# *P*ortuguese *L*exicon *A*cquisition *IN*terface (PLAIN)

José G. P. Lopes, Adelino M. M. Santos

## Abstract

Acquisition of new vocabulary by a natural language understanding system (NLUS), either during an interaction with a user, or during construction of a new NLUS, is a major problem that has attracted researchers' attention. In this paper we describe PLAIN, a graphical interface for interactive semi-automatic generation of monolingual lexicons for NLUSes. We also explain background knowledge that supports PLAIN's use and how it can be ported to acquisition of monolingual and bilingual lexicons.

## 1. Introduction[1]

With PLAIN we aim at:

- having fast and safe development of new lexicons for new NLUSes and making this job pleasant (semi-automatic and interactive) and virtually error free;
- keeping apart, as far as possible, world knowledge representation from linguistic knowledge representation (handling these two kinds of knowledge requires different types of expertise);
- focusing lexicographers' attention on lexicon's content, not on its form. This reduces drastically internal details of NLUSes a lexicographer must know about;
- enhancing reusability of NLUS's building blocks.

In order to achieve these goals, for every application, we require reusability of:

- Portuguese syntax description [LR90];
- application independent lexicons for word morphological analysis and syntactic categorization of determiners, adverbs, prepositions, some verbs and adjectives;
- application independent mapping from syntactic parses into meaning representations. Our research group has adopted Discourse Representation Theory [KR90] for semantic representation;
- application independent dialogue handler [Lo90];
- PLAIN for developing application dependent lexicons.

[1] This work has been supported by JNICT (under contract No. 87439), INIC (under project SAPIA), Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa and Gabinete de Filosofia do Conhecimento.

We defend that behaviour of NLUSes should result from cooperation among a *recognition system* (using morphologic, syntactic, semantic and pragmatic knowledge), a *generating system* (using knowledge sources that are also applyed for recognition), at least one kind of *planning system* (for organizing interaction with users according to the system's own goals), and a *conversation handler* to control NLUS's behaviour ([Lo86; Lo91]). Each of these systems has parts. Each part should be conceived as a knowledge kernel that never changes (i.e. knowledge that can be reused in any other application without further modification) on top of which one adds a layer of knowledge specific for each particular application. For example, our recognizer for Portuguese has three main invariant (reusable) components —Syntactic Kernel, Semantic Kernel and Pragmatic Kernel. On top of these kernels we have an application dependent knowledge layer— the application dependent lexicons. PLAIN has been built as a tool for helping the construction of these lexicons.

PLAIN is currently used for acquiring:

- morphological information about words in a particular context. An unknown word is a verb, an adjective, a proper name, an adverbial, etc;
- information relative to subcategorization[2] of words. This subcategorization will restrict the type of structures where a word can appear.

In future editions it will also be used for acquiring:

- semantic information about nouns, adjectives, proper nouns and verbs. This information includes classification of a word in a hierarchy of types and its semantic representation;
- pragmatic information telling the NLUS dialogue handler if a particular argument is obligatory or not. This information is necessary for system's decisions about relevant questions it must pose to its users ([Lo86]).

Currently PLAIN requires three types of knowledge:

- morphological knowledge about suffixes, prefixes and word formation (this is coded in the lexicon for morphological analysis);
- rules for finding out if a particular word is known. These rules are also used for picking up additional morphological information from the morphological database;
- already existing lexicons.

We impose the following restrictions:

- Application dependent lexicons must be produced in three steps:

  1) through machine text reading and parsing, words not yet known and new word uses are identified;

[2] «Subcategorization, or valence, of a lexical, or a phrase sign, is a specification of the number of and the kind of other signs that the sign in question characteristically combines with in order to become complete», [PS87], p. 678.

2) lexicographers, using an interactive knowledge acquisition interface, are asked to fill in missing information (mostly through a graphic interface dialogue);
3) knowledge engineers using interactive world knowledge acquisition interface will be asked to convey missing information.

- Lexicon revision must be machine controlled.

In the rest of this paper we will elaborate on the contents of two different kinds of lexicons:

- lexicon for syntactic analysis;
- lexicon for morphological analysis.

We will also describe rules that are used for identifying known words using both lexicons. Then we will focus on the algorithm that underlies PLAIN's behaviour and will show some window pictures of currently implemented lexicon acquisition system. After we will elaborate on PLAIN's portability to other natural languages. Finally we just mention how our work compares with related work.

PLAIN was totally implemented using ALPES XProlog environment (*A*dvanced *L*ogic *P*rogramming *E*nvironment*S* was a result of Esprit project - P973] ([Ab89; Ab98b]).

## 2. Lexical knowledge representation

Due to space restrictions, in this paper we will not present examples for lexicon entries. Those of you interested on an expanded version of it should ask us for [LS90].

### 2.1. Lexical knowledge for syntactic analysis

The lexicon schema outlined is currently used for parsing Portuguese sentences ([LR90]). The parser for Portuguese was developed using Extraposition grammars ([Pe81]). The lexicon for syntactic analysis is currently represented as a Prolog database. Each lexicon entry is a Prolog fact.

#### 2.1.1. Entries for common nouns have the form:

dicnoun(Noun, Number, Gender, PFF, Meaning, Subcat)

where **Noun** denotes a common noun being represented, **Number** denotes one of two values: **sin** (for singular) or **plu** (for plural). Most Portuguese nouns are represented in the lexicon in singular form. However there is a limited number of nouns that do not accept a singular form. If this wasn't the case this argument wouldn't be needed. **Gender** denotes word gender: **masc** (for masculine) or **fem** (for feminine), for Portuguese. **PFF** denotes the class of words that can inflect in the gender and in the number using a given rule (cf. sections 2.3 and 2.4). **Meaning** denotes noun genus in a hierarchy of semantic categories[3] and **Subcat** denotes a class of subcategorization.

### 2.1.2. Entries for proper nouns have the form:

dicname(Noun, Number, Gender, Def, PFF, Meaning, Subcat)

where **Noun** denotes a proper noun, **Number** denotes one of two values: singular or plural, **Gender** denotes the gender of the word being represented (masculine or feminine, for Portuguese), **Def** indicates if this noun should be preceded by a defined article or not. In Portuguese most proper names must be preceded by a definite article. Lisbon, as the name of Portugal's capital, will never be preceded by a definite article. Lisbon, as the name of any other thing or person, will be preceded by a definite article. **PFF** denotes the class of words that can inflected in gender and in number using a given rule (cf. sections 2.3 and 2.4). This information will be important for those cases where a proper noun is used as a common noun. This special kind of use is identified during parsing, either by incorrect employment of definiteness, or by occurrence of a proper noun inflected form (example: **all** johns I know are introverted). **Meaning** indicates genus of the entity denoted by that particular name for a specific application. **Subcat** denotes a class of subcategorization. It can be used for compounding proper names.

### 2.1.3. Entries for adverbials have the form:

dicadverb(Adverb, Cat, Subcat)

where **Adverb** denotes the represented adverb and **Cat** denotes values: mode, place, time, intensity, etc. **Subcat** denotes subcategorization class for the represented adverb.

### 2.1.4. Entries for adjectives have the form:

dicadj(Adjective, PFF, Cat, Subcat)

where **Adjective** denotes an adjective, **PFF**, as in preceding explanations for **dicnoun** and **dicname**, denotes the class of words that can inflected in gender and in number using a given rule (cf. sections 2.3 and 2.4). **Cat** denotes the kind of adjective being considered. Currently, it denotes values:

- *temp* for temporal adjectives such as annual, etc.;
- *quant* for adjectives that can be intensified (nice, nicer, very nice...);

[3] In future work it will probably denote a disjunction of possible noun genuses. However, the adoption of such a solution brings along other problems (that are not yet solved) to the description of subcategorization. As a matter of fact it is not yet clear how genus selection for a noun influences its subcategorization.

- *restr* for those that cannot be intensified and restrain meanings of nouns they are modifying (chemical reaction);
- *ordinal* for those that specify an order and precede nouns they are specifying.

**Subcat** denotes a class of subcategorization of the represented adjective. Notice that it is not left any slot for representing gender and number of an adjective. This is due to the fact that adjectives are represented in its basic masculine singular form and this information is implicitly taken into account by lexicon users.

### 2.1.5. Entries for verbs have the form:

dicverb(Verb, SubcatArgO, SubcatArg1, ConjC)

**Verb** stands for the infinitive form of a verb; **SubcatArgO** denotes the syntactic form of verb external argument, currently known as verb subject; and **SubcatArgl** denotes expected syntactic form of verb internal arguments. **ConjC** denotes the conjugation class of the represented verb.

### 2.1.6. Entries for pronouns have the form:

dicpron(W, P, N, G, Cat, Case, PFF, Sem, Sc)

W denotes a pronoun; **P, N** and **G** denote its person, number and gender morphological features; **Cat** denotes one of the values: **dem** (for demonstrative), **indef** (for indefinite), **neg** (for indefinite negative), **pes** (for personal), **int** (for interrogative), **rel** (for relative); **Case** denotes case value(s) the pronoun may assume (it is particularly important for Portuguese personal pronouns); **PFF** denotes class of inflection to which a pronoun belongs; **Sem** denotes most general pronoun genus in a hierarchy of semantic categories; **Sc** denotes pronoun subcategorization.

### 2.1.7. Entries for determiners have the form:

dicdet(W, Num, Gen, PFF, Def, Sc)

W denotes a determiner; **Num** and **Gen** denote its number and gender; **PFF** denotes the class of inflection to which the determiner belongs; **Def** denotes its definiteness. It may represent values: **interrog** (for interrogative), **def** (for definite), **indef** (for indefinite) and **gen**, when there is no determiner. Variable **Sc** denotes determiner subcategorization.

### 2.1.8. Entries for adjective determiners have the form:

dicadjdet(W, Def, Num, Gen, PFF, Cat, Sc)

Variables denote the same kind of things that we have explained previously. This kind of words appear always after determiners. Some of them cannot follow a pronoun or a noun.

## 2.2. Lexicon for syntactic analysis of irregular forms

For words that are the result of irregular conjugation or inflection there is a lexicon, made up of Prolog facts described by:

```
form(Cat.IrregDW, W, MI, Sem, Sc)
```

**Cat** denotes morphological category (noun, verb, adjective, adverb, determiner, etc.) of an irregularly derived word, denoted by **IrregDW**, whose basic form is denoted by **W; MI** denotes a bundle of morphological information about the irregularly derived word (its gender and number, for nouns and adjectives; its tense, mode, person, number and gender, for verbs; and other features that depend on its category); **Sem** denotes semantic type of identified word; **Sc** denotes the class of subcategorization to which it belongs.

## 2.3. Lexicon for morphological analysis

This lexicon is made up of Prolog facts described by syntactic forms:

```
ending(Cat, DMI, End, TEnd, MI, S, OCat)
prefix(Prefix)
```

where **Cat** denotes morphological category of a derived word. It may denote values **v** (for verb), **n** (for noun), **adv** (for adverb), **adj** (for adjective), **pron** (for pronoun), **det** (for determiner) and **adj_det** (for adjectival determiner). Variable **OCat** denotes morphological category of word that is submitted to a derivation process. Variable **End** denotes the ending of derived word, by conjugation (for verbs), by inflection (for nouns, pronouns, adjectives, adverbs and determiners) andy by word formation through substitution of root word endings by suffixes. Variable **TEnd** denotes the ending of a root word, in a derivational process. Variable **DMI** denotes a bundle of morphological information that indicates: tense, mode, person, number and gender of a derived verb form; number, gender, rules for formation of plural and feminine forms for adjectives and nouns derived from adjectives, nouns or verbs; number, gender and rules used for regular inflection of adjectives, nouns, pronouns and determiners, etc; variable **MI** denotes morphological information related to denotation of variables **TEnd** and **OCat** that is required in a derivational process. Variable **S** denotes the suffix identified (conjugation and inflection give rise to a null suffix). **Prefix** denotes a list of characters of Portuguese prefixes.

## 2.4. Morphological analysis and word identification

When it is necessary to find out if a word, denoted by variable **W**, is known one must solve goal:

```
lexicon(Cat, W, BW, MI, Sem, Sc)
```

where **Cat** denotes morphological category (noun, proper noun, pronoun, verb, adjective, adverb, determiner, etc.) of a word represented by **W**, derived somehow from some known word denoted by **BW. MI** denotes morphological information about **W** (its gender and number, for nouns, determiners, adjectives, and other categories; its tense, mode, person, number and gender, for verbs; and additional information important for the morphological category indicated by **Cat**); **Sem** denotes semantic type of identified word; **Sc** denotes its subcategorization class.

Resolution of this goal can be achieved in 4 different ways:

1) either it is a known form of a verb (irregularly conjugated) or of a noun, adjective or specifier (irregulary inflected). Then, basic form of identified word can be found in the lexicon for syntactic analysis of irregular forms (cf. section 2.2):

```
lexicon (Cat,W,BW,MI,Sem,Sc) :- form(Cat,W,BW,MI,Sem,Sc).
```

2) or the word exists in lexicons for syntactic analysis (cf. section 2.1) and resolution of this goal is accomplished through rules as:

```
lexicon(n,W,W,Num+Gen+MPluF+SFemF+PFemF,Sem,Sc) :-
    dicnoun(W,Num,Gen,pff(I),Sem,Sc),
    pff(I,MPluF+SFemF+PFemF).
lexicon(adv,W,W,_,Sem,Sc) :- dicadv(W,Sem,Sc)., etc.
```

These Prolog clauses assure direct consultation of those lexicons. **pff/2** allows determination of the word inflexion class in the masculine plural and in the singular and plural feminine forms. It is indexed by the value of its first argument.

3) or the word is the result of a regular derivation, by prefixation, suffixation, conjugation or inflection. Then, in order to check if this obtains, it is neccessary:

- — to separate prefixes,
- — to identify possible suffix transformations, and
- — to confirm if is there any word with expected morphological category that, through derivation, may generate the word one wants to categorize. If this objective is fulfilled a category is assigned to the apparently not known word.

This process is described by Prolog clause:

```
lexicon(Cat,DW,W,MI,Sem,Sc) :- consult(Cat,DW,W,Mi,Sem,Sc).
```

4) or the word cannot be identified by the system and then, using results of precedent process for word identification, a dialogue for acquiring a new lexicon entry may start.

## 3. PLAIN

PLAIN is a flexible graphical interface for acquisition of new vocabulary. It can be used in two different modes:

- word mode can be automatically used during a parsing process in order to acquire knowledge about unknown words. Instead of a parser trying to cope with unknown words a linguistic knowledge acquisition process may be initiated, in order to get more information about unknown words. This way the parsing process can proceed with less ambiguities to solve and, at the same time, lexicon is improved.
- text mode is used by computational linguistics engineers when they need to create prototypes for entirely new applications. It requires existence of textual corpus for the application. PLAIN will pick up unknown words from those texts. An interaction with a computational lexicographer will start in order to produce new entries for lexicons required by each particular NLUS.

Word mode is useful for the testing phase of NLUS prototypes. During this phase it's not pleasant to receive answers such as *«Bad input; can not parse it!!!»* without initiating a helpful dialogue for acquiring knowledge about words that are not yet known by the NLUS. Word mode will also be important for end users of an NLUS. However, in such case, one must consider end-user's models.

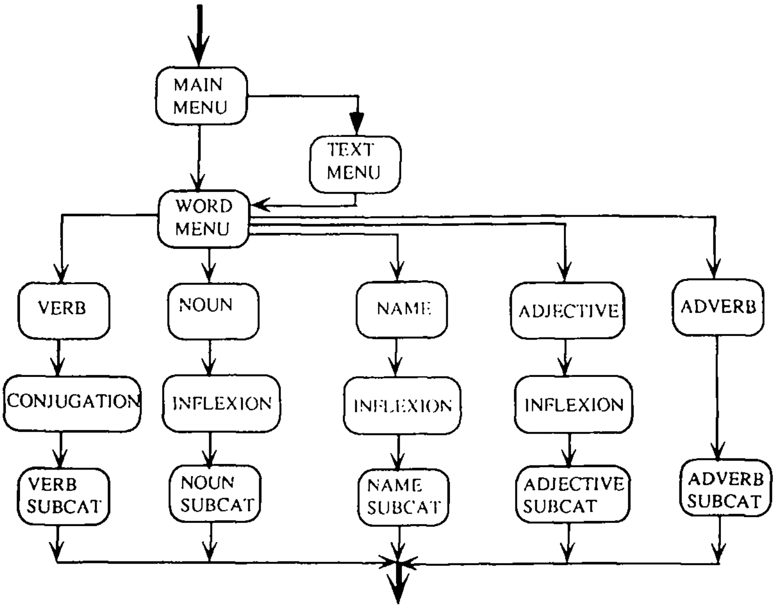
### 3.1. How does PLAIN work

Once a word has been picked up (either from a text or supplied by a user) a graphical dialogue starts. The user is supplied with a window where he/she can choose among various possible categories for the word selected. This work is currently based on consultation of existing lexicon for morphological analysis:

- user selects one or more alternatives for an unknown word's morphological category;
- for each selected category a specific graphic dialogue will start. A lexicographer will confirm or correct information supplied by PLAIN;
- selection of alternative paths (see Figure 3.1) by a lexicographer leads to creation of new lexicon entries and to lexicon updating;
- a new word is picked up and this process restarts.

Paths that a user can select are branches of a menu tree (actually a DAG[4]) sketched below in Figure 3.1:

[4] Direct Acyclic Graph.



Back to WORD MENU

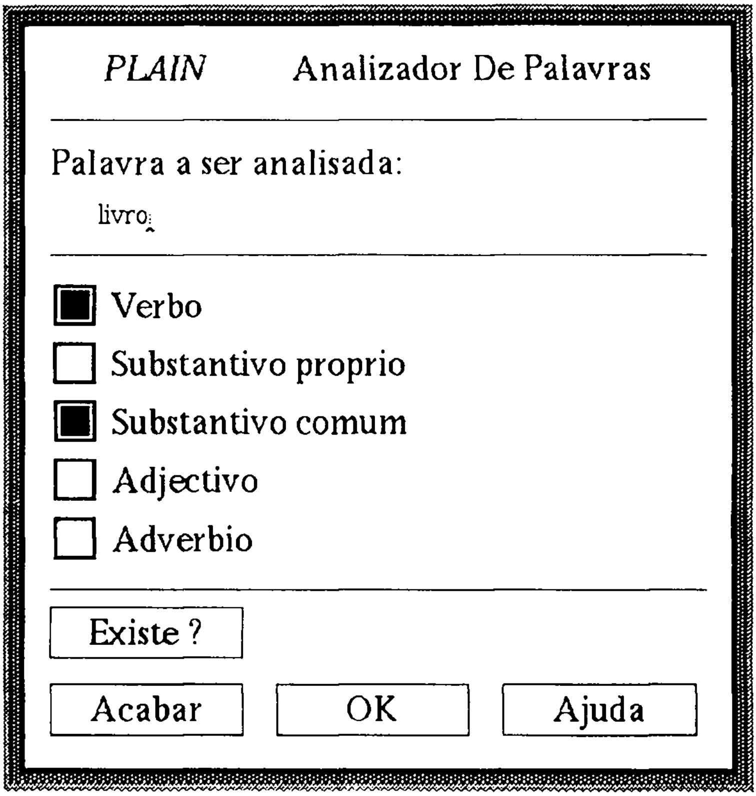
Figure 3.1 Branches of a menu DAG for building lexicon entries

As you can see from illustrations in next subsection, in all stages of activity, a user may press any of the buttons:

- **help ('AJUDA')**, for obtaining additional information about the meaning of a value, in a given situation;
- **cancel ('ACABAR')**, for cancelling what the user has just been doing (as a result of clicking this button, PLAIN goes to a previously confirmed knowledge state);
- **ok**, for confirming information displayed for a certain state (by clicking this button the user is directed to a another knowledge state in the menu tree).

## 3.2. Illustration of PLAIN's behaviour

In Figure 3.2 we display a window for acquisition of Portuguese words. There is shown word *'livro'* which can be either a noun *('substantivo comum')* or a verb *('verbo')*. These two possibilities are selected by clicking the corresponding square buttons. Then a specific graphic dialogue would start for acquiring information about word *'livro'* taken as a noun. Once this data has been obtained, the dialogue will continue in order to pick up additional facts about same word taken as a verb.



*PLAIN* Analizador De Palavras

Palavra a ser analisada:

livro

☒ Verbo

□ Substantivo proprio

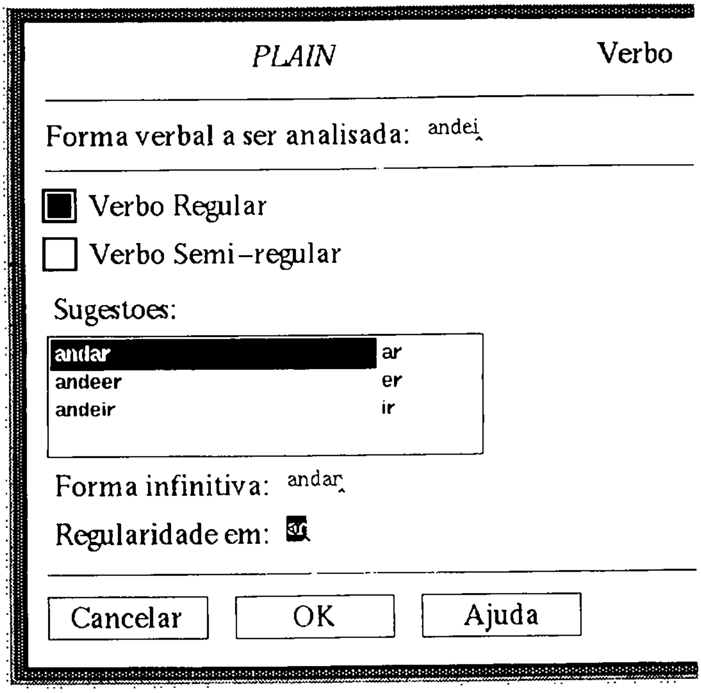
☒ Substantivo comum

□ Adjectivo

□ Adverbio

Fig. 3.2 Window for acquisition of Portuguese words.
In this particular example, information about word 'livro' is at stake.

Due to space limitations it is not possible to show and explain more steps of the graphic dialogue that follows. However it's worth saying that once a category is chosen for a word, PLAIN uses its morphological knowledge and displays different possibilities for word's basic form. Together with these possibilities there is information used for deriving that form. So the lexicographer just has to choose the correct basic form together with rules to derive it.



*PLAIN* Verbo

Forma verbal a ser analisada: andei

■ Verbo Regular

□ Verbo Semi-regular

Sugestoes:

| | |
|---|---|
| andar | ar |
| andeer | er |
| andeir | ir |

Forma infinitiva: andar

Regularidade em: ar

Fig. 3.3 Window for data acquisition for verbs.

For example, in Figure 3.3 it is shown part of a window for acquiring data about verbal word form '*andei*' (I walked). Knowing that it is a verb, allows PLAIN to infer that its infinitive form can be either '*andar*' (which is the correct form) or '*andeer*' or '*andeir*' (that do not correspond to existing verbs in Portuguese). Selection of '*andar*' (a verb ending in '-*ar*') and of regular conjugation button is sufficiently informative for allowing PLAIN to conjugate this verb. If verb wasn't regular then PLAIN would conjugate it as a regular verb and would invite the user to correct all forms incorrectly conjugated. Then would prepare entries for the irregular forms lexicon. This procedure has an inconvenient does not allow PLAIN to capture new classes of conjugation regularity.

### 3.3. Preliminary text treatment

When text mode is chosen, textual corpus is submitted to a preliminary treatment — a filtering process acts upon texts *throwing out* every known term and leaving just unknown ones for classification.

This filtering process is implemented in C/LexN[5] (a shell process is created and a command is invoked that takes as input the text file and processes as output a new file). The link from XProlog to C enables an appreciable gain in performance since it is necessary to build a data structure to store all words (to detect repetitions) in the text and to consult the database (to check for already known terms).

However this treatment poses problems. The severest one is related to destruction of text structure. A text becomes a bunch of words, with no organization or context notion, which brings some undesirable results, from the point of view of lexicography (especially for a restricted application area), where it is important to know where a word appears in a text in order to support its subcategorization. However we plan to have PLAIN working together with a text searching tool[6] to overcome this problem.

### 3.4. Portability

PLAIN was implemented using ALPES XProlog ([Ab89], [Ab89b]). This language has an interface with X Window System Toolkit that enables easy use of windows and alike concepts extensively employed in this implementation.

Portability of PLAIN to another knowledge representation or natural language poses no problem at all if one has a specification of lexicon schemas. Some cosmetic arrangements will be necessary, in order to have window buttons and messages written in that NL and the possibility of having acquisition of other kind of information.

Changes to PLAIN are easy due to declarative style of programming used.

## 4. Future work

Work currently under development is aimed for constructing a generator of graphical interfaces. We intend to have a tool with which we can rapidly change graphical form and content of interface windows. This generator will have an editor for plugging in desired functionality to each graphic object identifiable in a window. Such a tool will enable us to build new graphical interfaces adapted to each kind of task and type of user.

## 5. Related work

As we want to have graphical interfaces for lexical acquisition adapted to specific classes of users, plugged in robust NLUSes in order to allow them to cope with non-canonic input (incorrect, correct but unrecognizable, elliptic,...), our work relates (but doesn't overlap) with well known publications on acquisition from correct input and from machine readable dictionaries. However, due to space limitations, we will not elaborate on this subject in this paper.

## References

[AB89] ABREU, Salvador Pinto. «A Prolog interface to the X Window System Toolkit». In *Proceedings of tne NACLP'89 Workshop on Logic Programming Environments: The Next Generation*, 1-9.

[Ab89b] ABREU, Salvador Pinto. *Alpes X-Prolog Programming Manual*. CRIA/UNINOVA, Monte da Caparica, Portugal, 1989.

[LR90] KAMP, Hans and REYLE, Uwe, 1990. *From Discourse to Logic - An introduction to model theoretic semantics of natural language, formal logic and discourse representation theory*. Institute for Computational Linguistics, University of Stuttgart, Germany.

[Lo86] LOPES, J.G.P. 1986. *Conceptualization of an Automatic Interlocutor System*. Ph.D. Thesis. Instituto Superior Técnico. Universidade Técnica de Lisboa. (In Portuguese).

[Lo91] LOPES, J.G.P. 1991. *Architecture for Intentional Participation of Natural Language Interfaces in Conversations*. Proceedings of the Third Natural Language Understanding and Logic Programming 1991.

[LR90] LOPES, J.G.P. & RODRIGUES, I.P. 1990. *Partial Description of Portuguese Syntax*. Technical Report CRIA/Uninova, Monte da Caparica. (In Portuguese).

[LS90] LOPES, J.G.P. and SANTOS, A.M.M. 1990. *Portuguese Lexicon Acquisition INterface: PLAIN*. Uninova/CRIA Technical Report, October 1990.

[Pe81] PEREIRA, F.C.N. 1981. «Extraposition Grammars». *American Journal of Computational Linguistics* 7(4), pp. 243-255.

[PS87] POLLARD, C. & SAG, I. 1987. *Information Based - Syntax and Semantics, Fundamentals* Vol. 1. CSLI Lecture notes number 13. Stanford.