Ontological Semantics

after Ontological Semantics by Sergei Nirenburg and Victor Raskin

by Łukasz Stafiniak

Overview

- a theory is a set of statements determining the format of descriptions of the phenomena with which the theory deals; a methodology is means to obtain the descriptions
- an integrated heterogeneous complex of theories, methodologies, descriptions and implementations
    - specific language phenomena,
    - world knowledge organization,
    - processing heuristics
    - issues relating to knowledge representation
    - and implementation system architecture
- the crucial component of success of large applications is content, not formalism

- static knowledge sources: an ontology, a fact database, a lexicon connecting an ontology with a natural language and an onomasticon, a lexicon of names
- morphological and syntactic components largely independent of the central ontosemantic component

A Model of Language Communication Situation

Relevant Components of Intelligent Agent's Model:

- Knowledge about the world, which we find useful to subdivide into:
    - an ontology, which contains knowledge about types of things (objects, processes, properties, intentions) in the world; and
    - a fact database, an episodic memory module containing knowledge about instances (tokens) of the above types and about their combinations; a marked recursive subtype of this knowledge is a set of mental models of other agents
- Knowledge of natural language(s), including, for each language:
    - ecological, phonological, morphological, syntactic and

prosodic constraints;

- the ecology of a language includes information about punctuation and spelling conventions, representation of proper names, dates, numbers, etc.

  - semantic interpretation and realization rules and constraints, formulated as mappings between lexical units of the language and elements of the world model of the producer;
  - pragmatics and discourse-related rules that map between modes of speech and inter-agent situations, on the one hand, and syntactic and lexical elements of the meaning representation language, on the other;

- Emotional states that influence the "slant" of discourse generated by an agent.
- An agenda of active goal and plan instances (the intentional plane of an agent).

A goal of the discourse producer can be to modify **any** of these components.

Given an input text, the discourse consumer must first attempt to match the lexical units comprising the text, through the mediation of a special lexicon, with elements in the consumer's model of the world. To facilitate this, it will have to analyze syntactic dependencies among these units and determine the boundaries of syntactic constituents. The next step is filtering out unacceptable candidate readings through the use of selectional restrictions, collocations and special heuristics, stored in the lexicon. The consumer must then also resolve the problems of co-reference by finding referents for pronouns, other deictic lexical units and elliptical constructions. Furthermore, information on text cohesion and producer attitudes has to be determined, as well as, in some applications, the goals and plans that lead the producer to produce the text under analysis.

In other words the meaning representation of a text is derived through:

- establishing the lexical meanings of individual words and phrases comprising the text;
- disambiguating these meanings;
- combining these meanings into a semantic dependency structure covering
    - the propositional semantic content, including causal, temporal and other relations among individual statements;
    - the attitudes of the speaker to the propositional content; and
    - the parameters of the speech situation;
- filling any gaps in the structure based on the knowledge instantiated in the structure as well as on ontological knowledge.

It follows that text meaning is, on this view, a combination of

- the information directly conveyed in the NL input;
- the (agent-dependent and context-dependent) ellipsis-removing (lacuna filling) information which makes the input self-sufficient for the computer program to process;
- pointers to any background information which might be brought to bear on the understanding of the current discourse,
- records about the discourse in the discourse participants' fact database.
- It includes detecting and representing a text component as an element of a script/plan, or determining which of the producer goals are furthered by the utterance of this text component.

Toward Constraint Satisfaction Architectures

The left hand sides of the text meaning representation rules can draw on the entire set of knowledge sources in comprehensive NLP processing: the lexicons, the ontology, the fact database, the text meaning representation and the results of ecological, morphological and syntactic processing.

- **blackboard systems** usually use the agenda mechanism: a queue of knowledge source instantiations (KSI) each corresponding roughly to rules=situation/action pairs (situation=a combination of constraints); control heuristics, metalevel rules
- a graph of choices resulting in an implicit ordering of KSIs established automatically through the availability of constraints.

Soft constraints, by introducing a confidence measure for all decisions; and developing procedures for relaxing the constraints based, among other things, on the confidence values of the knowledge used to make decisions.

The Major Dynamic Knowledge Sources

A comprehensive text analyzer consists of:

- a tokenizer that treats ecological issues such as all special characters and strings, numbers, symbols, differences in fonts, alphabets and encodings as well as, if needed, word boundaries (this would be an issue for languages such as Chinese);
- a morphological analyzer that deals with the separation of lexical and grammatical morphemes and establishing the meanings of the latter;
- a semantic analyzer, including possibly:
  - a lexical disambiguator that selects the appropriate word sense from the list of senses listed in a lexicon entry;
  - a semantic dependency builder that constructs meanings of clauses;

- a discourse-level dependency builder that constructs the meanings of texts;
- a module that manages the background knowledge necessary for the understanding of the content of the text; this module centrally involves processing reference and co-reference;
- a module that determines the goals and plans of the speaker, hearer and the protagonists of the text;
- a module that tracks the attitudes of the speaker to the content of the text;
- a module that determines the parameters (indices) of the speech situation, that is, the time, the place, the identity and properties of the speaker and the hearer, etc.; and
- a module that determines the style of the text.

Text generators can include the following modules:

- a content specification module that determines what must be said; this module sometimes includes
  - a communicative function specification module
  - an interpersonal function module how much of the input can be assumed to be already known by the hearer;
- a text structure module that organizes the text meaning by organizing the input into sentences and clauses and ordering them;
- a lexical selection module that takes into account not only the semantic dependencies in the target language but also idiosyncratic relationships such as collocation;
- a syntactic structure selection module;
- a morphological realizer for individual words;
- the clause- and word-level linearizer.

The prototypical ontological semantic system is a learning system: in order to enhance the quality of future processing, the results of successful text analysis are not only output in accordance with the requirements of a particular application but are also recorded and multiply indexed in the fact database.

The Static Knowledge Sources

The static knowledge sources of a comprehensive NLP system include:

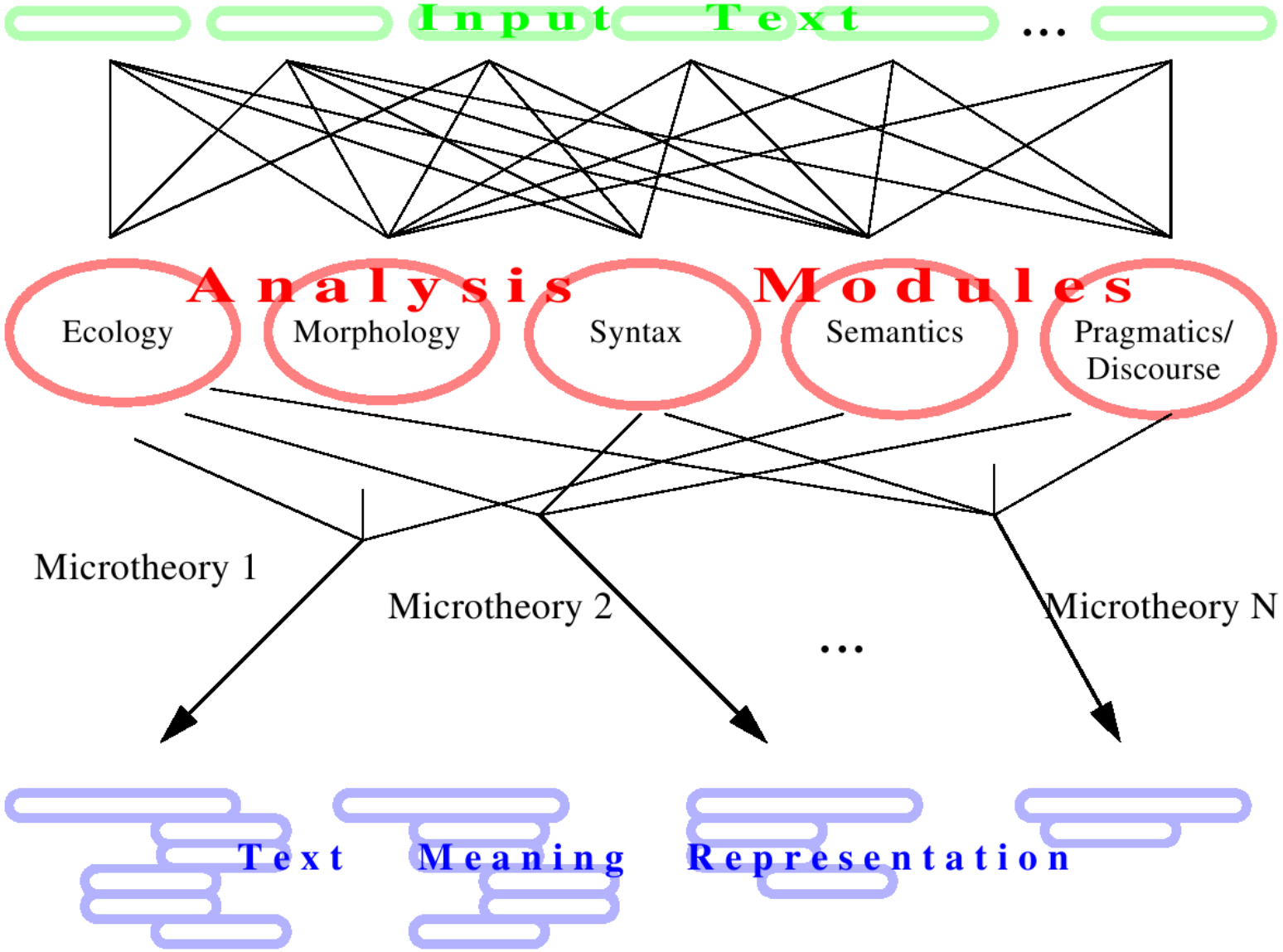
- An ontology, a view of the intelligent agent's world, including knowledge about types of things in the world; the ontology consists of
  - a model of the physical world;
  - a model of discourse participants ("self" and others) including knowledge of the participants' goals and static attitudes to elements of the ontology and remembered instances of ontological objects; and
  - knowledge about the language communication situation;
- A fact database containing remembered instances of events and objects; the fact database can be updated in two ways: either as a result of the operation of a text analyzer, when the facts

(event and object instances) mentioned in an input text are recorded or directly through human acquisition;

- A lexicon and an onomasticon for each of the natural languages in the system; information required for analysis and generation; entries for polysemic lexical items include knowledge supporting lexical disambiguation, this is also used to resolve synonymy in lexical selection during generation; the entries also include information for the use by the syntactic, morphological and ecological dynamic knowledge sources;
- A text meaning representation formalism;
- Knowledge for semantic processing (analysis and generation), including
    - structural mappings relating syntactic and semantic dependency structures;
    - knowledge for treatment of reference (anaphora, deixis, ellipsis);

- knowledge supporting treatment of non-literal input (including metaphor and metonymy);
- text structure planning rules;
- knowledge about both representation (in analysis) and realization (in generation) of discourse and pragmatic phenomena, including cohesion, textual relations, producer attitudes, etc.

The Concept of Microtheories



When meaning representation rules are bunched according to a single principle, they become realizations of a microtheory.

Lexical Semantics

Automating (or augmenting) lexical acquisition:

- using paradigmatic lexical relations of a lexeme, such as **synonymy, antonymy, hyperonymy and hyponymy** to specify the lexical meaning of another lexeme;
- using a broader set of paradigmatic relations for the above task, such as the one between an organization and its leader (e.g., company: commander, department: head, chair, manager);
- using syntagmatic lexical relations for the above task, for instance, those between an object and typical actions involving it (e.g., key: unlock, lock,...).

Lexical rules in generative lexicon derive lexicon entries of new senses from old senses at runtime. In OntoSem, the acquisition methodology allows for the application of lexical rules also at acquisition time. For small-coverage/many-exceptions rules, senses are just enumerated.

An ontology (in OntoSem, opposite to "language grounding" philosophy of MultiNet) is seen not as a natural language but rather as a language-neutral body of knowledge about the world (or a domain) that:

- is a repository of primitive symbols used in meaning representation;
- organizes these symbols in a tangled subsumption hierarchy;
- further interconnects these symbols using a rich system of semantic and discourse-pragmatic relations defined among the concepts.

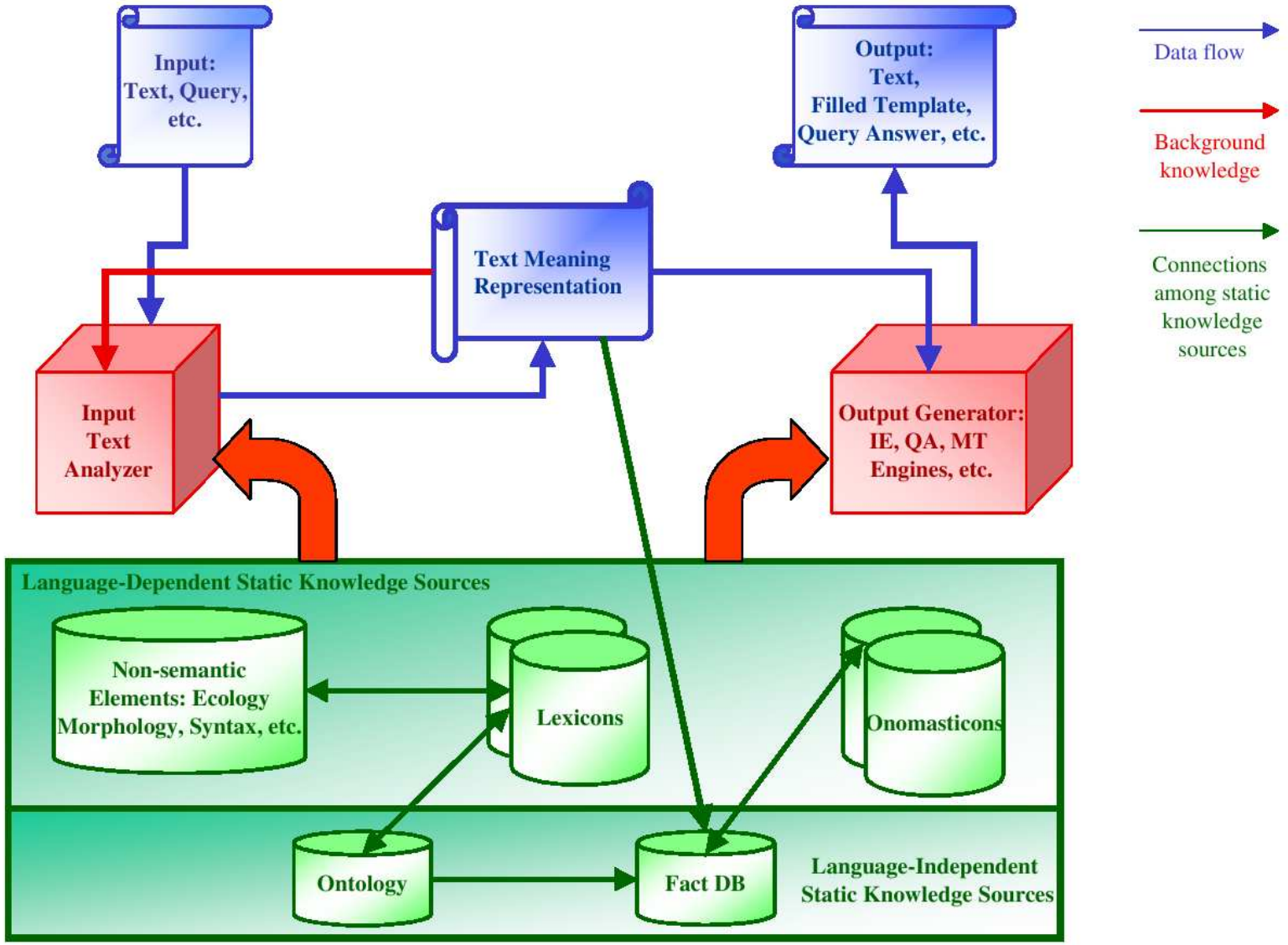
OntoSem proposes a metalanguage

- with ontology – world model – as a central component;
- common across languages;
- common for **lexical** and **sentential semantics** and glueing them.

OntoSem stresses the role of formulating right **concepts** in its "rigid" (not learnable I think) ontology. The concepts can be ambiguous to avoid unlimited fine-graining; the more unambiguous ontology the more polysemous lexicon.
In practical applications, ontologies seldom, if ever, are used as the only knowledge resources. In the representative application of knowledge-based MT, for example, the ontology is used to:

- supply the language for explaining lexical meanings, which are recorded in the lexicons of particular languages;
- provide the contentful building blocks of a text meaning representation language;
- provide the heuristic knowledge for the dynamic knowledge resources such as semantic analyzers and generators.

Architecture



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 (also recorded as property fillers in Fact DB frames). The Fact DB also contains other, unnamed, concept instances.
OntoSem uses a standard **frame-based representation**.

Text Meaning Representation

- Text meaning does not include presuppositions and implications (consider machine translation), (esp. probable but not necessary inferences), but recording them explicitly in a well-indexed manner for future retrieval is essential for supporting a variety of computational applications.

Examples of Ontology Entry and Lexicon Entry

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

```
say-v1
 syn-struc
    root  say          ; as in "Spencer said a word"
    cat   v
  1 subj  root  var1
          cat   n
    obj   root  var2
          cat   n
    root   say          ; as in "Spencer said that it rained"
    cat    v
  2 subj   root  var1
          cat   n
    comp  root  var2
 sem-struc
  1 2  inform                    ; synt. structs share sem. struct
       agent  value  ^var1       ; "^" is read "the meaning of"
       theme  value  ^var2
```

Creating a TMR

If the fillers are not available directly in the input, there are special procedures to try to establish them. If this fails for an obligatory filler, the special filler UNKNOWN is used.

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

The lexicon entry for `say` essentially states that the meaning, `^var1`, of the syntactic subject of `say`, 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.

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

AMOUNT is measured on a scale $[0,\ 1]$, with *major, large, great, much, many* $= >0.7$ and *enormous, huge or gigantic* $= >0.9$.
The slots (also *unknown*) are constrained, e.g. the THEME of `import-1` and `manufacture-1` is constrained to GOODS.

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-1inform-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 Nature and Format of TMR (as in "Mikrokosmos")

- Ontological vs. semantic parameters: aspect, modality, time and other proposition-level parameters is defined for concept instances, not ontological concepts themselves.

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

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

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

TMR-TIME ::= set
    element-type proposition-time
    cardinality $\geqslant 1$

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

TIME-EXPR ::=
$\ll | < | > | \gg | \geqslant | \leqslant | = | \neq$
{ABSOLUTE-TIME | RELATIVE-TIME}

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

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

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

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

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

REFERENCE ::= SET
 element-type SET
 element-type concept-instance
 cardinality $\geqslant 1$

SET ::=

set
 element-type: concept | concept-instance
 cardinality: [$<$ | $>$ | $\geqslant$ | $\leqslant$ | $\neq$] integer
 complete: boolean
 excluding: [concept | concept-instance]*
 elements: concept-instance*
 subset-of: SET

Representing modifier-modified relation:

- if modifier expresses a property defined for the ontological concept corresponding to its head, it sets the property's value (both are components of the meaning of the modifier)
- 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.

Example 2. *The Iliad was written not by Homer but by another man with the same name.* $\Longrightarrow$ (processing ellipsis) *Iliad was not written by Homer, Iliad was written by a different man whose name was Homer.*

author-event-1
agent value Homer
theme Iliad

modality-1
scope author-event-1
modality-type epistemic
modality-value 0

author-event-2
agent value human-2
name Homer
theme value Iliad

co-reference-3
Homer human-2

modality-2
scope co-reference-3
modality-type epistemic
modality-value 0

Example 3. *Was it Arsenal who won the match?*

win-33

agent	value	Arsenal
theme	value	sports-match-3

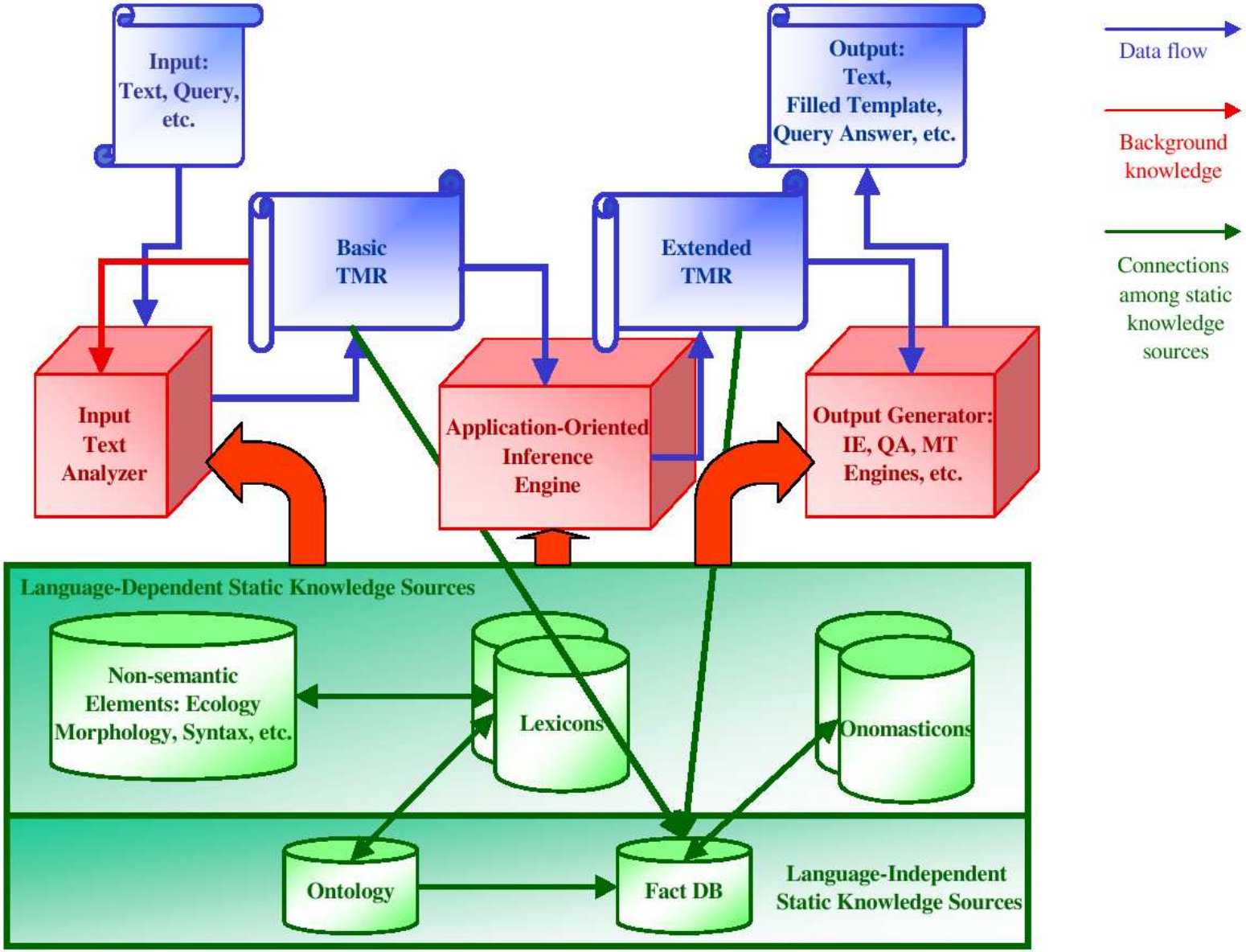
request-information-13

theme	value	win-33

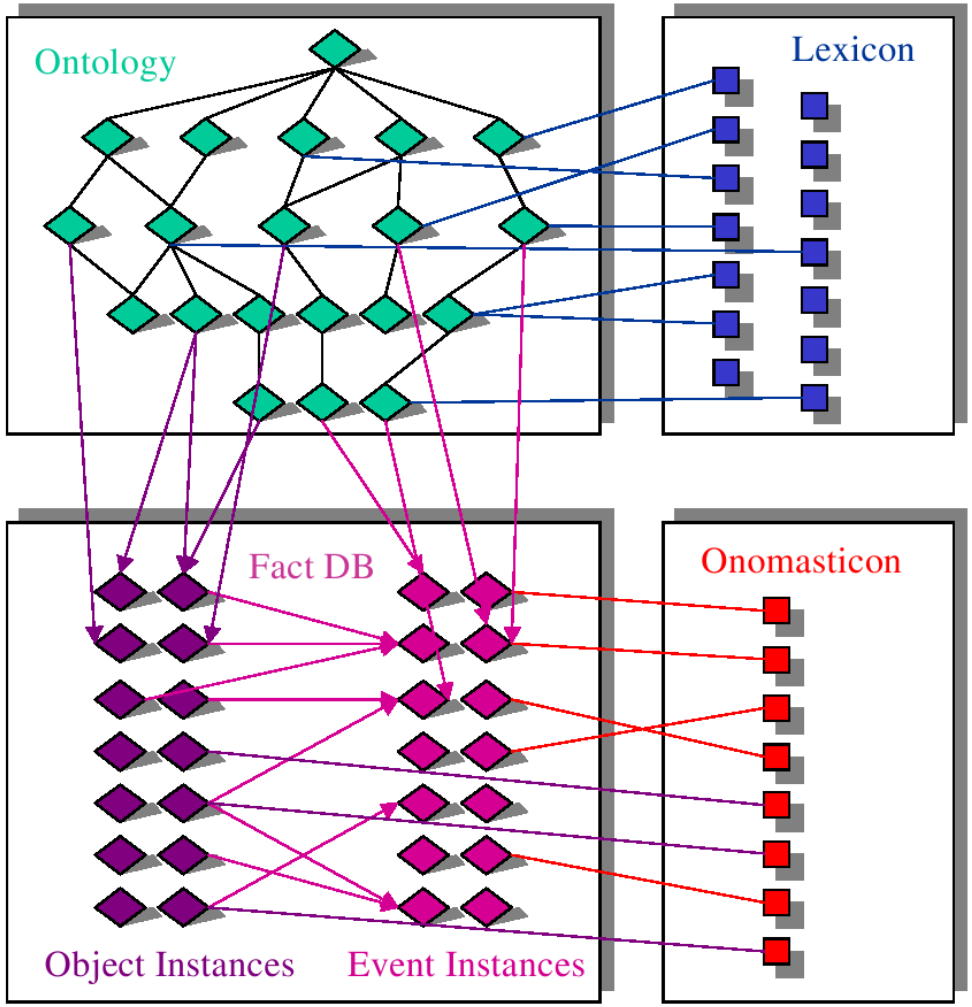
modality-11

type	salience
scope	win-32.theme
value	1

- indirect speech acts are not part of TMR: can be inferred later
- the basic TMR is "literal-minded", only slots for overtly stated facts are filled (only the first two levels) – targeted at Machine Translation; uses the VALUE facet
- the extended TMR (all levels) – targeted at Information Extraction and Question Answering; uses also DEFAULT, SEM and RELAXABLE-TO facets
- levels, in order of precedence (of overriding):
    - currently processed input,
    - other parts of the TMR,
    - Fact DB,
    - the ontology.


Input:
Text, Query,
etc.
Output:
Text,
Filled Template,
Query Answer, etc.
Data flow
Background
knowledge
Connections
among static
knowledge
sources
Basic
TMR
Extended
TMR
Input
Text
Analyzer
Application-Oriented
Inference
Engine
Output Generator:
IE, QA, MT
Engines, etc.
Language-Dependent Static Knowledge Sources
Non-semantic
Elements: Ecology
Morphology, Syntax, etc.
Lexicons
Onomasticons
Ontology
Fact DB
Language-Independent
Static Knowledge Sources

The Static Knowledge Sources: Ontology, Fact Database and Lexicons



The Ontology

An ontological model must define a large set of generally applicable categories for world description:

- 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.

ONTOLOGY ::= CONCEPT+

CONCEPT := ROOT | OBJECT-OR-EVENT | PROPERTY

ROOT ::= **ALL** DEF-SLOT TIME-STAMP-SLOT SUBCLASSES-SLOT
(DEF-SLOT and TIME-STAMP-SLOT are not really used by applications.)

OBJECT-OR-EVENT ::= CONCEPT-NAME DEF-SLOT TIME-STAMP-SLOT
ISA-SLOT [SUBCLASSES-SLOT] [INSTANCES-SLOT] OTHER-SLOT*

PROPERTY ::= RELATION | ATTRIBUTE | ONTOLOGY-SLOT

OTHER-SLOT ::= RELATION-SLOT | ATTRIBUTE-SLOT

RELATION-SLOT ::= RELATION-NAME FACET CONCEPT-NAME+

ATTRIBUTE-SLOT ::= ATTRIBUTE-NAME FACET {number | literal}+

FACET ::= value | sem | default | relaxable-to | not | default-measure | inv | time-range | info-source

- **value**: it may be the instance of a concept, a literal symbol, a number, or another concept. In the ontology, in addition to ontology slots, the VALUE facet is used to carry factual truths.
- **sem**: either another concept or a literal, number, or a scalar range. A selectional restriction on the filler of the slot (a "type"). An abductive (weak) constraint: can be violated in certain cases.
- **default**: the most frequent or expected constraint, always a subset of the filler of the SEM facet.
- **relaxable-to**: a bound on violations of the SEM facet; the program first attempts to perform the match on the selectional restrictions in DEFAULT facet fillers, then the SEM facets and, failing that, RELAXABLE-TO facets.
- **not**: "exclusion set"
- **default-measure**: a measuring unit for the number or numerical range.

- **inv**: marks the fact that the filler was obtained by traversing an inverse relation from another concept.
- **time-range**: a facet used only in facts, that is, concept instances; specifies the temporal boundaries within which the information listed in the fact is correct.

pay

definition	value	"to compensate sb for goods or services rendered"
agent	sem	human
	relaxable-to	organization
theme	default	money.amount
	sem	commodity
	relaxable-to	event
patient	sem	human
	relaxable-to	organization

ONTOLOGY-SLOT ::= ONTOLOGY-SLOT-NAME DEF-SLOT TIME-STAMP-SLOT ISA-SLOT [SUBCLASSES-SLOT] DOMAIN-SLOT ONTO-RANGE-SLOT INVERSE-SLOT

DEF-SLOT ::= DEFINITION value "an English definition string"

TIME-STAMP-SLOT ::= time-stamp value time-date-and-username+

ISA-SLOT ::= IS-A value { ALL | CONCEPT-NAME+ | RELATION-NAME+ | ATTRIBUTE-NAME+ }

SUBCLASSES-SLOT ::= subclasses value {CONCEPT-NAME+ | RELATION-NAME+ | ATTRIBUTE-NAME+}

INSTANCES-SLOT ::= instances value instance-name+

INSTANCE-OF-SLOT ::= instance-of value concept-name+

DOMAIN-SLOT ::= domain sem concept-name+

INVERSE-SLOT ::= inverse value relation-name

ONTO-RANGE-SLOT ::= REL-RANGE-SLOT | ATTR-RANGE-SLOT

RELATION ::= RELATION-NAME DEF-SLOT TIME-STAMP-SLOT ISA-SLOT
[SUBCLASSES-SLOT] DOMAIN-SLOT REL-RANGE-SLOT INVERSE-SLOT

ATTRIBUTE ::= ATTRIBUTE-NAME DEF-SLOT TIME-STAMP-SLOT ISA-SLOT
[SUBCLASSES-SLOT] DOMAIN-SLOT ATTR-RANGE-SLOT

REL-RANGE-SLOT ::= RANGE SEM CONCEPT-NAME+

ATTR-RANGE-SLOT ::= RANGE SEM { number | literal }*

- **definition**: for human consumption during the acquisition
- **time-stamp**: an update log
- **is-a**: mandatory for all concepts except roots (**all**); instances don't have it
- **subclasses**: mandatory except for leaves; instances don't count as "ontological children"
- **instances**: "backpointers" from concepts to the Fact DB
- **instance-of**: from TMR and FactDB instances to Ontology concepts
- **inverse**: points to the inverse relation of a given relation
- abstract scales: e.g. *hot* $= [0.75, 1]$, absolute scale for *water* is $[0, 100]°C$ and for *bath* it is, say, $[20, 50]°C$, so *hot water* $= [75, 100]°C$, *hot bath* $= [42.5, 50]°C$

Inheritance

When X is-a Y,

- All slots that have not been overtly specified in X, with their facets and fillers, but are specified in Y, are inherited into X.
- ONTOLOGY-SLOTs (IS-A, SUBCLASSES, DEFINITION, TIME-STAMP, INSTANCE-OF, INSTANCES, INVERSE, DOMAIN, RANGE) are excluded from this rule.
- If a slot appears both in X and Y, then the filler from X takes precedence over the fillers from Y.
- Use the filler NOTHING to locally block inheritance on a property. This has the same effect as removing the slot from the concept. The slot an be explicitly reintroduced in descendants.
- Block the inheritance of a filler that is introduced through the NOT facet. E.g. we can put the AGENT slot in EVENT and put a SEM NOTHING in PASSIVE-COGNITIVE-EVENT and INVOLUNTARY-PERCEPTUAL-EVENT. (like a default slot)

Case Roles for Predicates

- Agent
- Theme
- Patient
- Instrument
- Source
- Destination
- Location
- Path
- Manner

See chapter 7, pages 175-180 for definitions and examples.
OntoSem tries to minimize the opportunities to paraphrase a TMR (i.e. tries to provide unique representations).

Complex Events (aka Scripts, Scenarios)

- events have preconditions and postconditions (EFFECTs)
- Complex events are represented in ontological semantics using the ontological property HAS-PARTS.
- The notion of ontological instance is introduced. They are not indexed in the Fact DB. They appear in appropriate slots of complex events and their fillers are all references to fillers of other ontological instances within the same complex event or the complex event itself.
- Reification is a mechanism for allowing the definition of properties on properties by elevating slots in frames to the level of a free-standing concept frame (e.g. to avoid introducing a concept of DRIVER if it could always be referred to as DRIVE.AGENT)

See chapter 7, page 186.

Lexicon

superentry ::=
    ORTHOGRAPHIC-FORM: “form”
    ({syn-cat}: <lexeme> *) *
lexeme ::=
    CATEGORY: {syn-cat}
    ORTHOGRAPHY:
        VARIANTS: “variants”*
        ABBREVIATIONS: “abbs”*
    PHONOLOGY: “phonology”*
    MORPHOLOGY:
        IRREGULAR-FORMS: (“form” {irreg-form-name})*
        PARADIGM: {paradigm-name}
        STEM-VARIANTS: (“form” {variant-name})*
    ANNOTATIONS:
        DEFINITION: “definition in NL” *
        EXAMPLES: “example”*
        COMMENTS: “lexicographer comment”*
        TIME-STAMP: {lexicog-id date-of-entry}*
    SYNTACTIC-FEATURES: (feature value)*
    SYNTACTIC-STRUCTURE: f-structure
    SEMANTIC-STRUCTURE: lex-sem-specification

```
buy-v1
        cat             v

        morph           stem-v          bought v+past
                                        bought v+past-participle

        anno            def             "when A buys T from S, A acquires possession of T previously owned
                                        by S, and S acquires a sum of money in exchange"
                        ex              "Bill bought a car from Jane"
                        time-stamp      dha; 12-13-94                   ;the acquirer and the date

        syn             syn-class       trans           +               ;redundant with SYN-STRUC; may be
```

```
                                                        ;useful for some applications
syn-struc
                root            buy
                subj            root            $var1
                                cat             n
                obj             root            $var2
                                cat             n
                oblique         root            from
                                cat             prep
                                opt             +
                                obj             root            $var3
                                                cat             n

sem-struc
                buy
                                agent           value           ^$var1
                                                sem             HUMAN
                                theme           value           ^$var2
                                                sem             OBJECT
                                source          value           ^$var3
                                                sem             HUMAN
```

```
acquire-v3
        cat             v
        anno            def             "when company A buys company T"
                        ex              "Bell Atlantic acquired GTE"
        syn-struc
                        root            acquire
                        subj            root            $var1
                                        cat             n
                        obj             root            $var2
                                        cat             n
        sem-struc
                        buy
                                        agent           value   ^$var1
                                                        sem     CORPORATION
                                        theme           value   ^$var2
                                                        sem     CORPORATION
                                        source          value   ^$var2.OWNED-BY
                                                        sem     HUMAN
```

Actually, instead of `sem-struc buy source value ^var2.owned-by`, it should be encoded in the ontology as `buy source default theme.owned-by`.

```
big-adj1
        cat             adj
        syn-struc       1       root            $var1
                                cat             n
                                mods            root        big
                        2       root            big
                                cat             adj
                                subj            root        $var1
                                                cat         n
        sem-struc
                        1 2     size-attribute
                                                domain      value       ^$var1
                                                            sem         physical-object
                                                range       value       > 0.75
                                                            relaxable-to > 0.6
```

The first subcategorization pattern in the SYN-STRUC zone describes the Adj-N construction (*big thing*); the second describes the N-Copula-Adj construction (*thing is big*).

In the verb entries the main concept refers to the syntactic constituent corresponding to the lexeme itself; in the entries for modifiers, the main concept refers to the syntactic constituent marked as $var1, the head of the modifier.

good-adj1

cat	adj			
syn-struc				
1	root	var1		
	cat	n		
	mods	root	good	
2	root	var0		
	cat	adj		
	subj	root	var1	
		cat	n	
sem-struc				
modality	type		evaluative	
	value		value	> 0.75
			relaxable-to	> 0.6
	scope		^var1	
	attributed-to		*speaker*	

avoids the problem of determining the salient property by shifting the description to a coarser grain size, that is, scoping not over a particular property of an object or event but over an entire concept. In MT this approach "gambles" on the availability across languages of a "plastic" adjective corresponding to the English good.

Plasticity of meaning affects also the analysis of nominal compounds. E.g., *the IBM lecture*, is even more difficult than analyzing adjectival modification because in the former case there is no specification of any property on which the connection can be made. IBM may be the filler of the properties OWNED-BY, LOCATION, THEME as well as many others.

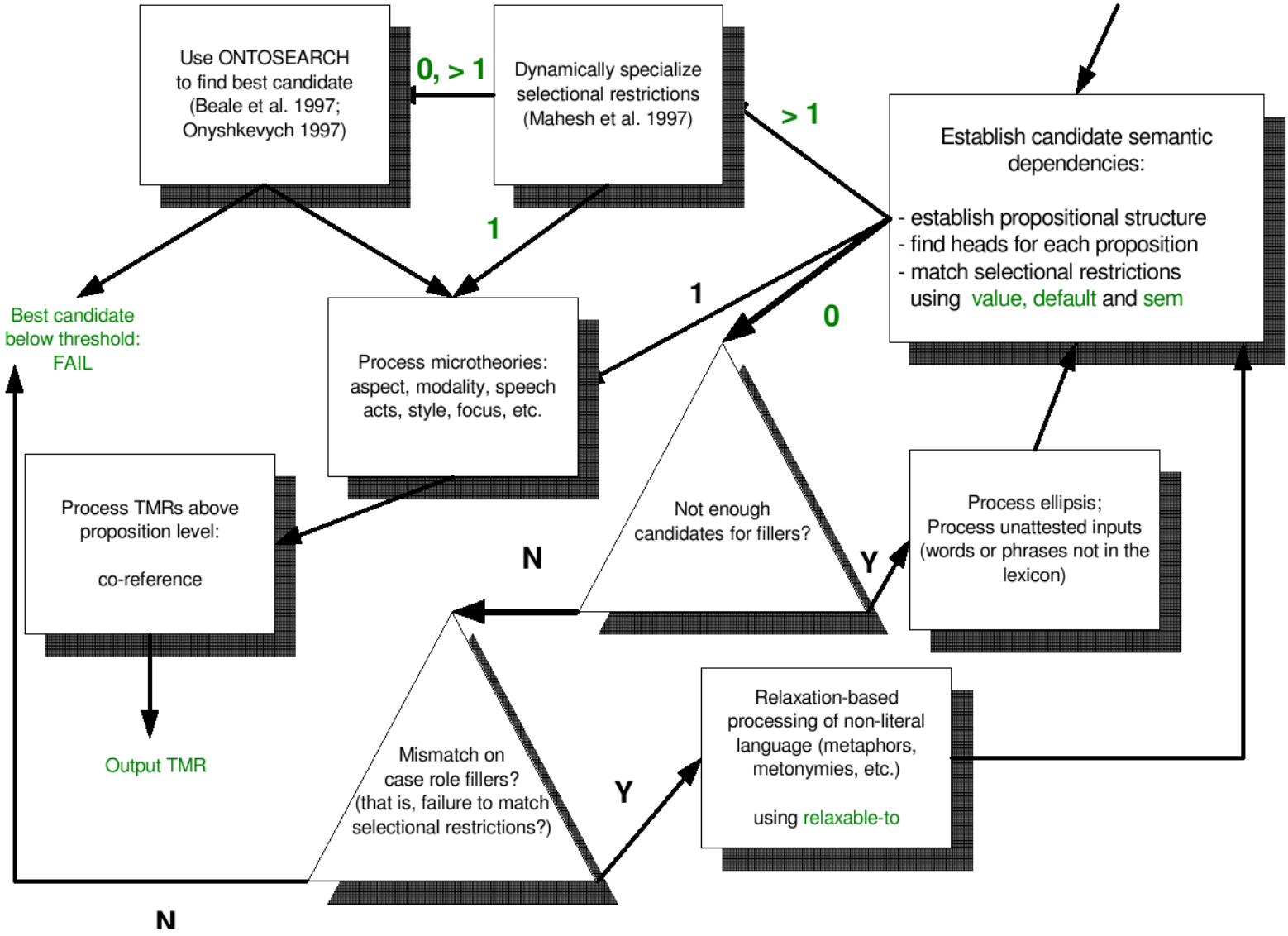
try-v3

syn-struc			
	root	try	
	cat	v	
	subj	root	$var1
		cat	n
	xcomp	root	$var2
		cat	v
		form	OR infinitive gerund
sem-struc			
	set-1	element-type	refsem-1
		cardinality	>=1
	refsem-1	sem	event
		agent	^$var1
		effect	refsem-2
	modality		
		type	epiteuctic
		scope	refsem-2
		value	< 1
	refsem-2	value	^$var2
		sem	event

$< 1 =$ lack of success

Many closed-class lexemes enjoy special treatment: personal pronouns, determiners, possessives and other deictic elements, such as *here* or *now*, as well as copulae are treated as triggers of reference-finding procedures; some conjunctions may introduce discourse relations in the TMR; numerals and some special adjectives, e.g., *every* and *all*, characterize set relations.

Basic Processing



- simplest heuristic: lexical entries containing event instances = TMR propositions (events fill the heads)
- *fetch* triggers two event instances: GO and BRING
- heads are the elements not dependent on others, and not parametric (i.e. not specifying aspect, time, modality etc.)
- "free-standing" object instances as heads: *The car is blue. My son John is a teacher.*

```
human-j
   name  value  John
son-of-k
   domain  value  *speaker*
   range   value  human-1
teach-m
   aspect
      iteration  multiple
```

```
car-i
   color  value  blue
modality-1
   type   saliency
   scope  car-i.color
   value  1

co-reference-n
   human-j  human-1  teach-m.agent
```

Semantics-driven selection of filler

- *The IBM lecture will take place tomorrow at noon.*
- *John went to the IBM facility to give a lecture. The IBM lecture started at noon.*
- *IBM sponsored a series of lectures on early computer manufacturers. Naturally, the IBM lecture was the most interesting.*

Disambiguation

- mostly done through selectional restrictions ("type checking") given in the ontology and refined in the lexicon (not only between a head and its arguments, but also between arguments, and on other properties)
    - when restrictions are based on context by a unification process, they're called dynamic (type-inference-like)
- marker passing and spreading activation are effective on well-designed and sparse networks but become less and less effective as the degree of connectivity increases
    - problem with semantically "adversarial" paths
    - statistical methods based on sense-tagged corpus analysis are subject to the same limitations as the network search methods: in a sufficiently general corpus, ample collocations of word senses may lead to irrelevant interference in sense disambiguation

- basic disambiguation method checks selectional constraints exhaustively, using a very efficient search mechanism **Hunter-Gatherer**, based on constraint satisfaction, branch-and-bound, and solution synthesis methods
- to process dynamic selectional restrictions, we introduce the Context Specialization Operator (CSO): If a sense P is selected for a word w, and the rest of the word senses in the environment satisfy the constraints on P, examine the constraints on children of P; if exactly one child C of P satisfies the constraints, then infer that the correct sense of w is C; apply the constraints on C to other words
- when still left with several senses, try the **lowest common ancestor** for meanings if it is not too general

- implemented for semantic analysis in a Spanish-English MT system (based on the Mikrokosmos OntoSem); employed an ontology of 5,000 concepts, where each node had an average connectivity of 16, a Spanish lexicon of about 37,000 word senses mapped them to nodes in this network
- Ontosearch is a network-distance method that uses network semantics, uses a state transition table to assess the appropriate cost for traversing an arc (based on the current path state) and to assign the next state for each candidate path being considered. (The weight assignment transition table has about 40 states, and has individual treatment for 40 (out of 300) types of arcs.

No Parse

- relaxing from DEFAULT to SEM to RELAXABLE-TO
  (e.g. sortal incongruity: *The baby ate a piece of paper*)

- metonymy cannot be derived dynamically in current OntoSem, it is treated with an extended RELAXABLE-TO:

play-musical-instrument
  theme sem musical-composition

relaxable-to **match** human-1
expansion musical-composition
composed-by value human-1.name

- metaphors are currently treated as metonymy, but ideally e.g. an event would be searched having the required types for arguments and being somehow "semantically related" (e.g. using Ontosearch) *Mary won an argument with John. The ship plowed the waves.*

For unattested input (words lacking in the lexicon and onomasticon), first heuristics that recognize proper (e.g. company) names are used, then morphological and syntactical analysis tries to extract as many properties as possible, to place the word into the TMR; a tentative lexical definition for the word is created; *Fred locked the door with the kheegh*

lock-event-6

agent	value	human-549
theme	value	door-23
instrument	value	artifact-71
...		

kleegh-n1

syn-struc

root kleegh

cat n

sem-struc

artifact

instrument-of value lock-event

Failure to match a selectional restriction due to the lack of lexical material in the input to fill a case role, signals the need for processing semantic ellipsis.

- *John shaved*: for reflexive verbs we record that the required NP fills both the AGENT and the PATIENT.
- *I finished the book, Mary enjoyed the movie, Mary enjoyed the cake*: these verbs require an EVENT as a filler for THEME; READ, SEE and INGEST are DEFAULT fillers for THEME-OF properties of *book, movie* and *cake*, providing the missing event.
  fast motorway: the missing event DRIVE is recovered as DEFAULT facet of LOCATION-OF property on the concept ROAD.
- *Mary enjoyed the lizard*: no DEFAULT value for THEME-OF, so the lowest common ancestor for SEM values is taken; similar to treatment of unattested verbs: semantics of the EVENT realized by the (elided or unattested) verb is determined by the inverse case role properties (THEME-OF, INSTRUMENT-OF, AGENT-OF).

Beyond Basic Semantic Dependencies

- Aspect: PHASE – BEGIN, CONTINUE, END, INSTANT; ITERATION – num, MULTIPLE
- Duration (momentary vs. prolongued) and telicity (resultative vs. non-resultative) dropped from current OntoSem
- phasal verbs: *begin, cease, commence, stop, finish, desist from, carry on, keep, continue*, etc.
- phasal value can also be contributed by a closed-class lexical morpheme (either free, a preposition or a particle, or bound, an affix)

begin-v2

```
syn-struc
   root     begin
   cat      v
   subj     root   var1
            cat    n
   xcomp    root   var2
            cat    v
            obj    root   var3
                   opt    +
sem-struc
   event
            agent  value  ^var1
            theme  value  ^var3
   aspect
            phase  begin
```

drink-v23

```
syn-struc
   root     drink
   cat      v
   subj     root   var1
            cat    n
   obj      root   var2
            cat    n
            opt    +
   oblique  root   up
sem-struc
   ingest
            agent     value  ^var1
            theme     value  ^var2
                      sem    liquid
   aspect
            phase     end
            iteration 1
```

wednesday-n1

```
syn-struc
 root  wednesday
 cat   n
sem-struc
 time    get-proposition-time
 aspect  iteration     1
```

wednesday-n2

```
syn-struc
 1  root     wednesday
    cat      n
    mods     root        OR every each
 2  root     wednesday
    cat      n
    number   plural
 3  root     on
    cat      prep
    object   root        wednesday
             cat         n
             number      plural
sem-struc
 1 2 3  wednesday
        aspect      iteration  multiple
```

Modality

Modal verbs: *plan, try, hope, expect, want, intend, doubt, be sure, like (to), mean, need, choose, propose, want, wish, dread, hate, loathe, love, prefer, deign, disdain, scorn, venture, afford, attempt, contrive, endeavor, fail, manage, neglect, undertake, vow, envisage*.

modality

type	epistemic | epiteuctic | deontic | volitive | potential | evaluative | saliency
attributed-to	*speaker*
scope	<any TMR element>
value	[0.0, 1.0]
time	time

- epistemic *does not believe that* – value 0, *believes that possibly* – value 0.6, *believes that* – value 1, *estimates* – value [0.8,0.9]

  with no reference to beliefs, this modality scopes with value 1

- epiteuctic refers to the degree of success in attaing; *fail, neglect, omit, try, attempt, succeed, attain, accomplish, achieve, almost, nearly, practically*
- deontic: obligation and permission;
- volitive: desirability
- potential: ability of the agent to perform an action
- evaluative: adjectives *good, bad*, verbs *like, admire, appreciate, praise, criticize, dislike, hate, denigrate*, etc.
- saliency: expresses the importance that the speaker attaches to a component of text meaning, usually scope a part of a proposition; *important / unimportant* and their synonyms, but also the focus / presupposition (or topic / comment, or given / new, or theme / rheme) distinction, and to specify *wh*-questions

Supraprepositional Level: reference, discourse, style

- co-reference: reference to the same independent (i.e. object or event) concept instance
- the same language techniques are used to mark identical property values across instances
- single references and co-reference chains established as a result of text-level (standard) reference resolution are checked with FactDB for subsumption, e.g., if a Fact DB entry says about John Smith that he resides at 123 Main St. in a certain town and the new text introduces an instance of John Smith at the same address, this state of affairs licenses co-reference

- discourse analysis: Lexically, e.g. *so, finally, therefore, anyway, however*, most prepositions ranging over clauses (e.g., *After John finished breakfast he drove off to work*). Grammatically, e.g. the relative tense and aspect forms of verbs and a subordinate clause: *Having finished breakfast, John drove off to work*. "Ontologically", if e.g. two or more propositions are recognized as components in the same complex event, then, even if the overt textual clues are missing, the module will establish discourse relations among them based on the background world knowledge from the ontology or the Fact DB, in the case when the corresponding complex event was already instantiated.

- the stylistic zone of the lexicon provides arguments to the style computation function; it was present in the Mikrokosmos implementation of OntoSem but did not make it into the CAMBIO/CREST one because neither application used it; (example: the persistent use of passive voice in a text signifies a higher level of formality)

Acquisition of Static Knowledge Sources

The bootstrapping of the ontology consists of:

- developing the most general concepts;
- acquiring a rather detailed set of properties, the primitives in the representation system (for example, case roles, properties of physical objects, of events, etc.);
- acquiring representative examples of ontological concepts that provide models (templates) for specification of additional concepts;
- acquiring examples of ontological concepts that demonstrate how to use all the expressive means in ontology specification, including the use of different facets, of sets, the ways of specifying complex events, etc., also to be used as a model by the acquirers, though not at the level of an entire concept.

The bootstrapping of the lexicon for the recent implementations of ontological semantics involved creating entries exemplifying:

- all the known types of syntax-to-semantics mapping (linking);
- using every legal kind of ontological filler – from a concept to a literal to a numerical or abstract range;
- using multiple ontological concepts and non-propositional material, such as modalities or aspectual values, in the specification of a lexical entry;
- using such expressive means as sets, refsems and other special representation devices.

An information extraction system fills ontologically inspired templates that become candidate entries in the FactDB.

A new concept is decreasingly less warranted when:

- it differs from its parents in the inventory of properties,
- if the difference between a concept and its parent is in the values of relations other than the children of ONTOLOGY-SLOT (e.g., IS-A or INSTANCES)
- if there are differences between a concept and its ancestor on more than one attribute
- if the constraint on an attribute in the parent is an entire set of legal fillers or if a relation has as its filler a generic constraint "OR EVENT OBJECT," and the child introduces stricter constraints

It is legal, for constraints in a child to be looser than those in an ancestor (e.g. an ancestor may have inheritance on a property completely blocked using the special filler NOTHING, but a child could revert to a contentful filler).

Acquisition of Lexicon

- the information supporting inference resides largely in the PRE-CONDITION and EFFECT properties of EVENTs in the ontology, not in the lexicon

The steps in lexical acquisition may be presented as follows:

- **polysemy reduction**: decide how many senses for every word must be included into a lexicon entry: read the definitions of every word sense in a dictionary and try to merge as many senses as possible; discard a sense when
    - it requires further disambiguation if used in a short text example
    - it has a property with a small set of fillers; it will become a part of the meaning of a phrasal.
- **syntactic description** of every sense of the word;

- **ontological matching**: describe the semantics of every word sense by mapping it into an ontological concept, a property, a parameter value or any combination thereof;
- **adjusting lexical constraints**;
- **linking** syntactic and semantic properties of a word sense.

Hypothesis of Practical Effability for Computational Applications: Any text in the source language can be translated into the target language in an acceptable way on the basis of a lexicon for the source language and a lexicon for the target language with a comparable ratio of entries per superentry.

The parametric representation of **abhor** (as modality type evaluative value < 0.1) historically emerged in the Mikrokosmos implementation after an earlier attempt to place it in the EVENT branch of the ontology failed: there were no concepts in it that were similar to it, due to the strategic decision not to represent states as EVENTs.