Semantic Modelling and Lexical Knowledge

Antonio Moreno Ortiz

0. Abstract

In the present paper I describe the conceptual level of representation of a database, currently in development, which aims at fulfilling the Functional-Lexematic model. I will concentrate on the semantic component, as it is central for the theory of Functional Grammar as well as for the database overall structure. A brief introduction to database architecture and, more specifically, semantic modelling, is provided. The focus is on semantic modelling as a useful tool to represent lexical information.

1. Introduction

Semantic modelling is commonplace in database architecture to represent the conceptual level of any information system. We will use this methodology to present the semantic component in a lexical database, currently in development, devised to suit the lexicographic model postulated by Martín Mingorance (1984, 1987, 1990) which integrates Simon Dik's Functional Grammar (Dik, 1978, 1989) and Coseriu's theory of Lexematics (Coseriu, 1981). This model is not the centre of discussion in this paper. Its functionality and validity are, as far as this paper is concerned, taken for granted. It is the implementation of it -and by extension of any other hypothetical NLP system- at the conceptual level that makes the core of the paper.

Some remarks on the terminology used here seem to be at place. Semantic modelling and linguistics share several terms, which may lead to confusion. Different uses will be signalled by changes in the typeface of the expressions.

Semantics: in linguistic discourse this term refers to a level of description which focuses o the meaning of expressions which occur in human communication. A semantic description o a lexical item looks like this: "if you <u>acquire</u> something you obtain it" or "a <u>chair</u> is a piece o furniture used to sit on."

SEMANTIC: in the discourse of information theory and information management, this term refers to the representational properties of bits of data which lead to constraints i applicability and manipulation of these bits of data.

SEMANTIC **data modelling**: in a wide sense, semantic data modelling means mapping o some facts of a given world/enterprise/slice of reality to a description in some forma language, thereby abstracting away features of reality that do not lend themselves t normalization under the descriptive language chosen or features which for one reason c another are not relevant or distinctive. In a more specific sense, data modelling means th interpretation of a formalism in a strict mathematical sense. In what follows, we will use the expression "semantic data modelling" in the first sense.

To represent: in linguistic discourse, an expression, i.e. a chunk of graphemes or morphemes, represents, or stands for a certain object, idea or mental image[1].

TO REPRESENT: in the context of information management, an expression, i.e. a chunk of bits, represents a bit of information which depends on its interpretation, or on the model supporting the data structure this chunk of bits is part of.

Conceptual level: in linguistic discourse, the term "conceptual level" refers to the cognitive structures with which a linguistic sign is said to be related by parts of its meaning. On the conceptual level, lexical signs of one or more natural languages are linked by their property of being instantiations of some concepts they share[2].

CONCEPTUAL level: in the discourse of information management, the "CONCEPTUAL level" designates a level of description of data which abstracts away from its machine internal encoding as well as from then end-user's view of the system. A data type called "text" on the conceptual level might be instantiated as a chunk of signed characters or as a chunk of unsigned characters.

Conceptual schema: in Cognitive Linguistics conceptual schemata are used as an abstract means of representing human cognitive structures around which linguistic meaning is said to be organized.

CONCEPTUAL SCHEMA: in the discourse of information management this term refers to a certain kind of diagram which serves as a tool for developing information representation and retrieval systems. This term is of great importance in the context of this paper as the description of one kind of conceptual schemata is one of the central issues of it.

At this point, it seems appropriate to put forward some basic facts about database architecture which may help to clarify some of these notions. Basically three distinct levels are distinguished in database architecture[3], the **external level** (the data as seen by the end-user), the **internal level** (the data as seen by the system), and the **CONCEPTUAL LEVEL**. It is this conceptual level that we are dealing with here. This level, embodied by the CONCEPTUAL SCHEMA is the keystone of the whole architecture (Gardarin and Valduriez, 1989).

The conceptual level standing in between the physical and the external level, serves as a reference point for the system implementor. It contains a description of the entities, relationships, and properties relevant to the database and constitutes a stable platform to map the different external schemata, which describe the data as seen by the programmers onto the internal -physical- one, which describes the data as seen by the system.

---

[1] Depending on the different philosophical positions which need not be distinguished in our context.

[2] Again, this is just a rather sketchy description, leaving aside the considerable differences in the use of this term in the different mentalistic branches of linguistics and schools of thought. This characterization, however, suffices to contrast the use of this term in linguistics from its use in information management.

[3] This classification corresponds with the ANSI/X3/SPARC general architecture of database management systems.

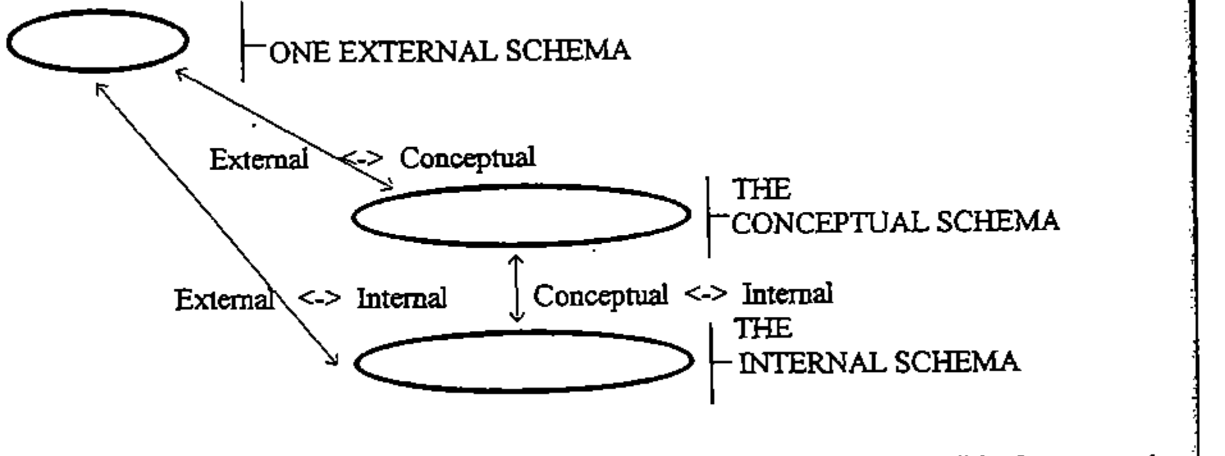

Fig. 1: ANSI/SPARC Architecture and possible data mapping

2. Data Models

Semantic modelling originated in the 1970's as a necessity of providing information systems with some more "understanding" of the data they contain. In database terms, usually, the DBMS's knowledge of the data it handles is very poor. Normally, it is restricted to the type of data (text, numeric, Boolean, etc.), and it would be appropriate if this limited "awareness" could be expanded to get to know what these data really "mean," or, at least, to increase the comprehension of the entities and relationships that the data designate in the real world.

Before getting into semantic modelling proper, we should first offer an overview of data models in general.

Data modelling, with respect to database design, can be formally defined as follows

> "given the information and processing requirements of a data intensive application (e.g. an information system), construct a representation of the application that captures the static and dynamic properties needed to support the desired processes (i.e. transitions and queries). The representation must be able to meet ever-changing requirements." (Brodie 84: 39).

Data models are thus central to information systems, they provide a formal basis for tools and techniques used in the development of the information system in question (Brodie 84), in our case the lexical database.

There have been several types of data models. A typical classification of data models is the following:

1. **Primitive data models**. These early model were file-oriented; objects are represented as records that are grouped in files.

2. **Classical data models**. Improved elaborations of these primitive models are the hierarchic data model, which is a direct extension of the file model, the network data model, which is a superset of the hierarchic model, and the relational model. Finally, the entity set model, based on set theory falls into this category.

The relational data model meant a significant departure from the former two models. Thus it deserves closer examination. First and foremost, the relational data model is an application of sound mathematical formalism, relational calculus and/or relational algebra[4]. Second, the corresponding query language is no longer procedural but declarative. In short, this means that the end user of an information system based on the relational data model just specifies what bits of data he wants to get, and not how to access them. Constraints on the manipulation of data which are due to the SEMANTICS of these data cannot be easily, if at all, handled by a relational query language, such as SQL (Standard Query Language). This deficiency is attacked by a series of different approaches to SEMANTIC data modelling.

3. **Semantic data models**. The orientation towards the conceptual level achieves top importance in these data models. They were developed, as we have already mentioned, in order to provide data modelling with a powerful tool to analyze the world/enterprise/slice of reality which conforms a given information system in a more natural way. Brodie (Brodie 84: 28) distinguishes five types of semantic data models, viz.:

i. Direct extensions of classical models:
- The structural model
- The Object-Role model
- The E/R model

ii. Mathematical models.
- Based on set theory
- Based on first order logic.
- Based on the universal instance assumption.

iii. Irreducible data models

iv. Static Semantic Hierarchy models.

v. Dynamic Semantic Hierarchy models.

4. **Special purpose data models**. The latest trend in data modelling implies a specialization of the model depending of the punctual necessities of the application in question.

3. The E/R Model

Chen's (Chen 1976) entity-relationship (E/R) model was first presented by its creator in 1976 and successively updated and refined afterwards (Chen 1983). Although initially it was meant to substitute former data models, more explicitly the relational model, as a matter of fact, the initiatives to make extensive use of the model as an actual engineering tool have not caught on as far as commercial database management systems are concerned[5]. Nevertheless, since its appearance, it has attracted the interest of database designers and software engineers in general, who have used the model as the basis to develop their particular applications. Thus, for example, we can equate E/R components to relational counterparts, as we will show in our exposition. First we will offer a general view of the E/R model.

[4] Cf. Codd (1990) and Date (1988) for the formal details.

[5] See in this respect Stonebraker (1988).

In this approach, the world/enterprise/slice of reality is depicted by means of a number of semantic concepts in the following way: the world is made up of *entities*, an entity is an distinguishable object, these entities possess a number of *properties*, which are pieces of information which describe the entities in some way. Those entities have an identity, i.e., they are uniquely identifiable, and they maintain certain *relationships* with each other. There are also *subtypes* of entities: the entity Y is a subtype of the entity X if and only if every Y is also an X[6] This semantic concepts (entity, property, relationship, and subtype) are the basic concepts of the E/R model. They are used as the basic building bricks in the construction of the conceptual schema. Applications are represented as networks in which objects (entities) are nodes and relationships are edges.

The consideration of these entities is thus a key question. In our case, to give an example, choice has to be made between regarding a semantic field[7] either as a group of objects with number of characteristics, that is, an entity with some properties, or, alternatively, we can think of lexemes which hold certain -paradigmatic- relationships among one another, thus regarding the paradigmatic axis as set of relationships[8].

Basically, the world - in theory any possible world - we are trying to depict, that is, the lexicon, under the Functional-Lexematic viewpoint, fits well in this model: there are lexical items which have certain properties and which maintain certain relationships with each other - no matter what we consider as entities, relationships or properties, this will just reflect our conception of the real-world counterparts.

The E/R model provides a very useful tool for representing our entities, relationships and properties at this conceptual level in a graphical manner: E/R diagrams. By means of E/R diagrams the abstract conceptual schema is presented graphically, so that it can be mapped on to actual implementation[9]. E/R diagrams are similar to flow charts in that they use rectangles, diamonds, and ovals, but their meanings are different. Rectangles depict entities, diamonds depict relationships and ovals depict properties (see Fig. 2). Also, a major difference is that, whereas flowcharts have a beginning and an end, E/R diagrams do not; this is obvious since flow charts reflect processes and E/R diagrams reflect states.

---

[6] This description of the model is not meant to be comprehensive, rather, it should be seen as a very informal introduction to the subject. For a thorough description of the E/R model see Chen (1983).

[7] We use the term "semantic field" in the sense of lexematics (cf. Coseriu 1981 and Geckeler 1986). The main paradigmatic operator for us is that of *hyponymy*.

[8] These linguistic concepts will be clarified later on.

[9] No description of the internal or external levels of our system is offered in the present paper.

There are other conventions which stand for different types of entities, relationships, and properties. We mention here the ones that we will be using in our charts[10]. A rectangle has a double line when the entity represented is *weak*, i.e., it's existence depends on the existence of another entity. The diamond is also double to signal the relationship between a weak entity and the entity on which it depends. The marks "1" and "M" are used to signal the kind of relationships (one-to-one, one-to-many, many-to-many). Sometimes the sign "∞" is used instead of "M." "Key" properties[11] are underlined. The key property is the one that provides unique identification for a given entity.

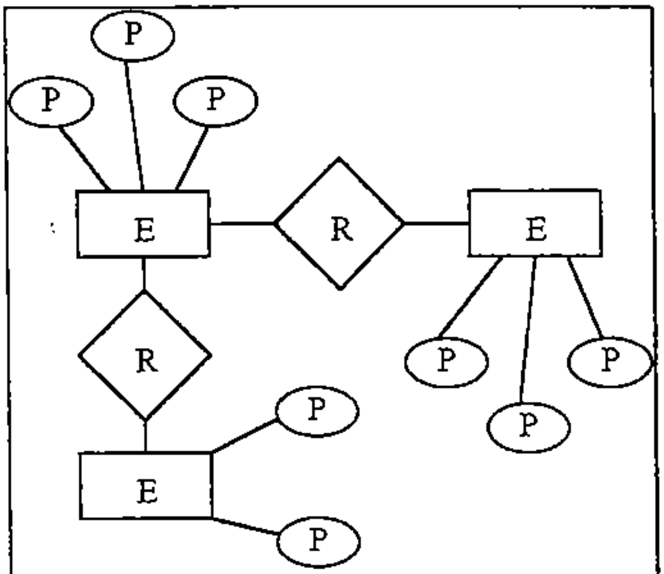

Fig. 2. E/R Diagram

The equation between E/R and relational terms is sketched next. We also include the -commonly used- informal equivalents:

E/R Term	Relational Term	"Informal" Term
	Relation	Table
Entity	Tuple	Record
Property	Attribute	Field
"Key property"	Primary Key	Unique Identifier
Relationship	Relationship	
Type - Subtype	Self-Joint	

Before we present our E/R diagrams, we need to know exactly which information and which degree of granularity of this information is required for a successful implementation.

4. Semantic information.

The level of structure within NLP usually reflects the various components typically postulated in linguistics, that is, phonological, morphological, syntactic, semantic and pragmatic (Obermeier, 1989). The system we present here is no exception, as far as it sticks to the FG postulates presented by Dik (1989).

[10] For a thorough description of E/R diagrams see Chen (1988); Date (1990).

[11] We will use the informal term "key property" as the conceptual counterpart of primary key attributes in relational terms.

The building bricks our model use are lexemes, so these will be main kind of entity that w will take as a starting point. Nevertheless, there is no widely agreed upon definition of this tem Two readings can be clearly distinguished. According to the first, a lexeme is a lexical sign (i.e. a entity at the level of the language system) by which one expression - represented phonologically by means of graphemes - is related to one meaning. According to the second, a lexeme is a lexic sign by which an expression is related to all its meanings[12]. We shall stick to the first of the readings, as far as our model considers different entities for different meanings of the same sig These entities, though related to each other are regarded by the system as differentiating, as th posses different primary keys. Our system contemplates phonological, morphological, syntacti semantic and pragmatic properties. Each of these are fairly complex, especially the semantic on which we are going to deal with here. However, it is impossible, to isolate meaning from oth components, especially syntax, as they are intimately interwoven (Dik 1989, Cruse 1986). Thus, our conceptual schemata we will offer open doors in the form of pointers to the schemata of the re of the components wherever the links are relevant. Moreover, following FG postulates, we w pursue an integrating Syn-Sem component.

Lexemes (verbal, nominal, and adjectival) are treated under FG as *predicates*, in much t same way as it is done in predicate logic. Predicates have two main types of properties: syntagma properties and paradigmatic properties. The former account for the lexeme's combinatory properti with other neighbouring lexemes within the sentence, that is its valency. The latter deal with t reasons why a given lexeme and not another -similar in meaning- is chosen at a given point wit the sentence. This properties conform the intricate set of semantic fields in a language. These t kinds of properties make up two axis which determine clauses and sentences -the syntagmatic a paradigmatic axes[13].

Concerning the syntagmatic axis, our model should account both for its quantitative a qualitative valency. That is, the database should know how many *terms* a given predicate is liable take and which kind of terms these are liable to be. In FG, terms which are required by t semantics of the predicate are *arguments*, and terms which provide further information a *satellites*.

We hope to show how the conceptual schemata serve to clarify all this. In figure 3 we sh the basic schema of a verbal predicate. The underlined properties are the key properties ("prim key" in relational terms): the Lemma -the base form of the verb- and the Complementation Patte it is a compound primary key. Each predicate is uniquely marked by these two properties. All properties ending in "ID" are identification codes which provide different kinds of informati

[12] The pros and cons of both definitions cannot be discussed here. The fact is that there is often no clear distinction of these two readings (see for example Lehrer and Kittay (1992)). In my opinion, though, one has to commit oneself to one these readings.

[13] Much leaves to be said about these two axes and especially about the relationship between them. For a thorough discussion of this, see Martin Mingorance (1990) and Mairal Usón (1993).

"Phon ID," for instance, is an arbitrary number which points at the phonetic information available for that predicate, and so on.

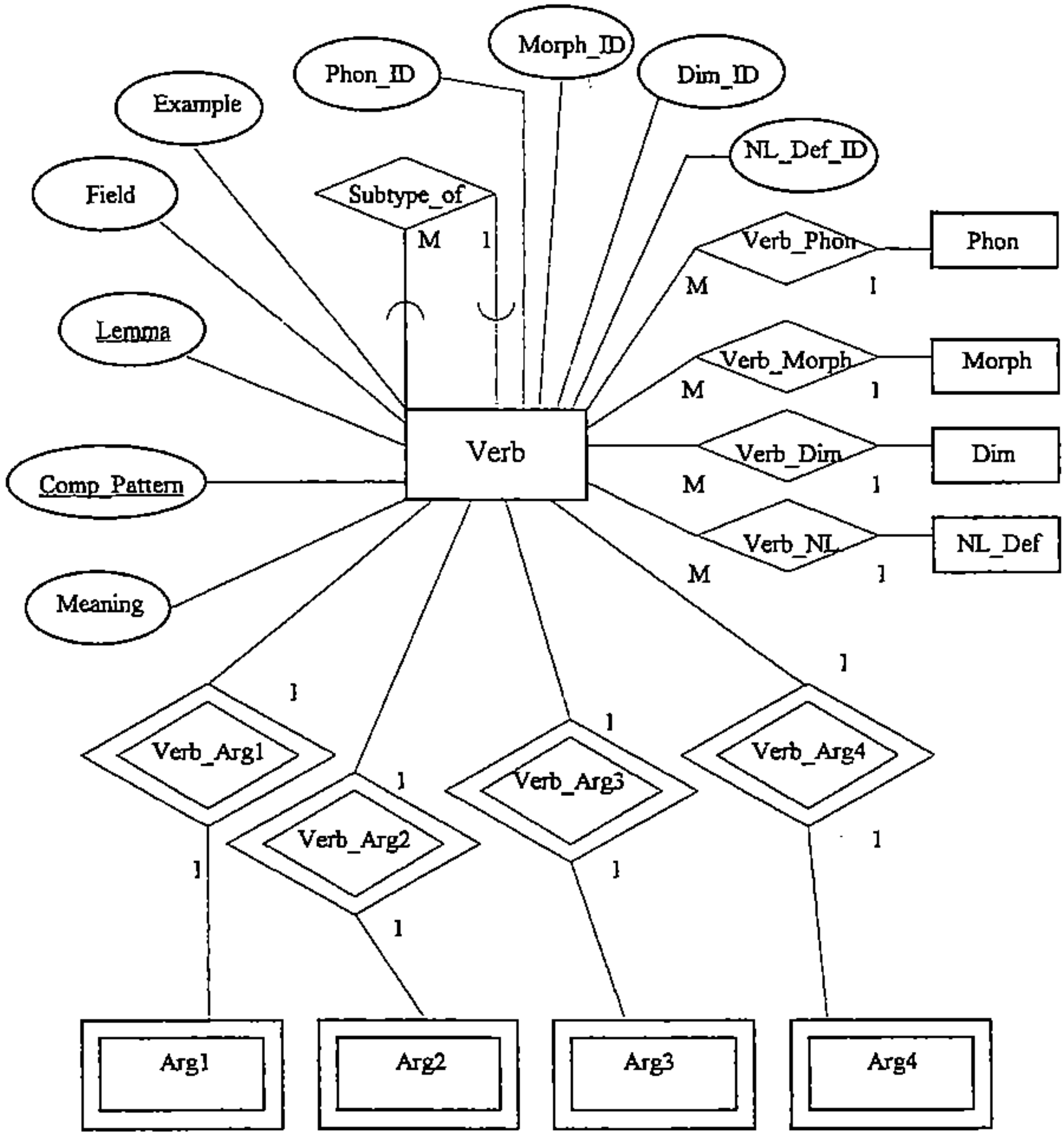

Fig. 3: Basic Verbal Predicate Schema

It should be noted that this is just a partial view of the whole schema. Thus, the relationship Verb_Phon points at another "entity," characterized as the set of phonetic properties relevant to that predicate.

In Figure 4 we show how we have modelled all the syntagmatic features of verbal predicates. To begin with, the same diagram is applicable to all arguments; this common layout proves a convenient feature since we do not have to define each argument for each predicate. Instead we assign a code (Arg_ID) and establish a relationship with yet another entity, "Arg_Desc," which is the set of properties applicable to a given type of argument.

An argument is defined in terms of five properties: selection restrictions (SR1 and SR2), semantic functions (SF1 and SF2) and a state of affairs (SoA). Selection restrictions inform us about the semantic type of the terms which can be inserted into the argument position if one is to arrive at a non-metaphorical type of predication. They are polarity features such as ±*human*, ±*animate*, ±*concrete*, ±*artifact*, etc. Semantic functions tell us about the role played by the argument in the state of affairs indicated by the predicate; examples of semantic functions are: *goal*, *message*, *recipient*, *force*, *agent*, *affected*, etc. State of affairs designate embedded propositions typically realized as *that-* clauses. Several types are distinguished cutting across two scales: the Scale of Knowledge and Behaviour and the Scale of Evaluation (Mairal Usón 1993). Examples of Stales of Affairs are *subjective probable fact*, *direct speech act*, *objective certain event*, etc.

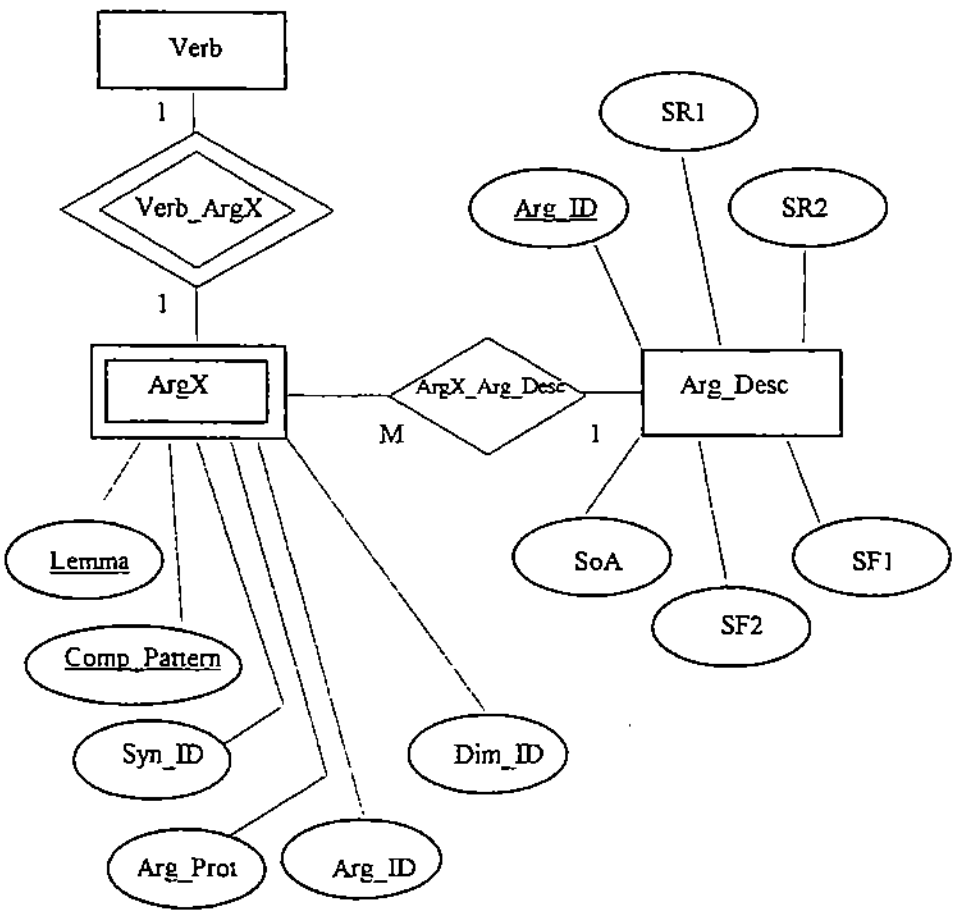

Fig. 4: Modelling the Syntagmatic Axis

All these features are pointed at by a single code assigned to the argument description as a property (Arg_ID). Again we see how this modelling is appropriate since we only access this information when it is required, thus giving the system great flexibility. Complex tasks such as pattern-matching are made simpler with this disposition. The system also gains in compactness. We are exploiting one of the most obvious advantages of the relational model: the use of functional dependencies to decompose the universal relation into smaller relations from which the original information can all be recovered (Hodgson 1991).

The property "Arg_Prot" accounts for the status of prototypicality, and it is used by the semantic parser. The property "Dim_ID," standing for Dimension, takes us to put forward how we have modelled the Paradigmatic Axis.

Our system makes extensive use the notion of type and subtype in the relational model. In figure 5 we show the E/R diagram of this.

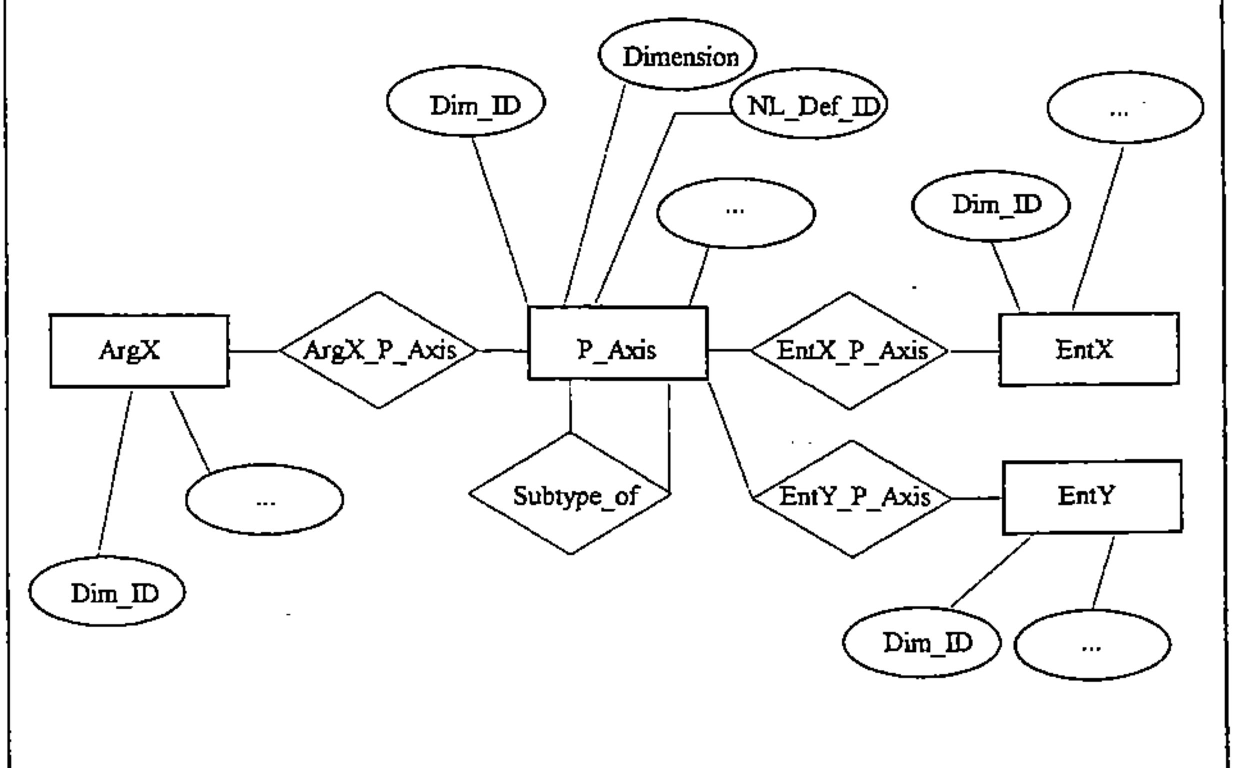

Fig. 5: Modelling the Paradigmatic Axis

Whenever we need characterization of an entity, e.g. an argument, regarding its paradigmatic status, we provide the entity with a feature ("Dim_ID") which is a pointer (foreign key) at the dimension to which it belongs in the hierarchy. A relationship is then established between this entity and the Paradigmatic Axis definition. In this, dimensions are regarded as entities which are described in terms of a number of features. One of these features is a pointer at the dimension immediately above, which in turn contains a pointer to its superordinate dimension, and then recursively until the top is found, that is, we reach the archilexeme of the field. The information attached to each of these dimensions is then at hand.

5. Sample Implementation

It would be very illustrative to follow step by step the "process" in which a given lexical item is defined and represented in the modelling we have put forward. We will do this with a limited number of verbal lexemes in the hope that it will serve to clarify our conception of lexical knowledge representation.

We will take as an example the verbal lexeme BELLOW. The different complementation patterns and meanings we wish to represent in the database are shown below in the kind of lexicographic description we have used previous to entering the data[14]:

bellow$_1$ to shout in a loud deep voice.

SV Adjunct (in/with+O)
(a) S= prototyp. human (Ag)
(b) O= -concrete ∈ emotions, feelings <physical state in which the subject produces the sound and which usually causes it > (Manner)
e.g. *Presently he began to cry and wail, then to scream and bellow with grief.*

SV Adverbial of manner
(a) S= prototyp. human (Ag)
(b) A= <adverbial description of the way the sound is made/ perceived> (manner)
<prototyp. a loud sound, reinforcing the one implicit in the definition>
e.g. *The headmaster bellowed cheerily.*

SVO$_1$ (PO (at+O$_2$)
(a) S= prototyp. human (Ag/Sp.)
(b) O$_1$= -concrete ∈ orders, warnings, commands.(Go/Mess)
(c) O$_2$= prototyp. human (Rec)
e.g. *He used to bellow orders at us from the yard.*

SVO-*Direct quote*
(a) S= prototyp. human (Ag/Sp)
(b) O= Direct Speech Act (Go/Mess)
e.g. *"Go on," the father bellowed, "get on with it before I come and give you a clout across the ear-hole."*

SVO-*That-Clause*
(a) S= prototyp. human (Ag)
(b) O= 'Subjective Probable Fact' (Go/Mess)
e.g. *The more I hear Americans bellowing that Japan is unfair, the more I would like them to calm down and think.*

bellow$_2$ to make a loud, deep sound. Esp. a large animal, usu. a bull or an elephant.

SV(A)
(a) S= animals (Ag)(prototyp. a bull or an elephant)

[14] I would like to express my gratitude to Dr. Pamela Faber and Ms. Chantal Pérez for providing me with the lexicographic information necessary for this work.

(b) A= <Adverbial description of the way the S produces the sound (manner) (prototyp angrily, fiercely, threateningly>
e.g. *The cattle lowed and __bellowed__.*

All this forms a complex data structure which has to be given some order, consistency and homogeneity.

A typical "core" tuple for a verbal lexeme looks like this:

Lemma	Phon_ID	Morph_ID	Dim_ID	Meaning_ID	Field	Comp_Pattern	Sub_Of	NL_Def_ID	Example
bellow	83	1	16	1	7	1		11	Presently...
bellow	83	1	16	1	7	2		11	The head...
bellow	83	1	16	1	7	3		11	He used...
bellow	83	1	16	1	7	4		11	"Go on"...
bellow	83	1	16	1	7	5		11	The more...
bellow	83	1	94	2	7	6		73	The cattle...

Fig. 6: Verb tuples for BELLOW

The verb BELLOW is considered to have two meanings, the first of which can be realized by five different complementation patterns, whereas the second has only one realization. We have already commented on some of the features present here when discussing the conceptual modelling. We will concentrate, then, on the semantic information. We will take first the information corresponding to the syntagmatic axis. Then we need to present the relations Arg1, Arg2, Arg3, Arg4 (see Fig. 3). The tuples that contain the information about these lexemes are shown in figures. 7 to 10.

Lemma	Comp_Pattern	Syn_ID	Arg_Prot?	Arg_ID	Dim_ID
bellow	1	1	Yes	3	
bellow	2	1	Yes	3	
bellow	3	1	Yes	3	
bellow	4	1	Yes	10	
bellow	5	1	Yes	10	
bellow	6	1	No	23	

Fig. 7: Arg1 tuples for BELLOW

As shown in Fig. 3 above, the kind of relationship between the relation "Verb" and each of these is one-to-one. This kind of relationship is not very usual; presently we will give our reasons to use such kind of relationship. A direct consequence of this, however, is that there will be as many tuples in the Arg1 relation as there are in the Verbs relation, since Arg1 stands for the only obligatory argument: the grammatical subject.

Lemma	Comp_Pattern	Syn_ID	Prep1	Prep2	Arg_Prot?	Arg_ID	Dim_ID
bellow	3	14			No	11	
bellow	4	11			No	14	
bellow	5	12			No	15	

Fig. 8: Arg2 tuples for BELLOW

Lemma	Comp_Pattern	Syn_ID	Prep1	Prep2	Arg_Prot?	Arg_ID	Dim_ID
bellow	3	8	at		Yes	8	

Fig. 9: Arg3 tuples for BELLOW

In relations Arg2 and Arg3 some complementation patterns are not found. This already gives us some information: the quantitative valency; complementation patterns 1, 2, and 6 are 3-place predicates 4, and 5 are 2-place predicates, and complementation pattern 3 is a 3-place predicate. The foreign key Syn_ID in each of these relations tells us which kind of syntactic category the argument belongs to: 1 stands for Subject, 14: Direct Object, 11: Direct Object-Direct Quote, 12: Direct Object-That-Clause, 8: Prepositional Object. All this is defined in the relation Syn[15].

Lemma	Comp_Pattern	Syn_ID	Prep1	Prep2	Arg_Prot?	Arg_ID	Dim_ID
bellow	1	5	with	in	No	5	
bellow	2	9			No	5	
bellow	6	5			No	5	

Fig. 10: Arg4 tuples for BELLOW

The Arg4 relation is somewhat special, since arguments included in this relation are considered as satellites by FG. Hpwever, as far as its representation is concerned, we treat satellites in the same fashion as arguments. We can easily deduce whether a given argument is obligatory or not from the syntactic information (5: Adjunct; 9: Adverbial of Manner). There is no need then to specify a different status for the relation as a whole.

The attributes Prep1 and Prep2 in the relations Arg2, Arg3, and Arg4 specify which prepositions, if any, the argument in question is introduced by. The attribute Arg_Prot?, on the other hand, is a Boolean data type which contains information about the prototypicality of the argument.

The -graphically- last attribute is again a foreign key which provides all the information concerning the qualitative valency. This attribute, shared by all four arguments points at the same relation: Args_Desc. The tuples pointed at by our example are the following:

[15] This relation is also structured in terms of types and subtypes, thus, for example, categories 11 and 12 (Od-Direct Quote and Od-That-Clause) are both subtypes of 14 (Od).

Arg_ID	SR1	SR2	SoA	SF1	SF2
3	+Human			Agent	
10	+Human			Agent	Speaker
23	-Concrete			Goal	Area
11	-Concrete			Goal	Message
14			Direct Speech Act	Goal	Message
15			Subjective Probable Fact	Goal	Message
8	+Human			Recipient	

Fig 11: Args_Desc tuples for BELLOW

This explains why we have used a one-to-one relationship for the definition of arguments instead of the more comfortable solution of appending the information to the relation Verbs. We would have had to define the same features for each and every lexeme, and this involves a lot of undesired redundancy.

Finally, the representation of the paradigmatic axis "starts" with a pointer (foreign key) in each of the linguistic entities we have mentioned: the verbal predicate and the arguments this takes. Thus, complementation patterns 1 to 5 all belong to the same dimension 16, whereas complementation pattern 6 (meaning 2), obviously belongs to a different dimension, 94. This foreign key, thus points at the definition of dimensions in the P_Axis relation:

Dim_ID	Lexeme	Subtype_of	NL_Def_ID
16	bellow	8	11
94	bellow	87	73

Fig. 12: P_Axis Tuples for BELLOW

Again we avoid redundancy by accessing the NL description of the dimension in the same way we go about lexemes. On the other hand, the attribute Subtype_of is, as we have already mentioned, the one we use for relating a given dimension to its superordinate dimension in the hierarchy, by means of a self-joint. Thus, the codes 8 and 87 point at the following tuples in the same relation:

Dim_ID	Lexeme	Subtype_of	NL_Def_ID
8	shout	7	3
87	Dim_21	86	230

Fig. 13: P_Axis tuples for BELLOW (1 level above)

This same technique is then applied recursively in a bottom-up fashion until the archilexeme for the field is found, which renders the hierarchies shown in figures 14 and 15.

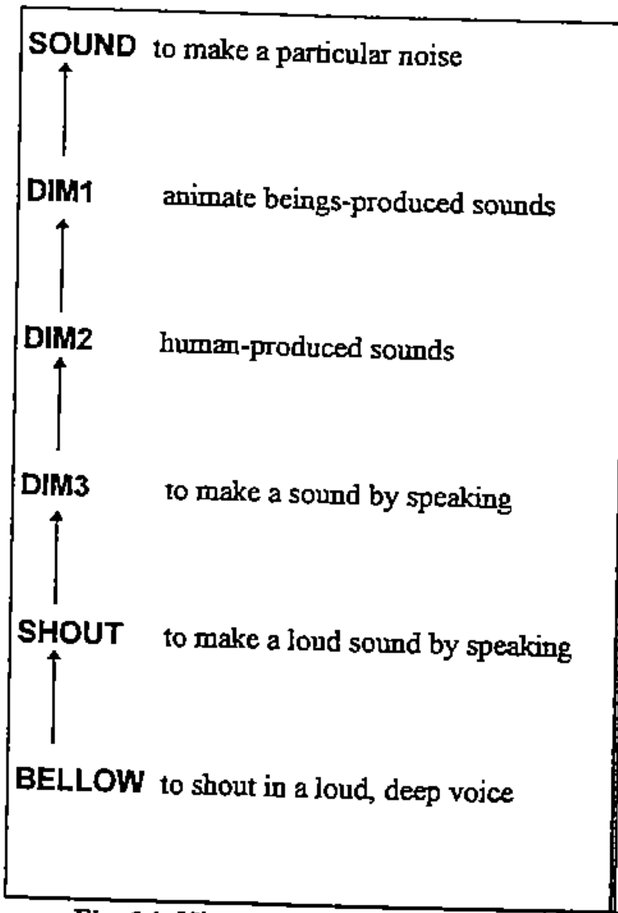

Fig. 14: Hierarchy for BELLOW (meaning 1)

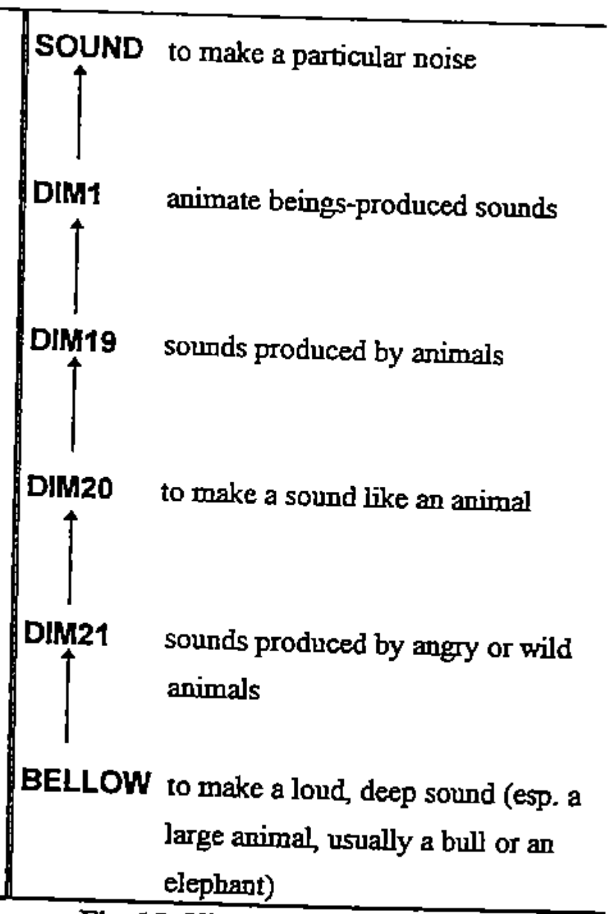

Fig. 15: Hierarchy for BELLOW (meaning 2)

6. Conclusion.

Much leaves to be said about several of the features we have touched upon, especially concerning a more detailed description of dimensions as well as lexemes themselves in the paradigmatic axis. We would also have liked to comment on the subject of *missing information*, but this is in itself a major topic of debate in the database field.

We hope, nevertheless, to have shown the kind of help that conceptual modelling can provide for lexical knowledge representation. The techniques we have described concerning conceptual modelling and database design are liable to be used under any other formalism. As a matter of fact we would have liked to show how the information, structured in the way we have presented here, is reusable, with minimal changes in the database overall structure, for other linguistic models as long as they are based on valency theory. Data independence, then, seems as important as avoiding data redundancy.

These two issues, it should be clear by now, are to be pursued when the complex task of knowledge representation is undertaken. An appropriate conceptual modelling of the database will help to render a reusable, multifunctional database, which may become the solid basis for a serious natural language processing system.

References

Brodie, Michael L. (1984) "On the Development of Data Models," in Brodie, M.L., Mylopoulos, J. & J.W. Schmidt. *On Conceptual Modelling. Perspectives from Artificial Intelligence, Databases, and Programming Languages*. 19-47. New York: Springen-Verlag.
Barr, Aaron. & Edward A. Feigenbaum (1981): *The Handbook of AI. Volume 1*. Los Altos, CA: William Kaufmann.
Chen, Peter Pin-Shan (1976): "The Entity-Relationship Model -Toward a Unified View of Data." *Transactions on Database Systems*, VI, N1, March 1976, pp. 9-36
——— (1983): *Entity-Relationship Approach to Information Modelling and Analysis*. Proceedings of the Second International Conferemce on Entity-Relationship Approach. Amsterdam: Elsevier Science Publishers.
Codd, E. F. (1990): *The Relational Model for Database Management: Version 2*. Reading, Mass.: Addison-Wesley Publishing.
Coseriu, E. (1981): *Principios de Semántica Estructural*. Madrid: Gredos.
Cruse, D. A. (1986): *Lexical Semantics*. Cambridge: Cambridge University Press.
Date, C.J. (1990): *An Introduction to Database Systems, Volume 1, Fifth Edition*. Reading, Mass.: Addison-Wesley Publishing.
Dik, Simon C. (1989): *The Theory of Functional Grammar. Part 1: The Structure of the Clause*. Dordrecht, Holland/Providence RI-U.S.A.: Foris Publications.
Gardarin, Georges & Patrick Valduriez (1989): *Relational Databases and Knowledge Bases*. Reading, Mass.: Addison-Wesley Publishing.
Hodgson, J. P. E. (1991): *Knowledge Representation and Language in AI*. New York: Ellis Horwood.
Lehrer, A. & E. F. Kittay (1992): *Frames, Fields and Contrasts*. Hillsdale, New Jersey: Lawrence Elbaum.
Mairal Usón, Ricardo (1993): *Complementation Patterns of Cognitive, Physical Perception and Speech Act Verbs in the English Language. A Functional-Cognitive Approach*. Unpublished PhD, Facultad de Filosofia y Letras, Universidad de Zaragoza.
Martín Mingorance, L. (1984): "Lexical fields and Stepwise Lexical Decomposition in a Contrastive English-Spanish Verb-Valency Dictionary". *LEX'eter'83: Proceedings of the International Conference on Lexicography*. Ed. R. Hartmann. Tübingen: Max Niemeyer, 226-237.
——. (1987a): "Classematics in a Functional-Lexematic Grammar of English". *Actas del X Congreso de la Asociación Española de Estudios Anglo-Norteamericanos*. Zaragoza: Publicaciones de la Universidad, 377-382.
——. (1987b): "Semes, Semantic Classemes and Dimensions: The Lexicological and Lexicographic Perspectives". Paper presented at the XIV Congress of Linguists. Berlin.
——. (1987c): "Pragmatic Features in the Lexicon of a Functional Grammar". Paper presented at the International Pragmatics Conference. Antwerp.
——. (1990): "Functional Grammar and Lexematics". In Tomaszczyk, J. and Lewandowska- Tomaszczyk. (Ed.) *Meaning and Lexicography*. Amsterdam: John Benjamins, 227-253.
Mylopoulos, John & H. E. Levesque (1984): "An overview of Knowledge Representation," in *On Conceptual Modelling. Prespectives from Artificial Intelligence, Databases and Programming Languages*. 3-17. New York: Springen-Verlag.
Obermeier, Klaus K. (1989): *Natural Language Processing Technologies in Artificial Intelligence*. Chichester, West Sussex: Ellis Horwood Limited.
Stonebraker, Michael (1988): *Readings in Database Systems*. Introduction to Chapter 6: "New Data Models." San mateo, CA: Morgan Kaufmann Publishers Inc.