A. BASIC NOTIONS WHEN DESIGNING LEXICAL TEMPLATES WITHIN THE LEXICAL CONSTRUCTIONAL MODEL

(by Socorro Bernardos (UPM), Rocío Jiménez Briones (UAM) and María Beatriz Pérez Cabello de Alba (UNED), as a result of the meetings on November 8th, 2007 and April 23rd, 2008).

1. **A lexical template (LT)** is made up of a module containing semantic information – the semantic component- and a module with syntactic information or the syntactic component. Each module appears within []. Pragmatic information, if present, appears in its own module at the beginning of the LT within <…>:

   <pragmatic features> [semantic representation] + [syntactic representation] = lexical template

2. **THE SYNTACTIC COMPONENT** is realized by means of the logical structures (LSs henceforth) proposed in Role and Reference Grammar (RRG; Van Valin & LaPolla 1997; Van Valin 2005). It captures the following information:

   a. the argument structure, that is, the number of obligatory participants the predicate requires: x, y, z;
   b. the verbal class or lexical inheritance, expressed by the appropriate semantic prime which terms the predicate can be decomposed into: **feel'**, **exist'**, **be'**, **move'**, **know', think'**, etc.;
   c. the interface with syntax, in other words, how different types of LSs map out different syntactic constructions;
   d. the argument relation with the semantic module. By default, the 1st external variable in the LS (usually *x*) is linked with the internal variable *1* in the semantic component. The 2nd external variable (usually *y*) is understood to correspond to the internal variable *2*, whereas the 3rd external variable or *z* would match the internal variable *3*. Therefore, it is not necessary to make explicit the linking between internal and external variables in LTs, unless it is an exception to the already-mentioned default linking.

   For the time being, we will keep on using the type of formalization employed in RRG with respect to the placement of external variables, the usage of brackets, square brackets, etc.

   Computationally, if a LT does not include a LS, it'll mean that the LT inherits it from its most immediate hypernym. Linguistically, however, LSs can be repeated or not within the same lexical class according to necessity.

   Following the battery of tests in Van Valin (2005) or Mairal Usón and Cortés (2006), the appropriate LS will be chosen for each verbal predicate. When working with Spanish verbs, González Vergara (2006) may be quite useful too. Below are the types of LSs proposed in RRG:

VERB CLASS	LOGICAL STRUCTURE	EXAMPLE	INSTANTIATION OF LS
State	**predicate'** (x) or (x,y)	See	**see'** (x,y)
Activity	**Do'** (x, [**predicate'** (x) or (x,y)])	Run	**do'** (x,[**run**' (x)])
Achievement	INGR **predicate'** (x) or (x,y), *or* INGR **do'** (x, [**predicate'** (x) or (x,y)]	Pop (burst into tears)	INGR **popped'** (x)
Semelfactive	SEML **predicate'** (x) or (x,y) SEML **do'** (x, [**predicate'** (x) or (x,y)]	glimpse, cough	SEML **see'** (x,y)
Accomplishment	BECOME **predicate'** (x) or (x,y), *or* BECOME **do'** (x, [**predicate'** (x) or (x,y)]	Receive	BECOME **have'** (x,y)
Active accomplishment	**do'** (x, [**predicate$_1$'** (x, (y))] & INGR **predicate$_2$**' (z,x) or (y)	Drink	**do'** (x,[**drink**' (x,y)]) & INGR **consumed'** (y)
Causative accomplishment	α CAUSES ß where α, ß are LS of any type	Kill	[**do'** (x, ∅)] CAUSE [BECOME [**dead'** (y)]

Table 1. Logical structures with RRG (adapted from Van Valin 2005)

Some further clarification:

- BECOME is now decomposed into PROC & INGR: according to Van Valin (2005: 44), the LSs below, which would represent "me enfrié" (*I became cold*), are exact equivalents. Both claim that there is a change of state from 'less' cold to 'more' cold that entails that the process has reached the endpoint of being cold:

  BECOME **cold'** (x) = PROC **cold'** (x) & INGR **cold'**(x)

  This new operator PROC is useful for those languages that lexically represent processes independent of a possible endpoint and result state – i.e. *to become colder*. Unlike BECOME, INGR and SEML, PROC does not occur with activity verbs.

  To ease the elaboration and understanding of the information in the syntactic module, BECOME will continue to be used in our representations whenever there is no need to distinguish each subprocess independently.

- All **predicates** used in the syntactic module of the LCM templates, unlike the 'canonical' RRG LSs, are semantic primes that identify the verb with its appropriate verbal class (lexical inheritance). Such primes correspond to the superordinate verbal predicates identified through extensive factorization of meaning definitions in the Functional Lexematic Model (henceforth FLM; Martín Mingorance 1998; Faber & Mairal Usón 1999) and/or with the semantic primitives put forward by the Natural Semantic Metalanguage (NSM; Wierzbicka 1996; Goddard & Wierzbicka 2002, 2005, 2007). Thus, in the examples of Table 1, **run'**, **popped'**, **drink'**, **consumed'** and **dead'**, since they are not primitive terms, must be replaced by the appropriate nuclear terms and/or semantic primes:

Lexical domain	Nuclear term
EXISTENCE	be/happen
CHANGE	become
POSSESSION	have
SPEECH	say
EMOTION	feel
ACTION	do, make
COGNITION	know, think
MOVEMENT	move (go/come)
PHYSICAL PERCEPTION	see / hear / taste / smell / touch
MANIPULATION	use

Table 2. FLM lexical domains and nuclear terms (Mairal Usón & Faber 2007: 147)

3. **THE SEMANTIC COMPONENT** captures those semantic parameters that differentiate each verb within the same lexical class. The semantic module is built up using: a) the inventory of lexical functions employed in Meaning and Text Theory (MTT; Mel'cuk 1989; Mel'cuk *et alii* 1995; Alonso Ramos 2002), b) the NSM semantic primes, and c) natural language words which, stored in an ontology of nouns and adjectives, will eventually be defined in terms of the primes already employed in LTs.

    a. As far as **lexical functions** are concerned, they are used differently in the LCM, that is to say, they are employed paradigmatically to define verbal predicates, whereas in the MTT they help to explain the lexical collocations of a language syntagmatically. Hence, within the LCM, the MTT lexical functions are considered *semantic functions*. Each function, written in capital letters, selects an argument or feature to operate on. Arguments vary among: internal variables (the subscripts 1, 2, 3), semantic primitives, other semantic functions, non-primitive words and hypernyms. Below are the functions identified so far, besides the ones included in Alonso (2002). The list is not exhaustive, since new ones may appear as new data and research require them:

Semantic Function **MTT Lexical Functions (with their application adapted to paradigmatic structure)**	**Definition**
ABLE	Ability to
ANTI	Antonym/negation
CAUS	Cause
CONT	Continuity/duration
CULM	The highest point
DEGRAD	To get worse
INCEP	The beginning of
INSTR	Instrument/means
INVOLV	Sub-activities implied by the predicate
MAGN	Intense(ly), very [intensifier], to a very high degree
MINUS	Less
OBSTR	To hinder/to work with difficulty
PLUS	More
SYMPT	Physical symptoms
Additional LCM semantic functions	**Definition**
BECAUSE	Reason
FIGUR	figuratively, metaphorically
LOC	Temporal location
LOCTF	Temporal location future
LOCTP	Temporal location past
LOCT	Temporal location present
LOCIN	Spatial location with directionality 'in'
LOCAD	Spatial location with directionality 'to'
MANIF	Showing in appearance
MANNER	Manner
MINUSCONTtime	for a short time
PLUSCONTtime	over/for a long period of time
PLUSMAGN	Extremely
PURP	Purpose
RESULT	The sub-activity is a direct, non-cancellable, result of the main predicate
=	Equal
&	And
/	Or

Table 3. MTT lexical functions & LCM semantic functions

There is no need for arguments of functions to appear between brackets, since the meaning or nature of the function will reveal its required arguments. For example, if the function is binary and contains a numerical subscript next to it, that'll be interpreted as one of its arguments, being its second argument everything that appears to the right of the function. If the function is binary but without any subscripts, its arguments will be whatever appears to its right and left. If the function is ternary and two numerical subscripts separated by a comma come along, those will be its arguments, taking as its 3rd argument the information that appears to its right. If the function is unary, whatever comes to the right of the function will be interpreted as its only argument. Besides, the function '&' is employed to separate unrelated semantic parameters that, however, occur at the same level. Below is a tentative classification of functions according to their nature or meaning:

Unary Functions	Binary Functions
ANTI	ABLE
CONT	CAUS
CULM	BECAUSE
DEGRAD	INSTR
FIGUR	INVOLV
LOC	LOCIN
MAGN	LOCAD
MINUS	MANIF
OBSTR	MANNER
PLUS	PURP
INCEP	RESULT
	SYMPT
	&
	=
	/

Table 4. Nature of the LCM semantic functions

Let's see how the underlined parts of the following definitions will be formalized within the LCM:

alarm: *to frighten somebody, making them feel anxious about something unpleasant or dangerous in the future*.

[$RESULT_2$ $SYMPT_2$ anxiety BECAUSE LOCTF somethingbad]

RESULT is a binary function because it expresses that something results in someone/something else. The subscript 2 is understood as one of its arguments – where something/someone results – and what comes afterwards as the second argument, that is, the very result: $SYMPT_2$ anxiety BECAUSE LOCTF somethingbad.

SYMPT is another binary function that establishes a relation between an entity that feels something and what is felt. This explains why the participant that develops the symptom of a physical or mental illness – subscript 2 – is understood as its first argument and what comes afterwards – the symptom itself – as its second argument: anxiety BECAUSE LOCTF somethingbad.

BECAUSE refers to the reason someone/something has for something, so it is a binary function that operates on 'anxiety' as one of its arguments and 'LOCTF somethingbad' as another one.

LOCTF is a unary function that expresses the temporal location of something or somebody in the future. Its only argument is then 'somethingbad'.

pinchar: *molestar a alguien repetidamente con la intención de que se enfade*.
('needle: annoy to-acc sb repeatedly so that s/he gets angry')

[$MANNER_1$repetido & $PURP_1$ $CAUS_2$ $SYMPT_2$enfado]

MANNER is a binary function that accounts for the way in which something/somebody carries out an event. In this example, it takes scope over

the subscript 1 – the participant that annoys someone - and over 'repetido' *repeated*, which expresses the manner in which the annoyance is done.

PURP introduces a new semantic parameter that reflects the purpose someone has in doing something. It is then a binary function that takes two arguments: 1 and 'CAUS$_2$SYMPT$_2$enfado'. Since PURP is not related to MANNER, it has been separated from it by means of '&'.

CAUS is another binary function whose 2 arguments are satisfied through the subscript 2, which signals the participant that causes something, and through 'SYMPT$_2$enfado', the new feeling of anger brought about in 2.

What follows is a sample of possible combinations of semantic functions and arguments:

(1) **Function** + **hypernym**:

annoy: *to anger somebody a little*

[MINUSanger]

(2) **Function** + **internal variable**:

pinchar: *molestar a alguien repetidamente con la intención de que se enfade*.

[MANNER$_1$…]

(3) **Function** + **function**:

pinchar: *molestar a alguien repetidamente con la intención de que se enfade*.

[PURP$_1$ CAUS…]

(4) **Function** + **semantic primitive(s)**:

alarm: *to frighten somebody, making them feel anxious about something unpleasant or dangerous in the future.*

[…LOCTF somethingbad]

(5) **Function** + **non-primitive word**:

alarm: *to frighten somebody, making them feel anxious about something unpleasant or dangerous in the future.*

[…SYMPT$_2$ anxiety…]

(6) **Function** + **internal variable & internal variable**[1]:

[1] So far, this is the way reciprocal verbs are captured in the LCM semantic module: by connecting two internal variables so that they both work as the only argument of that function. Thus, the semantic component of the LT for *enemistar* 'antagonize' represents a sentence like *Politics antagonized my friends*, where it is implied that politics caused my friends to get angry with each other, not to get angry with themselves.

Enemistar*: enfadar a alguien, haciendo que deje de ser su amigo.*
('antagonize: anger to-ACC sb, causing their friendship to break off')

[…RESULT$_{2\&3}$ ANTI SYMPT$_{2\&3}$amistad]

b. In relation to **semantic primitives**, they are language-neutral and written in small letters. This is the latest list of primes available, which differs a bit from the previously used lists:

Gramatical category	NSM Semantic Primes	Spanish exponents
Substantives	I, YOU, SOMEONE/PERSON, PEOPLE, SOMETHING/THING, BODY	YO, TÚ, ALGUIEN/PERSONA, GENTE, ALGO/COSA, CUERPO
Determiners	THIS, THE SAME, OTHER/ELSE	ESTO, LO MISMO, OTRO
quantifiers	ONE, TWO, SOME, ALL, MANY/MUCH	UNO, DOS, ALGUNOS, TODO, MUCHO
Evaluators	GOOD, BAD	BUENO, MALO
Descriptors	BIG, SMALL	GRANDE, PEQUEÑO
Augmentor, Intensifier	VERY, MORE	MUY, MÁS
Mental predicates	THINK, KNOW, WANT, FEEL, SEE, HEAR	PENSAR, SABER, QUERER, SENTIR, VER, OÍR
Speech	SAY, WORDS, TRUE	DECIR, PALABRAS, VERDAD
Actions, events, movement, contact	DO, HAPPEN, MOVE, TOUCH	HACER, PASAR, MOVERSE, TOCAR
Location, existence, possession, specification	BE (SOMEWHERE), THERE IS/EXIST, HAVE, BE (SOMEONE/SOMETHING)	ESTAR, HAY, TENER, SER
Life and death	LIVE, DIE	VIVIR, MORIR
Time	WHEN/TIME, NOW, BEFORE, AFTER, A LONG TIME, A SHORT TIME, FOR SOME TIME, MOMENT	CUÁNDO/TIEMPO, AHORA, ANTES, DESPUÉS, MUCHO TIEMPO, POCO TIEMPO, POR UN TIEMPO, MOMENTO
Space	WHERE/PLACE, HERE, ABOVE, BELOW; FAR, NEAR; SIDE, INSIDE	DÓNDE/SITIO, AQUÍ, ARRIBA, DEBAJO, CERCA, LEJOS, LADO, DENTRO
"Logical" concepts	NOT, MAYBE, CAN, BECAUSE, IF	NO, TAL VEZ, PODER, PORQUE, SI
Relational substantives	KIND, PART	TIPO, PARTE
Similarity	LIKE	COMO

Table 5. NSM semantic primitives (Goddard & Wierzbicka 2005, 2007)

In a plenary lecture at the *2005 4th International Contrastive Linguistics Conference*, Goddard provided the following table that could be helpful to 'translate' some of our dictionary definitions:

Semantic primes	Morphosyntactic construction types
ONE, TWO, SOME, MUCH/MANY	Number-marking (incl. duals)
THE SAME, OTHER	Switch-reference, obviation, reflexives, reciprocals
WANT	Imperatives, purposives, "uncontrolled" marking
KNOW, SEE, HEAR, SAY	Evidential systems
WORDS	Delocutive verbs
DO, HAPPEN	Active marking, passive voice, inchoatives
FEEL, THINK	Expressive derivation, experiencer constructions
GOOD, BAD	Benefactives, adversatives
BIG, SMALL	Diminutives, augmentatives
VERY	Superlatives, expressives
NOW, BEFORE, AFTER, A LONG TIME, A SHORT TIME	Tense systems (incl. degrees of remoteness)
FOR SOME TIME, IN ONE MOMENT	Aspect (punctual, durative)
ABOVE, BELOW, ON (ONE) SIDE, NEAR, FAR	Elaborate locational deixis
PART OF	Inalienable possession

Table 6. Some morphosyntactic construction types and associated semantic primes (Goddard & Wierzbicka, 2005)

c. About **selection restrictions**, the operator equal (=) is used for their formalization. It is a binary function that takes as its first argument the internal variable to which the restriction applies and the restriction itself as its second argument. Selection restrictions appear at the end of the semantic module, preceded by '&':

befall: (Fml)(Lit) *to happen (esp. of unpleasant events).*

<fml,lit> [happen & 1=unpleasant events]

d. **Lexical inheritance** is captured by means of the inclusion of the most immediate hypernym at the beginning of the semantic module, followed by '&':

alarm: *to frighten somebody, making them feel anxious about something unpleasant or dangerous in the future.*

[frighten & $RESULT_2$ $SYMPT_2$ anxiety BECAUSE LOCTF somethingbad] [[(**do'** (x, ∅)] CAUSE [**feel'** (y, [**fear'**])]]

4. **THE PRAGMATIC MODULE** encodes the pragmatic features and the register parameters that contribute towards the differentiation of the verbs within the same class. This information is written in small letters and placed before the semantic module within angled brackets. If two or more features co-occur, they are separated by a comma: <fml,old>. Following Martín Mingorance (1998: 96-97), the pragmatic module may include features related to these three functions:

a) the informative function: it is the type of discourse a word appears in (*legal*, *religious*, etc.);

b) the psychological function: the connotation of a word (*ironic*, *pejorative*, etc.);

c) the social function: it is related to differences concerning socio-cultural features like the formality of the word, its geographical use, etc.

These are the features identified so far:

formal	<fml>
informal	<infml>
colloquial	<col>
old-fashioned	<old>
literary	<lit>
emphatic	<emph>
philosophical	<ph>

Table 7. LCM pragmatic features

B. HOW TO READ AND INTERPRET A LCM LEXICAL TEMPLATE:

1. First, the **syntactic module** is approached:

Enrabietar/enrabiar: *enfadar mucho a alguien por motivo leve y durante poco tiempo.*
('nettle: anger a lot to-ACC sb for no reason and for a short time')

[(**do'** (x, ∅)] CAUSE [BECOME **sentir'** (y, [**enfado'**]] = "*x* does something that causes *y* to become angry"

2. Second, the **semantic module** is dealt with, reading it from left to right:

[MAGNenfadar & $BECAUSE_2$motivoleve & MINUSCONTtiempo] = "*enrabietar/enrabiar* – nettle- is to anger somebody a lot (MAGN*enfadar*) because s/he has no reason, and only for a short time"

3. Therefore, the full LT for *enrabietar/enrabiar* would be understood as follows:

[MAGNenfadar & $BECAUSE_2$motivoleve & MINUSCONTtiempo] [[(**do'** (x, ∅)] CAUSE [BECOME **sentir'** (y, [**enfado'**]]]= "in *enrabietar/enrabiar* – nettle - *x* does something that causes *y* to get angry, and *y* gets angry a lot, for no reason, and for a short time"

4. Finally, the **pragmatic information**, if present, is taken into account:

be: (literary) (philosophical) *to exist in space or time.*

<lit,ph> [LOCIN/LOCTexist] [**exist'**(x)] = "in *be x* exists in space or time as employed in the philosphical and literary discourse"

REFERENCES

Alonso Ramos, Margarita (2002). "Colocaciones y contorno en la definición lexicográfica" *Lingüística española actual* 24(1): 63-96.

Faber, Pamela y Ricardo Mairal Usón (1999). *Constructing a Lexicon of English Verbs*. Berlin/New York: Mouton de Gruyter.

Goddard, Cliff y Anna Wierzbicka (2002). *Meaning and Universal Grammar: Theory and Empirical Findings*. Amsterdam/Filadelfia: John Benjamins.

--- (2005). "Universal human concepts as a basis for Contrastive Linguistics". Conferencia plenaria leída en *The Fourth International Contrastive Linguistics Conference*. Universidad de Santiago de Compostela, septiembre 2005.
--- (2007). "NSM analyses of the semantics of physical qualities", *Studies in Language* 31 (4): 761-800.

González Vergara, Carlos (2006). "La Gramática del Papel y la Referencia: una aproximación al modelo". *Onomázein* 14 (2006/2): 101-140.

Mairal Usón, Ricardo y Francisco José Cortés Rodríguez (2006). "An overview of Role and Reference Grammar". En Ricardo Mairal Usón, Mª Ángeles Escobar Álvarez, Mª Sandra Peña Cervel y Eva Samaniego, eds., *Current Trends in Linguistic Theory*. Madrid: Universidad Nacional de Educación a Distancia, pp. 97-175.

Martín Mingorance, Leocadio (1998). *El Modelo Lexemático-Funcional. El legado lingüístico de Leocadio Martín Mingorance* [ed. A. Marín Rubiales]. Granada: Universidad de Granada.

Mel'cuk, Igor (1989). "Semantic primitives from the viewpoint of the Meaning-Text Linguistic Theory", *Quaderni di semantica* 10 (1): 65–102.

Mel'cuk, Igor, A. Clas y A. Polguère, A. (1995). *Introduction à la lexicologie explicative et combinatoire*. Louvain-la-Neuve (Belgium): Duculot/ Aupelf - UREF.

Van Valin, Robert D., Jr. (2005). *Exploring the Syntax-Semantics Interface*. Cambridge: Cambridge University Press.

Van Valin, Robert D., Jr. y Randy LaPolla (1997). *Syntax: Structure, Meaning and Function*. Cambridge: Cambridge University Press.

Wierzbicka, Anna (1996). *Semantics: Primes and Universals.* Oxford: Oxford University Press.