

II. Ontological Semantics As Such

In Part II of the book, we discuss the static and dynamic knowledge sources in ontological semantics. We start with an extended example of representing meaning of a natural language text. Next, we describe the static knowledge sources of ontological semantics, after which we present a sketch of ontological semantic processing. Figure 20 illustrates the interactions among the data (marked blue), the processors (red) and the static knowledge sources (green) in ontological semantics.

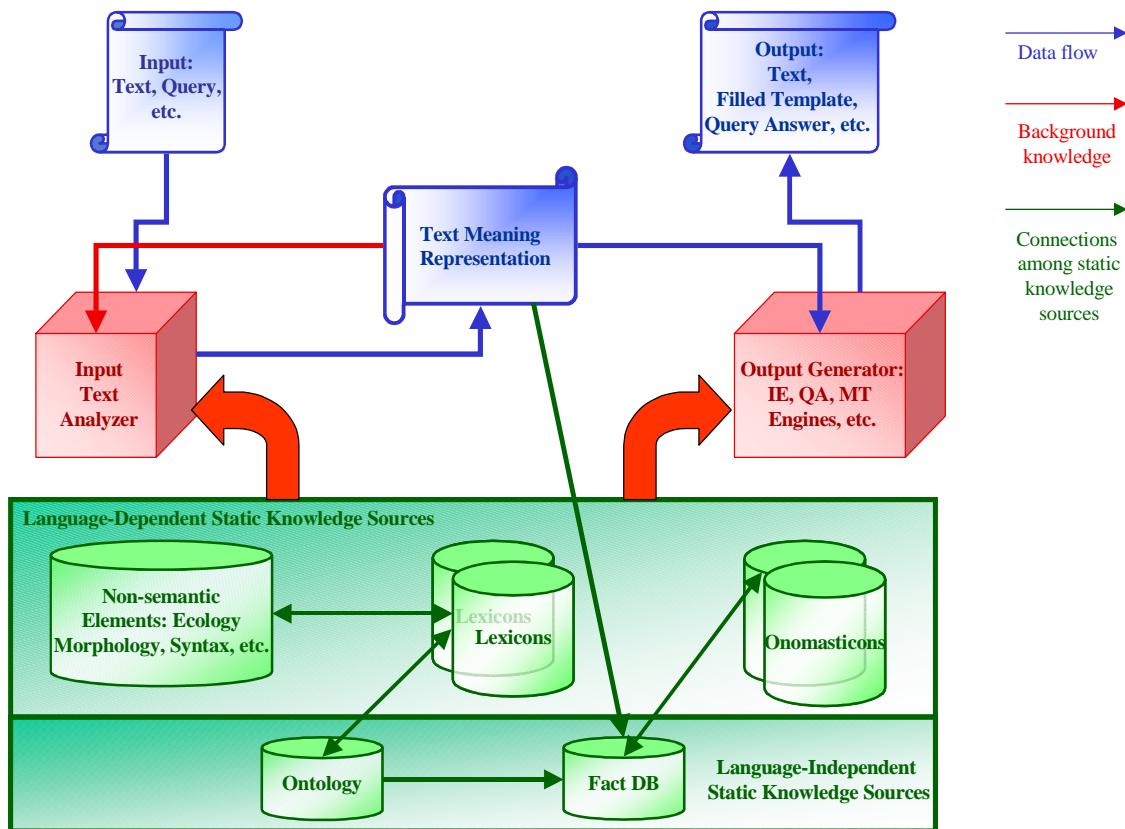


Figure 20. The Data, the Processors and the Static Knowledge Sources in Ontological Semantics.

All ontological semantic applications include analysis of input text as a crucial component. The production of a semantic text meaning representation (TMR) is the result of the analysis process. The analysis modules and output generators use all the available static knowledge sources. TMRs are selectively stored in Fact DB for future reference, support of various applications and treatment of reference.

Ontological semantic applications include machine translation, information extraction (IE), question answering (QA), general human-computer dialog systems, text summarization and specialized applications combining some or all of the above with additional functionality (e.g., advice giving systems). Of course, such applications are attempted without ontological semantics, or, for

that matter, without any treatment of meaning at all. If, however, these applications are based on ontological semantics, then any kind of input to the system (an input text for MT, a query for a question answering system, a text stream for information extraction, etc.) first undergoes several stages of analysis (tokenization, morphological, syntactic, semantic, etc.—see Chapter 8 below for details) that, in the case of success, in the end generate the meaning of a text, “text meaning representation” or TMR. The TMR serves as input to specialized processing relevant for a particular application. For example, in MT, the TMR needs to be translated into a natural language different from the one in which the input was supplied. The program that carries this task out is usually called text generator. In IE, TMRs are used by the special rules as sources of fillers of IE template slots. In question answering, the TMR presents the proximate meaning of the user’s query. The QA processor must first understand exactly what the user wants the system to do, then find the necessary information either in the background world knowledge sources (most often, Fact DB, but sometimes the ontology or the lexicons) and then generate a well-formed answer.

The static knowledge sources include the language-dependent ones—the rules for text tokenization, detecting proper names and acronyms and other preprocessing tasks (we call these tasks ecological), for morphological, syntactic and ontological semantic analysis. The information for the latter three types of analysis resides largely in the lexicons of the system, though special rules (e.g., syntactic grammars) are separate from lexicons. In the current state of ontological semantics, onomasticons, repositories of proper names, are separated from regular lexicons. The language independent static knowledge sources are the ontology and the fact database (Fact DB). The ontology contains information about how things can be in the world while the Fact DB contains actual facts, that is, events that took place or objects that existed, exist or have been reported to exist. In other words, the ontology contains concept types, whereas the Fact DB contains remembered concept instances. Onomasticons contain information about words and phrases in natural language that name remembered concept instances. These concept instance names are also recorded as property fillers in Fact DB frames. Note that the Fact DB also contains other, unnamed, concept instances. More detailed descriptions of all the static knowledge sources are given in Chapter 7.

In most applications of ontological semantics, a side effect of the system’s operation is selective augmentation of the Fact DB with the elements of TMRs produced during input analysis stage. This way, this information remains available for future use. It is in this sense that we can say that ontological semantic applications involve learning: the more they operate, the more world knowledge they record, the better quality results they may expect.

6. Meaning Representation in Ontological Semantics

6.1 Meaning Proper and the Rest

Consider the following text as input to an ontological-semantic processor.

- (1) Dresser Industries said it expects that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan.

In “Computerese,” that is, in the form that we expect that a semantic analyzer would be able to process and represent the above text, the latter will be glossed, for example, as follows:

- (2) A spokesperson for the company called Dresser Industries made this statement: Dresser Industries expects that imports into the US from Japan will decrease through large capital investment for the purpose of expanding the manufacturing potential in the US; the expenditure precedes expansion, which precedes reduction, and all of them take place after the statement.

In a somewhat more formal fashion, the meaning of (1) glossed in (2) can be seen to include the following meaning components:

- (3)
- (i) that *Dresser Industries* is a phrase, moreover, a set phrase, a proper name;
 - (ii) that it is the name of a company;
 - (iii) that this name is used in the original text metonymically—the company name, in fact, stands for its unnamed spokesperson(s);
 - (iv) that the spokesperson made a statement (that is, not a question or a command);
 - (v) that the company (once again, metonymically) has a certain belief, namely, an expectation;
 - (vi) that the scope of the expectation is the reduction of imports into US from Japan;
 - (vii) that the reduction of imports is expected to take place through capital investment;
 - (viii) that the purpose of the investment is to increase the capacity for manufacturing in the United States;
 - (ix) that United States refers to a nation, the United States of America and Japan refers to another nation, Japan;
 - (x) that the object of manufacturing, that is left unnamed in the original text is most likely to refer to goods;
 - (xi) that the decrease occurs in the amount of goods that the United States imports from Japan;
 - (xii) that the time at which reduction of imports occurs follows the time of investment which, in turn, preceded the expansion of manufacturing capacity;
 - (xiii) that the time at which the statement was made precedes the time of investment;
 - (xiv) that what is expanded is not necessarily the actual manufacturing output but the potential for it.

The set of expressions in (3) can be viewed as the meaning of (1). In fact, this is the level at which text meaning is defined in the Mikrokosmos implementation of ontological semantics. However, it is important to understand that there may be alternative formulations of what constitutes the meaning of (1) or, for that matter, of any text. So, it seems appropriate at this point to discuss the general issue of how exactly to define text meaning. It might come as a surprise that this is not such an easy question! One attempt at making the idea of meaning better defined is the introduction of the notion of literal meaning (cf., e.g., Hausser 1999:20). Thus, we could have declared that what we represent in our approach is the literal meaning of texts. However, this decision

meets with difficulties because the notion of literal meaning may not be defined sufficiently precisely. For instance, (3) can be construed as the literal meaning of (1). However, under a different interpretation, deciding to resolve the organization-for-employee metonymy in (3.iii) and (3.v) may be construed as going beyond literal meaning. (3) can be seen as the literal meaning of (1) if one agrees that *Dresser Industries*, being a company, cannot actually be the agent of saying. If this constraint is lifted, by allowing organizations to be agents of speech acts, then the literal meaning will not require the resolution of metonymy. In other words, this kind of literal meaning will be represented by eliminating (3.iii) and (3.v) from (3). In fact, if this approach is adopted throughout, the concept of metonymy will be summarily dispensed with (Mahesh *et al.* 1996; Section 8.4.2). As the concept of literal meaning can be understood in a variety of ways, we found it unhelpful for defining which kinds of information belong in text meaning and which remain outside it, while still possibly playing a role (of background knowledge used for inference making in reasoning applications) in text processing in a variety of applications.

We have just considered a possibility of representing the meaning of (1) using less information than shown in (3). It is equally possible to view an expanded version of (3) as the meaning of (1). One example of such expansion would add statements in (4) to the list (3):

- (4) (i) that the company Dresser Industries exists;
- (ii) that Dresser Industries has an opinion on the subject of reducing imports from Japan;
- (iii) that the most probable source of investment that would lead to the expansion of the US manufacturing capacity is either Dresser Industries itself or a joint venture of which it is a part;
- (iv) that the goal of reducing imports is a desirable one.

(4.i) is known as a(n existential) presupposition for (1). (4.ii) is an entailment of (1). Should they be considered integral parts of the meaning of (1)? Information in (4.iii) and (4.iv) is inferred from (1) on the basis of general knowledge about the world. For example, (4.iii) relies on the belief that if it is not stated otherwise, it is strongly probable that Dresser Industries also plans to participate in the expansion of the US manufacturing capacity. It is noteworthy that, unlike for (4.i) and (4.ii), (4.iii) and (4.iv) are not expected to be always true.

Let us explore a little further what this actually means. One way of approaching the task of determining the exact meaning of a text is by using the negation test, a typical linguistic tool for justifying an element of description by showing that its exclusion leads to some sort of deviance, for instance, a contradiction (see, e.g., Raskin 1985). Indeed, the negation of any element of (3) contradicts some component of the meaning of (1). We may take this as an indication that each element of (3) is a necessary part of the meaning of (1). But is it correct to say that *any* statement whose negation contradicts (1) is a necessary part of the meaning of (1)? Let us consider a few more cases.

It is easy to see why (5.1) and (5.2) are contradictory. Each of them consists of (1) and the negation of one of the component clauses of (1). Obviously, the contradiction results from the fact

that the negated component is an integral part of the meaning of (1).

- (5) (i) Dresser Industries **said** it expects that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan, **and** Dresser Industries **did not say** that it expects that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan;
- (ii) Dresser Industries said it **expects** that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan, **and** Dresser Industries said it **does not expect** that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan.

Similarly, contradictory statements will result from adding the negations of (4.i) and (4.ii) to (1), to yield (6.i) and (6.ii):

- (6) (i) Dresser Industries said it expects that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan, and Dresser Industries does not exist;
- (ii) Dresser Industries said it expects that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan, and Dresser Industries has no opinion on the subject of reducing imports from Japan.

The source of contradictions in (6) is different, however, from the source of contradictions in (5). The statements added in (6) do not negate anything directly stated in (1). They negate a presupposition and an entailment of (1), respectively: if it is not presupposed that Dresser Industries exists, (1) makes no sense; if it does not follow from (1) that Dresser Industries has an opinion on the subject of imports from Japan, (1) does not make sense, either. As we can see, the negation tool fails to distinguish between the actual elements of the meaning of (1), on the one hand and the presuppositions and entailments of (1), on the other. This outcome gives us two alternatives—either to include presuppositions and entailments in the meaning of (1) (or, by extension, of any statement) or to ignore the results of the negation test in this case.

This distinction turns out to be problematic for people as well. Thus, delayed recall experiments (Chafe 1977) show something that trial lawyers have always known about witness testimony, namely, that people never recall exactly what was said—only the gist of it—and that they routinely confuse the presuppositions and entailments of a statement with what the statements actually assert. The distinction may, however, be quite important in those NLP applications where it is important to distinguish between what is conveyed by the text directly and what is present only by implication. For example, at the text generation step of machine translation what must be translated is the actually made statements and not what they presuppose or entail, the reason being the assumption that the readers will be able to recreate all the implications that were present but not overtly stated in the original text.

The negation tool does, however, work well for (4.iii) and (4.iv). Adding their negations to (1) yields (7.i) and (7.ii) that are somewhat odd but not contradictory:

- (7) (i) Dresser Industries said it expects that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan, and it is not the case that Dresser Industries or a joint venture of which it is a part are the most probable source of investment in the US manufacturing capacity;
- (ii) Dresser Industries said it expects that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan, and the goal of reducing imports is not a desirable one.

We conclude that the reason for the absence of contradictions in (7) is that (4.iii) and (4.iv) do not negate any elements of the meaning of (1). In general, we assume that if adding the negation of a statement to another statement is not contradictory, then the former statement does not constitute a part of the meaning of the latter statement. One can also say then that there are no contradictions in (7) because (4.iii) and (4.iv) are possible but not necessary entailments from (1).

Many more such possible statements can be inferred from (1) based on the general knowledge about companies and how publicity works, for instance:

- (8) (i) that Dresser Industries has a headquarters;
- (ii) that it has employees;
- (iii) that it manufactures particular products and/or offers particular services;
- (iv) that the addressee of the statement by the spokesperson of Dresser Industries was the general public;
- (v) that the statement has been, most probably, made through the mass media, etc.

Even more inferences can be made from (1) based on the general understanding of goals that organizations and people typically pursue as well as plans that they use to attain those goals:

- (9) (i) that there is a benefit for Dresser Industries in expanding the US manufacturing capacity;
- (ii) that capital investment is a plan toward attaining the goal of expanding manufacturing capacity;
- (iii) that this goal can play the role of a step in a plan of attaining the goal of reducing imports; or
- (iv) that Dresser Industries knows about using mass media as a plan for attaining a variety of goals.

All the inferences in (7 - 9) are not “legal” (cf. Charniak and McDermott 1985:21) deductions but rather abductive, defeasible, negatable inferences. It is for this reason that none of them are included in the specification of the meaning of (1). The distinction between meaning proper, on the one hand, and presuppositions, entailments and inferences, on the other, may not be as important for NLP applications whose results are not intended for direct human consumption, e.g., for text data mining aiming at automatic population of databases. People, however, are capable of generating presuppositions, entailments and inferences on the fly from a brief message. Indeed, brevity is at a premium in human professional and business communication. Text meaning or even condensed text meaning are, thus, the central objects of manipulation in such common applications as machine translation and text summarization, respectively.

For computers, brevity of the kind to which we are referring has little real physical sense in these days of inexpensive storage devices and fast indexing and search algorithms. What is difficult for computer systems is precisely making reliable and relevant inferences. Therefore, spelling out as many inferences as possible from a text and recording them explicitly in a well-indexed manner for future retrieval is essential for supporting a variety of computational applications.

It is important for a computational semantic theory to provide the means of supporting both these precepts—of brevity and of explicitness. A representation of text meaning should be as brief as possible, if it is to be the source for generating a text for human consumption. The knowledge about both the building blocks of the meaning representation and the types of inferences that are possible from a particular text meaning should be stored in an accessible fashion. These kinds of knowledge are interchangeable with the change of inputs—what was a part of text meaning for one source text may end up being a source of inference for another. Any computational semantic application must support this capability of dynamically assigning some of the resident knowledge to direct meaning representations and reserving the rest for possible inferences. In ontological semantics, these goals are achieved through interrelationship among text meaning representations (TMRs), the lexicons and the ontology.

6.2 TMR in Ontological Semantics

Meaning of natural language texts is represented in ontological semantics as a result of a compositional process that relies on the meanings of words, of bound morphemes, of syntactic structures and of word, phrase and clause order in the input text. The meanings of words reside in the lexicon and the onomasticon (the lexicon of names). The bound morphemes (e.g., markers of Plural for nouns) are processed during morphological analysis and get their meanings recorded in special rules, possibly, added to classes of lexical entries. Information about dependency among lexical elements and phrases, derived in syntax, helps to establish relationships of semantic dependency. Word and phrase order in some languages play a similar role.

It is clear then that the knowledge necessary for ontological semantic analysis of text should include not only the lexical material for the language of the text but also the results of the morphological and syntactic analysis of the input text. Let us follow the process of creating an ontological-semantic TMR using the example in (1), repeated here as (10).

- (10) Dresser Industries said it expects that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan.

English is a morphologically impoverished language, but morphological analysis of (10) will still yield some non-trivial results:⁶⁸

Root	Part of Speech	Features
Dresser Industries	Phrase Proper	Number: Singular
say	Verb	Tense: Past
it	Pronoun	Number: Singular; Person: Third
expect	Verb	Tense: Present; Number: Singular; Person: Third
that	Binder	
major	Adjective	
capital	Noun	Number: Singular
expenditure	Noun	Number: Singular
for	Preposition	
expansion	Noun	Number: Singular
of	Preposition	
U.S.	Acronym	Number: Singular
manufacturing	Verb	Form: Gerund
capacity	Noun	Number: Singular
reduce	Verb	Tense: Future (<i>will</i> marks this in the text)
import	Noun	Number: Plural
from	Preposition	
Japan	Noun Proper	

Results of syntactic analysis of (10) can be represented in the following structure (which is modeled on the f-structure of LFG (e.g., L. Levin 1991):

68. The nature and format of morphological and syntactic analyses presented here are outside the purview of ontological semantics and of our narrative. We are fully aware that many other formulations and presentations of these analysis steps are possible. Ontological semantics is neutral to any such formulation and can be adapted to work with any good quality morphological and syntactic analyzer.

(11)

```
root    say
cat     verb
tense   past
subject
      root dresser industries
      cat  phrase-proper
comp
      root expect
      cat  verb
      tense present
      subject
          root dresser industries
          cat  phrase-proper
      object
          root reduce
          cat  verb
          tense future
          subject
              root expenditure
              cat  noun
              modifier capital
              cat  noun
              modifier major
              cat  adjective
          oblique root for
                  cat  preposition
          object  root expansion
                  cat  noun
                  oblique root of
                          cat  preposition
          object root capacity
                  cat  noun
                  modifier root manufacturing
                          cat  verb
                          modifier root u.s.
                                  cat phrase-proper
      object root imports
              cat  noun
              oblique root from
                      cat  preposition
          object  root japan
                  cat  noun-proper
```

We will now use the results of the morphological and syntactic analysis presented above in building a TMR for (10). TMRs are written in a formal language with its own syntax specified in Section 6.4. For pedagogical reasons, at many points in our presentation here, we will use a somewhat simplified version of that language and will build the TMR for (10) step by step, not necessarily in the order that any actual analyzer will follow.

The first step in ontological semantic analysis is finding meanings for heads of clauses in the syntactic representation of input. In our example, these are *say*, *expect* and *reduce*. As we will see,

they all will be treated differently in TMR construction. In addition, the TMR will end up containing more event instances (“proposition heads”—see Section 8.2.1 below) than there are verbs in the original text. This is because ontological semantics is “transcategorical” in that meanings are not conditioned by part of speech tags. Specifically, in (1) the nouns *expenditure* and *expansion* occupying the syntactic positions corresponding typically to heads of noun phrases, are mapped into instances of event-type concepts in the TMR.

In (12), we present the syntactic-structure (SYN-STRUC) and semantic-structure (SEM-STRUC) components of the entry for *say* in the ontological semantic lexicon of English. The meaning of *say* instantiates the ontological concept INFORM. The representation of this concept, shown in (13), contains a number of properties (“slots”), with a specification of what type of object can be a legal value (“filler”) for each property.

(12)

```
say-v1
  syn-struct
    1   root  say                ; as in Spencer said a word
        cat  v
        subj root  $var1
           cat  n
        obj  root  $var2
           cat  n
    2   root  say                ; as in Spencer said that it rained
        cat  v
        subj root  $var1
           cat  n
        comp root  $var2
  sem-struct
    1 2   inform                ; both syntactic structures have the same semantic structure,
        agent value ^$var1     ; ‘^’ is read as ‘the meaning of,’ and
        theme value ^$var2     ; the variables provide mappings between
                                ; syntactic and semantic structures
```

(13)

```
inform
  definition      “the event of asserting something to provide information
                  to another person or set of persons”
  is-a           assertive-act
  agent          human
  theme          event
  instrument     communication-device
  beneficiary    human
```

So far, then, the nascent TMR for (1) has the form:

(14)

```
inform-1
  agent value _____
  theme value _____
```

The arbitrary but unique numbers appended to the names of concepts during ontological semantic

processing identify instances of concepts. The numbers themselves are also used for establishing co-reference relations among the same instances. At the next step of semantic analysis, the process seeks to establish whether fillers are available in the input for these properties. If the fillers are not available directly, there are special procedures to try to establish them. If these recovery procedures fail to identify the filler but it is known that some filler must exist in principle, the special filler UNKNOWN is used.

The AGENT slot in (14) cannot be filled directly from the text. The reason for that is as follows. The procedure for determining the filler attempts to use the syntax-to-semantics mapping in the lexicon entry for *say*, to establish the filler for the particular slots. The lexicon entry for *say* essentially states that the meaning, $\wedge \$var1$, of the syntactic subject of *say*, $\$var1$, should be the filler of the AGENT slot of INFORM. Before inserting a filler, the system checks whether it matches the ontological constraint for AGENT of INFORM and discovers that the match occurs on the RELAXABLE-TO facet of the AGENT slot, because *Dresser Industries* is an organization. Note that the ontological status of DRESSER INDUSTRIES is that of a (named) instance of the concept CORPORATION—see Section 4.2.1 for a discussion of instances and remembered instances.

The TMR at this point looks as illustrated in (15).

(15)

```
inform-1
  agent  value  Dresser Industries
  theme  value  _____
```

The theme slot in (14) requires a more complex treatment.⁶⁹ The complement of *say* in the syntactic representation (11) is a statement of expectation. According to a general rule, the direct object of the syntactic clause should be considered as the prime candidate for producing the filler for THEME. Expectation, however, is considered in ontological semantics to be a modality and is, therefore, represented in TMR as a property of the proposition that represents the meaning of the clause that modifies it syntactically. Before assigning properties, such as this modality, we will first finish representing the basic meanings that these properties characterize. Therefore, a different candidate for filling the theme property must be found. The next candidate is the clause headed by *reduce*. Consulting the lexicon and the ontology and using the standard rules of matching selectional restrictions yields (16):

(16)

```
inform-1
  agent  value  Dresser Industries
  theme  value  decrease-1
decrease-1
  agent  value  unknown
```

Continuing along this path, we fill the case roles THEME and INSTRUMENT in (16), as well as their own properties and the properties of their properties, all the way down, as shown in (17):

69. We would like to apologize for using a complex, real-life text as our detailed example. Simple examples, often used to illustrate the properties of a representation language, fail to demonstrate in sufficient detail the features of the language or, more importantly, its ability to handle realistic inputs.

(17)

inform-1			
	agent	value	Dresser Industries
	theme	value	decrease-1
decrease-1			
	agent	value	unknown
	theme	value	import-1
	instrument	value	expend-1
import-1			
	agent	value	unknown
	theme	value	unknown
	source	value	Japan
	destination	value	USA
expend-1			
	agent	value	unknown
	theme	value	money-1
		amount	value > 0.7
	purpose	value	increase-1
increase-1			
	agent	value	unknown
	theme	value	manufacture-1.theme
manufacture-1			
	agent	value	unknown
	theme	value	unknown
	location	value	USA

Some elements of (17) are not self-evident and require an explanation. First, the value of the property AMOUNT of the concept MONEY (which is the meaning of *capital* in the input) is rendered as a region on an abstract scale between 0 and 1, with the value corresponding to the meaning of the word *major*. The same value would be assigned to other words denoting a large quantity, such as *large*, *great*, *much*, *many*, etc. The meanings of words like *enormous*, *huge* or *gigantic* would be assigned a higher value, say, > 0.9. THEME of INCREASE is constrained to SCALAR-OBJECT-ATTRIBUTE and its ontological descendants, of which AMOUNT is one. The filler of the THEME of increase-1 turns out to be the property AMOUNT itself (not a value of this property!) referenced as the THEME of manufacture-1, rendered in the familiar dot notation.

Now that we have finished building the main “who did what to whom” semantic dependency structure, let us add those features that are in ontological semantics factored out into specific parameterized properties, such as speech act, modality, time or co-reference. The top proposition in (18) reflects the speech act information that in the text (1) is not expressed explicitly, namely, the speech act of publishing (1) in whatever medium. The speech act introduces an instance of the ontological concept AUTHOR-EVENT (see also Section 6.5 below).

(18)

author-event-1

agent value unknown
theme value inform-1
time
time-begin > inform-1.time-end
time-end unknown

inform-1

agent value Dresser Industries
theme value decrease-1
time
time-begin unknown
time-end (< decrease-1.time-begin) (< import-1.time-begin) (< reduce-1.time-begin)
(< expend-1.time-begin) (< increase-1.time-begin)

decrease-1

agent value unknown
theme value import-1
instrument value expend-1
time
time-begin (> inform-1.inform-1.time-end) (> expend-1.time-begin) (> import-1.time-begin)
time-end < import-1.begin-time

import-1

agent value unknown
theme value unknown
source value Japan
destination value USA
time
time-begin (> inform.time-end) (< expend-1.begin-time)
time-end unknown

expend-1

agent value unknown
theme value money-1
amount value > 0.7
purpose value increase-1
time
time-begin > inform.time-end
time-end < increase-1.begin-time

increase-1

agent value unknown
theme value manufacture-1.theme
time
time-begin (> inform.time-end) (< manufacture-1.begin-time)
time-end unknown

```

manufacture-1
  agent      value  unknown
  theme      value  unknown
  location   value  USA
  time
    time-begin > inform.time-end
    time-end   unknown

modality-1
  type  potential ;this is the meaning of expects in (1)
  value 1 ;this is the maximum value of potential
  scope decrease-1

modality-2
  type  potential ;this is the meaning of capacity in (1)
  value 1
  scope manufacture-1

co-reference-1
  increase-1.agent manufacture-1.agent

co-reference-2
  import-1.theme manufacture-1.theme

```

The time property values in each proposition, all relative since there is no absolute reference to time in the input sentence, establish a partial temporal order of the various events in (1): for example, that the time of the statement by Dresser Industries precedes the time of reporting. The expected events may only take place after the statement is made. It is not clear, however, how the time of reporting relates to the times of the expected events because some of them may have already taken place between the time of the statement and the time of reporting.

Inserting the value UNKNOWN into appropriate slots in the TMR actually undersells the system's capabilities. In reality, while the exact filler might not be indeed known, the system knows many constraints on this filler. These constraints come from the ontological specification of the concept in which the property that gets the UNKNOWN filler is defined and, if included in the TMR, turn it into what we define as extended TMR (see Section 6.7 below). Thus, the AGENT of import-1 is constrained to U.S. import companies. The AGENT of expend-1 is constrained to people and organizations that are investors. The AGENT of increase-1 and manufacture-1 is constrained to manufacturing corporations. The THEME of import-1 and manufacture-1 is constrained to GOODS (the idea being that if you manufacture some goods then you do not have to import them). The facts that *Dresser Industries* is a company while *Japan* and *USA* are countries are stored in the onomasticon.

6.3 Ontological Concepts and Non-Ontological Parameters in TMR

The above example was presented to introduce the main elements of a TMR in ontological semantics. A careful reader will have established by now that our approach to representing text meaning uses two basic means—instantiation of ontological concepts and instantiation of semantic parameters unconnected to the ontology. The former (see (17) above) creates abstract, undindexed⁷⁰ propositions that correspond to any of a number of possible TMR instantiations. These instantiations (see the material in (18) which is not present in (17)) are obtained by supplementing

the basic ontological statement with concrete indexical values of parameters such as aspect, style, co-reference and others.

One strong motivation for this division is size economy in the ontology. Indeed, one could avoid introducing the parameter of, say, aspect opting instead for introducing the ontological attribute ASPECT whose DOMAIN is EVENT and whose RANGE is the literal set of aspectual values. The result would be either different concepts for different aspectual senses of each verb, e.g., the concepts READ, HAVE-READ, BE-READING, and HAVE-BEEN-READING instead of a single concept READ or the introduction of the ontological property ASPECT for each EVENT concept. The former decision would mean at least quadrupling the number of EVENT type concepts just in order to avoid introducing this one parameter. An objection to the latter decision is that aspect—as well as modality, time and other proposition-level parameters—is defined for concept instances, not ontological concepts themselves.

The boundary between ontological and parametric specification of meaning is not fixed in ontological semantics. Different specific implementations are possible. In the Mikrokosmos implementation of ontological semantics, the boundary between the parametric and ontological components of text meaning is realized as formulated in the BNF specification in the next section.

6.4 The Nature and Format of TMR

In this section, we introduce the format of the TMR. As it is presented, this format does not exactly correspond to those in any of the implementations of ontological semantics. We present a composite version that we believe to be easiest to describe. The TMR format in actual implementations can and will be somewhat different in details, for instance, simplifying or even omitting elements that are tangential to a particular application. The BNF below specifies the syntax of the TMR. The semantics of this formalism is determined by the purpose for which the BNF constructs are introduced. Therefore, the convenient place for describing the semantics of the TMR is in the sections devoted to the process of deriving TMRs from texts (see Chapter 8 below).

In the BNF, “{ }” are used for grouping; “[]” means optional (i.e., 0 or 1); “+” means 1 or more; and “*” means 0 or more.

Informally, the TMR consists of a set of propositions connected through text-level discourse relations. Parameters at this top level of TMR specification include style, co-reference and TMR time (see Section 8.6 below).

```
TMR ::=
    PROPOSITION+
    DISCOURSE-RELATION*
    STYLE
    REFERENCE*
    TMR-TIME
```

A proposition is a unit of semantic representation corresponding to a single predication in text (in

70. We use the term index in the sense of Bar Hillel (1954) and Lewis (1972) to refer to time, place, possible world, speaker, hearer and other coordinates that turn an abstract proposition into a real utterance.

Mikrokosmos, all TMRs have been produced as a result of analysis of a natural language text). Syntactically, single predications are typically realized as clauses. At the level of proposition, aspect, modality, time of proposition, the overall TMR time and style are parametrized.

PROPOSITION ::=

proposition

head: concept-instance
 ASPECT
 MODALITY*
 PROPOSITION-TIME
 STYLE

The terms in bold face are terminal symbols in the TMR. The main carrier of semantic information is the head of a proposition. Finding the head and filling its properties with appropriate material in the input constitutes the two main processes in ontological semantic analysis—instantiation and matching of selectional restrictions (see Section 8.2.2).

ASPECT ::=

aspect

aspect-scope: concept-instance
phase: **begin** | **continue** | **end** | **begin-continue-end**
iteration: integer | **multiple**

The symbols ‘concept-instance,’ ‘integer,’ ‘boolean’ and ‘real-number’ (see below) are interpreted in a standard fashion (see Section 7.1 for an explanation of the notion of instantiation) and not formally described in this BNF (see Section 8.5.1 for an explanation of the interpretation of aspect in ontological semantics).

TMR-TIME ::= set

element-type	proposition-time
cardinality	>= 1

TMR-time is defined as a set of all the values of times of propositions in the TMR. This effectively imposes a partial ordering on the propositions. Can be derived automatically from the values of proposition-time.

PROPOSITION-TIME ::=

time

time-begin: TIME-EXPR*
time-end: TIME-EXPR*

Time expressions refer to point times; durations are calculated from the beginnings and ends of time periods.

TIME-EXPR ::= << | < | > | >> | >= | <= | = | !=
 { ABSOLUTE-TIME | RELATIVE-TIME }

ABSOLUTE-TIME ::= { +/- } YYY YMMDDHHMMSSFFFF [[+/-] real-number temporal-unit]

The above says that times of propositions are given in terms of the times of their beginnings and

ends and can be expressed through a reference to an absolute time, represented as year-month-day-hour-minute-second-fraction-of-second (negative values refer to times before common era) or to a time point that is a certain time period before or after the above reference point.

RELATIVE-TIME ::= CONCEPT-INSTANCE.TIME [[+/-] real-number temporal-unit]

Alternatively, time-begin and time-end can be filled with relative times, that is, a reference to the time of another concept instance, e.g., an event, again possibly modified by the addition (*a week after graduation*) or subtraction (*six years before he died*) of a time period—see Section 8.5.2 for a detailed discussion of proposition time.

MODALITY ::=

modality

modality-type:	MODALITY-TYPE
modality-value:	(0,1)
modality-scope:	concept-instance*
modality-attributed-to:	concept-instance*

The value (0,1) refers to the abstract scale of values or intervals running between zero and unity. This and other types of property fillers (for example, the literal values of the `modality-type` property—see below) are discussed in greater detail in Section 7.1.

MODALITY-TYPE ::= **epistemic | deontic | volitive | potential | epiteuctic | evaluative | saliency**

The semantics of the above labels is described in Section 8.5.3.

STYLE ::=

style

formality:	(0,1)
politeness:	(0,1)
respect:	(0,1)
force:	(0,1)
simplicity:	(0,1)
color:	(0,1)
directness:	(0,1)

Definitions of the above properties are given in Section 8.6.4.

DISCOURSE-RELATION ::=

relation-type:	ontosubtree(discourse-relation)
domain:	proposition+
range:	proposition+

‘ontosubtree’ is a function that returns all the descendants of the ontological concept that is its argument, including the argument itself. In the above specification, the function returns all the discourse relations defined in the ontology.

```

REFERENCE      ::= SET
                element-type  SET
                                element-type concept-instance
                                cardinality  >=1
                cardinality    >= 1

```

The above is the way of recording co-reference information (see Section 8.6.1 for a discussion).

```

SET      ::=
set
  element-type:  concept | concept-instance
  cardinality:   [ < | > | >= | <= | <> ] integer
  complete:      boolean
  excluding:     [ concept | concept-instance]*
  elements:      concept-instance*
  subset-of:     SET

```

The set construct, as used in ontological semantics, is rather complex. The motivation for including all the above properties is the ease of formalizing a variety of kinds of references in natural language texts to groups of objects, events or properties. ‘Element-type,’ ‘cardinality’ and ‘subset-of’ are self-explanatory. The property ‘complete’ records whether the set lists all the elements that it can in principle have. In other words, the set of all college students will have the Boolean value ‘true’ in its ‘complete’ slot. This mechanism is the way of representing universal quantification. The value of the English word *some* (which can be understood as an existential quantifier) is represented by the Boolean value ‘false’ in the ‘complete’ slot. The ‘excluding’ property allows one to define a set using set difference, for instance, to represent the meaning of such texts as *Everybody but Bill and Peter agreed*. The ‘elements’ property is used for listing the elements of the set directly.

6.5 Further Examples of TMR Specification

In this section, we will discuss some of the standard ways of representing a few less obvious cases of meaning representation using the TMR format.

If an input text contains a modifier-modified pair, then the meaning of the modifier is expected to be expressed as the value of a property of the modified (see Raskin and Nirenburg 1996 on the microtheory of adjectival meaning as a proposal for modification treatment in ontological semantics). This property is, in fact, part of the meaning of the modifier whose other component is the appropriate value on this 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

```

car-5
  instance-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 may be represented in one of a number of available ways, including the following:

- as a modality value: for instance, the meaning of *favorite* in *your favorite Dunkin' Donuts shop* is expressed through an evaluative modality scoping over the head of the phrase;
- as a separate clause, semantically connected to the meaning of the governing clause through co-reference of property fillers;
- as a relation among other TMR elements.

On a more general note with respect to reference, consider the sentence *The Iliad was written not by Homer but by another man with the same name.*⁷¹ We will not discuss the processing of the ellipsis in this sentence (see Section 8.4.4 for a discussion of treating this phenomenon). After the ellipsis is processed, the sentence will look as follows: *Iliad was not written by Homer; Iliad was written by a different man whose name was Homer.* The meanings of the first mention of *Homer* and *Iliad* are instantiated from the concepts HUMAN and BOOK, respectively. Just like JAPAN and USA in (18), they will be referred to by name in the TMR. The second mention of *Homer* will be represented “on general principles,” that is, using a numbered instance of HUMAN, with all the properties attested in the text overtly listed. There are two event instances referred to in the sentence, both of them instances of AUTHOR-EVENT.

author-event-1

agent	value	Homer
theme	value	Iliad

modality-1

scope	author-event-1	;a method of representing negation
modality-type	epistemic	
modality-value	0	

author-event-2

agent	value	human-2	
		name	value Homer
theme	value	Iliad	

co-reference-3

Homer	human-2
-------	---------

modality-2

scope	co-reference-3
modality-type	epistemic
modality-value	0

Another example of special representation strategy is questions and commands. In fact, to deal with this issue, we must first better understand how we are treating assertions. All our examples so far have been assertions, though we have not characterized them as such, as there was nothing with which to compare them. In linguistic and philosophical theory, assertions, questions and commands are types of illocutionary acts, or less formally, speech acts (Austin 1962, Searle 1969, 1975). This brings about the general issue of how to treat speech acts in TMRs.

Our solution is to present every proposition as the theme of a communication event whose agent

71. We will not focus here on what makes this sentence funny—see Raskin 1985 and Attardo 1994 for a discussion of semantic analysis of humor.

is the author (speaker) of the text that we are analyzing. Sometimes, such a communication event is overtly stated in the text, e.g., *I promise to perform well on the exams*. Most of the time, however, such an event is implied, e.g., *I will perform well on the exams*, which can be uttered in exactly the same circumstances as the former example and have the same meaning.⁷² Note that we included the implicit communication event with the reporter as the author in the detailed example (18).

For questions and commands, similarly, also the implicit communication event must be represented in order to characterize the speech act correctly. We represent questions using values of the ontological concept REQUEST-INFORMATION with its theme filled by the element about which the question is asked. If the latter is the value of a property of a concept instance, then this is a special question about the filler of this property. For example, the question *Who won the match?* is represented as:

```
win-32
  theme  value  sports-match-2
request-information-13
  theme  value  win-32.agent
```

If an entire proposition fills the THEME property of REQUEST-INFORMATION, then this is a general yes/no question, e.g., *Did Arsenal win the match?* will be represented as

```
win-33
  agent  value  Arsenal
  theme  value  sports-match-3
request-information-13
  theme  value  win-33
```

The meaning of the sentence *Was it Arsenal who won the match?* will be represented as

```
win-33
  agent  value  Arsenal
  theme  value  sports-match-3
request-information-13
  theme  value  win-33
modality-11
  type   salience
  scope  win-32.theme
  value  1
```

Commands are treated in a similar fashion, except that the ontological concept used for the—often implicit—communication event is REQUEST-ACTION whose theme is always an EVENT.

Speech act theory deals with an unspecified large number of illocutionary acts, such as promises, threats, apologies, greetings, etc. Some such acts are explicit, that is, the text contains a specific

72. Incidentally, this treatment agrees with the theory of latent performatives (Austin 1958, Searle 1989, Bach and Harnish 1992).

reference to the appropriate communication event but most are not. To complicate matters further, one type of speech act—whether explicit or implicit—may stand for another type: thus, in *Can you pass the salt?* what on the surface seems to be a direct speech act, a question, is, in fact, an indirect speech act of request.

As speech act theory has never been intended or used for text processing, neither Austin nor Searle were interested in the boundaries of meaning specification and the differences between meaning proper and inferences. Thus, a significant distinction was ignored. This state of affairs has practical consequences, too. As we have discussed in Section 6.1 above, in NLP it is important to know when to stop meaning analysis.

Therefore, it is important to understand that such speech acts as assertions, questions and commands (and very likely nothing else) are part of the meaning of a text, while others are typically inferred, unless they are overtly stated in the text (e.g., *I regret to inform you that the hotel is fully booked*). As with all inferences, there is no guarantee that the system will have all the knowledge necessary for computing such inferences. As a result, the analysis may have to halt before all possible inferences have been made. As was mentioned in Section 6.1 above (see also Section 6.7 below), very few such inferences are needed for the application of machine translation.

6.6 Synonymy and Paraphrases

The issues we are discussing in this section is whether ontological semantics can generate two different TMRs for the same input and whether different inputs with the same meaning are represented by the same TMR. The former phenomenon is synonymy, the latter, paraphrase in natural language. What we are interested in here is whether these phenomena are carried over into the TMR.

In an ontological semantic analysis system, for a given state of static knowledge resources, a given definition of the TMR format and a given analysis procedure, a given input will always yield the same TMR statement. The above means that there is no synonymy of TMRs. There is a one-to-one relationship between a textual input and a TMR statement. Sentence or text level synonymy in natural language, that is, paraphrase, will not, therefore, necessarily lead to generating a single TMR for each sentence from a set of paraphrases, unless those paraphrases are purely syntactic, as, for instance, in active, passive and middle voice variations.

The sentences *Michael Johnson won the 400m*, *Michael Johnson got the gold medal in 400m*, *Michael Johnson finished first in the 400m* and even *Michael Johnson left all his rivals behind at the finish line the 400m in Sydney* and, in fact, many others may refer to the same event without being, strictly speaking, paraphrases. The analysis system will assign different TMRs to these inputs. This is because the analysis procedure is in some sense “literal-minded” and simply follows the rules of instantiating and combining the elementary meanings from the lexicon, the ontology and the Fact DB. In defense of this literal-mindedness, do not let us forget that all the above examples do, in fact, have, strictly speaking, different meanings and deal with different facts. It is another matter that these facts are closely connected and, if they indeed all refer to the final of the 400m run at the Sydney Olympics of 2000, characterize different aspects of the same event. In terms of formal semantics, any one of these co-referring sentences conjoined with the

negation of any other yields a contradictory statement. This means that an inferential relationship holds between any two of such sentences (cf. Example 9 in Section 6.1 above).

Is it important to know that all these (and possibly other) examples refer to the same event? It might be that for the application of MT this is not that important. However, in information extraction and question answering systems, it is essential to understand that all the examples above provide the same information. So, if the question was “Who won men’s track 400m in Sydney?” any of the above examples can provide the answer. Also, if we turn these examples into questions, all of them can be input into a QA system with the same intent. It is precisely the desire to recognize that a set of questions *Did Michael Johnson win the 400? Who got the gold in men’s track 400?* and others co-refer that makes it necessary, unlike in MT, to find a way of connecting them. To accommodate such a goal in these applications, additional provisions should be furnished in the static knowledge sources. Complex events in the ontology (Section 7.1.5) and the Fact DB (Section 7.2) fit the bill. The sentences in the example above would instantiate different components of the same instance of the complex event SPORTS-RESULT in the ontology. In order for a QA system to be able to provide an answer to questions based on these sentences, the Fact DB must contain this instance of SPORTS-RESULT with its properties filled.

6.7 Basic and Extended TMRs

The input text provides initial information to be recorded in the TMR by an ontological semantic analysis system. The input sentence *John sold the blue car* results in a TMR fragment with instances of BUY, HUMAN and CAR. The latter’s COLOR property is filled with the value ‘blue.’ The instances of HUMAN and CAR fill the properties SOURCE and THEME, respectively in the instance of BUY. The important thing to realize here is that in addition to having established the above property values triggered by direct processing of the input, the system knows much more about the ontological concept instances used in this TMR, for example, that BUY has an AGENT property, among many others. While the values of the properties that have not been overtly stated in the text do not become part of the specification of an instance, they can still be retrieved from the ontology, by traversing the INSTANCE-OF relation from the instance to its corresponding concept. Thus, the system can abductively infer that the car was sold to another person from the fact that AGENT of BUY has a SEM constraint HUMAN, even though the input does not overtly mention this. In principle, this conclusion can be overridden by textual evidence.

If the blue car has been already mentioned in the text (which is likely because of the definite article in the input), then the corresponding instance of car is already available. If the instance was created by processing the input *John owned a blue Buick and a red Ford*, then the TMR already contains an instance of a car whose COLOR property is ‘blue’ and whose MAKE property is filled by BUICK. The MAKE property in the ontological concept for CAR has CAR-MANUFACTURER as its filler. The constraint from the text is more specific. Therefore, if co-reference can be established, the quality of processing will be enhanced if the more specific constraint is used.

Now, in addition to other parts of the TMR, there is another source of such constraints, the Fact DB (see Section 7.2 below), where information about remembered instances of ontological concepts is stored. So, if for some reason it is important to remember John’s Buick, then any information from any text already processed by a system or entered by a human acquirer (from the point

of view of Fact DB itself, the method of acquisition is immaterial) can provide the most specific set of constraints for a concept instance in the TMR. Once again, successful co-reference resolution is a precondition. Thus, the Fact DB may contain information about John's Buick that its model is Regal or that its model year is 1998.

The overall picture of the TMR is, then as follows. It contains frames triggered by the input sentence, where some of the property fillers come from the currently processed input, some others, from other parts of the TMR, still others, from Fact DB, and the rest, from the ontology. The overridability status of the fillers of different provenance is not the same—the constraints from the input take overall precedence, followed by constraints from the same TMR, constraints from Fact DB and constraints from the ontology, in this order.

The basic TMR contains only the first two levels of constraint—those from the current input and from other parts of the TMR. Information from the Fact DB and the ontology that was not overtly mentioned in the text should not be, if at all possible, used in generating text in a target language for the application of MT. Some other applications, such as IE and QA generally cannot avoid using the inferred information (see examples in Section 6.6 above). The TMR that contains information from outside input texts is the **extended TMR**. The inferred information is listed using the DEFAULT, SEM and RELAXABLE-TO facets (see Section 7.1.1 below for the definition), while the **basic TMR** information is stored using the VALUE facet of the corresponding property. Figure 21 is a modified version of Figure 20 to which extended TMRs, the procedures that produce it and connections with other dynamic and static knowledge sources have been added.

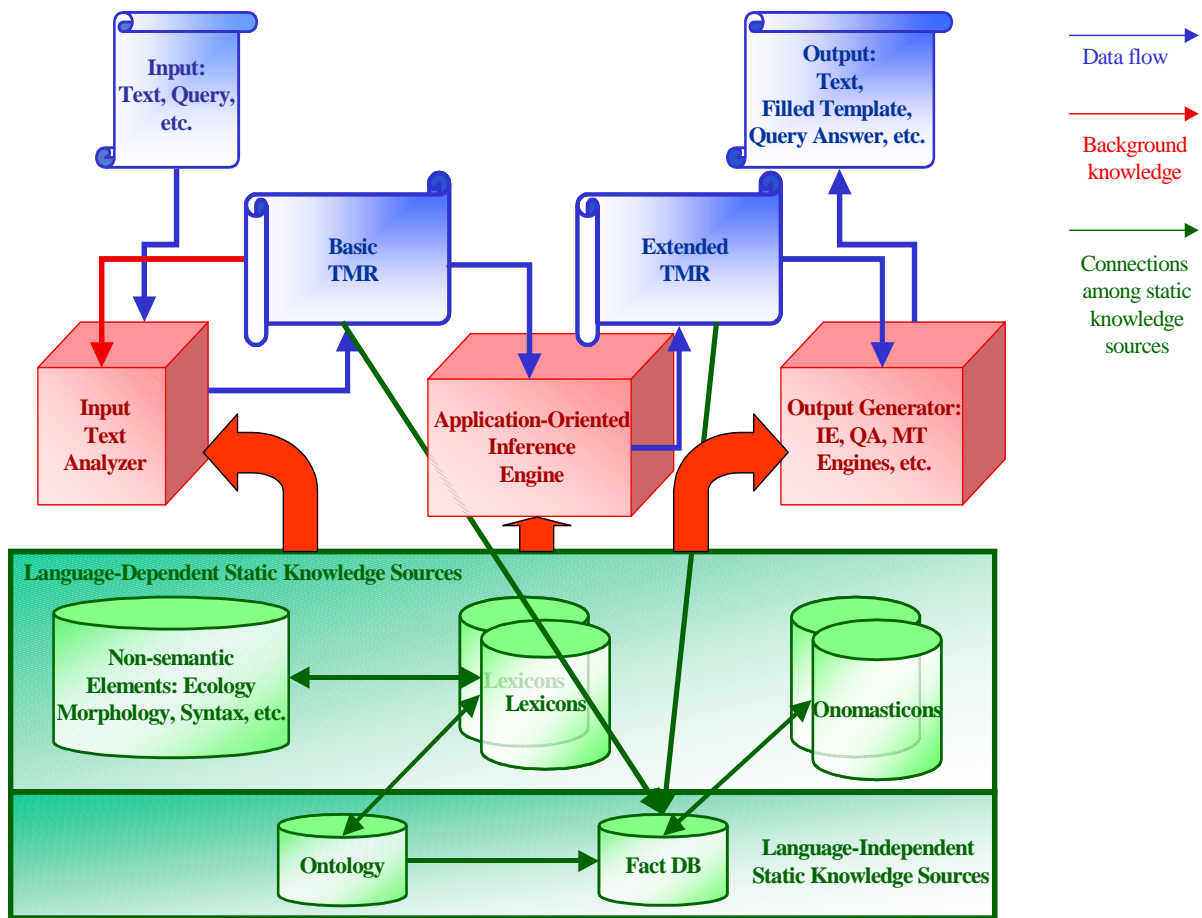


Figure 21. The Data, the Processors and the Static Knowledge Sources in Ontological Semantics II: With extended TMRs and the inference engine included.