0)
   (subj (root $var1) (cat N))
   (obj  (root $var2) (cat N)))
  ((root $var0)
```

[3] The entries were constructed by Ingrid Meier. Boyan Onyshkevych helped in the design of the entry structure. A comprehensive description of the lexicon in the Diana/Diogenes project is given in Nirenburg et al., *in preparation*.

```
   (subj (root $var1) (cat N))
   (xcomp (root $var2)(cat V) (form pres-part)))
  ((root $var0)
   (subj (root $var1) (cat N))
   (comp (root $var2) (cat V) (form fin)))))
   (SEM
       (LEX-MAP
       (%involuntary-perceptual-event
   (experiencer (value ^$var1))
   (theme (value ^$var2))))))
```

The above entry demonstrates our way of recording inflectional irregularities. The `syn-struc` zone describes the subcategorization classes of the entry head. IN the entry above there are three subcategorization variants, all with different types of direct objects that *find* may take. The variables in the specifications are used for binding the values of arguments. In the `lex-map` slot of the `sem` zone these bindings help to determine to which syntactic entities the intensions (semantic interpretations) of the arguments correspond (The "^" prefix marks the intensions). Intuitively, the lexical mapping above says that the given sens of *find* is mapped in TAMERLAN as an instance of the *%involuntary-perceptual-event* ontological concept. Moreover, the semantic interpretation of whatever occupied the `subj` position in the f-structure should be assigned as the value of the `experiencer` thematic role in the above concpet instance, while the meaning of whatever occupied the `obj`, `xcomp` or `comp` position in the f-structure should be assigned as the value of the `theme` thematic role in the concept instance.

```
(drop
    (make-frame
+drop-v1
(CAT (value v))
(STUFF
      (DEFN "to visit a place")
      (EXAMPLES "drop by your old favorite Dunkin'
      Donuts shop"))
(SYN-STRUC
  ((root $var0)
   (subj ((root $var1) (cat n)))
   (obliques ((root $var2) (prep by)))))
(SEM
    (LEX-MAP
    (%visit
   (AGENT (value ^$var1))
   (THEME (value ^$var2)
  (sem *building)
  (relaxable-to *object)))))))
```

As can be seen from the above example, verbs with particles are treated in our lexicon through the same mechanism as particle-less verbs. The `lex-map` slot above says that a) the meaning of the head of the structure which fills the `obliques` f-structure slot carries a semantic constraint — that it must be an instance of a concept in the ontological subnetwork rooted at `*building` and b) that this constraint is relaxable in real text to the subnetwork rooted at `*object`. The relaxation statement is used to process metonymy — as in the sentence *Drop by the committee meeting.*

```
(delicious
    (make-frame
+delicious-adj1
(CAT (value adj))
(STUFF
      (DEFN "very pleasing to sense of taste or smell or sight")
      ;this DEFN may have to be refined, since saying that something
       ; looks or smells delicious means that it
; looks or smell AS IF IT WOULD taste
; delicious
      (EXAMPLES "delicious meal" "delicious smell"
  "the meal looks delicious"))
(SYN
    (attributive + -))
(SYN-STRUC
  ((root $var1)
   (cat n)
   (mods ((root $var0)))))
; pattern shown for attributive use only
(SEM
    (LEX-MAP
             (^$var1
              (instance-of (sem (*OR* *ingestible
                      *olfactory-attribute
      *gustatory-attribute
                                              *visual-attribute))))
    (ATTITUDE
(type (value evaluative))
(attitude-value (value 0.8))
(scope (value ^$var1))
(attributed-to (value *producer*)))))))
```

The meaning of *delicious* is a speaker attitude of type evaluative, with a high value on the zero-to-unity scale. The attitude can be toward a perception attribute, as specified in the constraint on the meaning of the noun that *delicious* modifies.

```
(by
   (make-frame
+by-prep1
(CAT (value prep))
      (DEFN "using the instrument of")
      (STUFF "made by hand" "designec by computer"
      "produced by machine"))
(SYN-STRUC
   ((root $var1)
    (cat n)
    (pp-adjunct ((root $var0)
(obj (root $var2)
     (cat n)))))
(SEM
     (LEX-MAP
     (^$var1
    (instance-of (sem *physical-event))
    (instrument (value ^$var2)
                (sem (*OR* *hand *artifact)))))))
```

This sense of the preposition *by* is specified in terms of constraints on a) the head of the NP inside the PP introduced by *by* and the head of the phrase to which the prepositional phrase is attached. The latter is constrained to an instance of `physical-event` (which can be realized by a verb or a noun). The former must be an instance of an entity in the ontological subnetwork of either `*hand` or `*artifact`. Moreover, the latter should play the thematic role of `instrument` in the latter.

```
(only
     (make-frame
+only-adv1
(CAT (value adv))
(STUFF
       (DEFN "=merely, simply")
       (EXAMPLES "you'll not only find fresh
       donuts made by hand...you'll also find a fresh
       new Dunkin' Donuts shop"))
(SYN
     (neg +))
(SYN-STRUC
     ((root $var1)
      (cat v)
      (adjuncts ((root $var0)))
      (obj ((root $var2)))))

          (SEM
     (LEX-MAP
     (ATTITUDE
(type (value saliency))
(attitude-value (value 0.3))
(scope (value ^$var3))
(attributed-to (value *producer*)))))))
```

The `(neg +)` marker is used to show that *only* in this sense is preceded by *not* and is a part of the correlative *not only ... but also*. The meaning of *only* is represented through a relatively low saliency attitude value. Intuitively this means that the content of the clause introduced by *only* in this sense is considered less salient (or important) by the text producer than the content of the clause introduced by *but also*.

```
(your
    (make-frame
+your-poss1
(CAT (value poss))
(STUFF
      (DEFN "very general sense of association with
      a service-institution")
      (EXAMPLES "drop by your old favorite
      Dunkin Donuts shop" "your local post office
      should be able to help" "your friendly neighborhood
      gas station"))
      ; this is a special sense of your that is closely
      ; tied to a small set of adjectives, such as
      ;"favorite" "local"
(SYN
    (number s2 p2))
(SYN-STRUC
  ((root $var1)
   (cat n)
   (poss ((root $var0)))))
(SEM
    (LEX-MAP
    (^$var1
   (instance-of (sem *service-corporation))
   (has-customer (sem *human)))))
(PRAGM
      (ANALYSIS-TRIGGER
                 (coreferential ^$var1.has-customer *consumer*))))

    (make-frame
+your-poss2
(CAT (value poss))
(STUFF
      (DEFN "owned by/belonging to you")
      (EXAMPLES "can I borrow your book" "if you
      sell your store, you'll have lots of money"))
(SYN
    (number s2 p2))
(SYN-STRUC
  ((root $var1)
   (cat n)
   (poss ((root $var0)))))
(SEM
    (LEX-MAP
          (^$var1
     (instance-of (sem *all))
     (owned-by (sem *human)))))
(PRAGM
        (ANALYSIS-TRIGGER
                (coreferential ^$var1.owned-by *consumer*))))))
```

The salient point of the above definitions is the presence of the `analysis-trigger` slot in the pragmatics zone. The meaning of *your* includes the information that its referent is coreferential with the text consumer. The semantics of the second sense above also includes the indication that the relation between the object modified by *your* and the text consumer is that of ownership. The first sense above is constrained to modifying property meanings (typically realized in natural language through adjectives).

5. Summary

In this paper we illustrated the structure of the lexicon in the Dionysus project. This lexicon is created to be interfaced with an underlying ontological model in terms of which we describe the meanings of open-class lexical items. The treatment of closed-class items in the lexicon is guided by the specification of the text meaning representation language TAMERLAN. The lexicon entry contains a large number of zones with orthographic, morphological, syntactic, semantic and pragmatic information and information about mapping among various levels of representation (primarily, between the syntactic and the semantic dependency structures, that is, f-structures and TAMERLAN texts). The TAMERLAN language is very economical and mostly oriented at representing lexical-semantic meanings. It does not concentrate on the formalisms for deep representation of such phenomena as, for instance, quantification, which have been at the center of interest of many formal and computational semantic theories, often to the exclusion of any useful treatment of lexical meaning. One of the central points of this article has been to demonstrate that for a realistic treamtent of lexical semantics one has to develop all three of the ontology, the lexicon and the text meaning representation language.

Our approach to computational linguistics is based on the concept of microtheories working in an integrated complete testbed system. We envisage enhancements to our representation languages and static knowledge sources through the inclusion of computational interpretations of improved theories of various language phenomena — whenever the latter become available.

Acknowledgements.

Many thanks to the members of the Dionysus project. Boyan Onyshkevych, Ted Gibson and Ralf Brown helped with the construction of parts of TAMERLAN. Ingrid Meier was the Dionysus lexicographer. Lynn Carlson was responsible for acquiring the ontology. Ken Goodman did an excellent job criticizing the theoretical and methodological components of this paper.

Bibliography

Allen, 1983. Maintaining Knowledge about Temporal Intervals. *Communications of ACM*, 26:832-843.

Brachman, R. and R. Levesque (eds.) 1985. *Readings in Knowledge Representation.* San Mateo, CA: Morgan Kaufmann.

Carbonell, 1983. Derivational analogy and its role in problem solving. Proceedings of the 1983 National Conference on Artificial Intelligence, 64-69.

Dordrecht: Kluwer.

Gates, D., D. Haberlach, T. Kaufmann, M. Kee, R. McCardell, T. Mitamura, I. Monarch, S. Morrisson, S. Nirenburg, E. Nyberg, K. Takeda and M. Zabludowski. 1989. Lexicons. *Machine Translation*, 4:67-112.

Goodman, K. (ed.) 1989. Special Issue on Knowledge-Based Machine Translation. *Machine Translation*, 4:1-2.

Hirst, 1987. **Semantic Interpretation and the Resolution of Ambiguity.** Cambridge University Press.

Hirst, 1989. Ontological Assumptions in Knowledge Representation. Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning. Toronto, May.

Kolodner and Riesbeck (eds.) 1986. **Experience, Memory and Reasoning.** Hillsdale, NJ: Erlbaum.

Nirenburg, S. and C. Defrise. 1989. Aspects of Text Meaning. CMU CMT Technical Memo.

Nirenburg et al., in preparation. Nirenburg, S., L. Carlsson, I. Meier, B. Onyshkevych. Ontology and Lexicon in Dionysus.

Nirenburg, S., I. Monarch, T. Kaufmann and I. Nirenburg. Acquisition and Maintenance of Very Large Knowledge Bases. TR-88-108, Center for Machine Translation, Carnegie-Mellon University October 1988.

Nirenburg, S. and J. Pustejovsky. 1988. Processing Aspectual Semantics. Proceedings of the Tenth Annual Meeting of the Cognitive Science Society. Montreal, August

Nyberg, E. 1988. The FRAMEKIT User's Guide. CMU CMT Technical Memo.

Tulving, 1985. How Many Memories Are There? *American Psychologist*, 40:385-398.