Integration of NLP Tools in an Intelligent Computer Assisted Language Learning Environment for Basque: IDAZKIDE

A. Díaz de Ilarraza, A. Maritxalar, M. Maritxalar and M. Oronoz
Dept. of Computer Languages and Systems.
University of the Basque Country. 649 Postakutxa. 20080 Donostia. Spain.
{jipdisaa, jibmagoa, jipmaanm, jiboranm}@si.ehu.es
URL: http://ixa.si.ehu.es

Abstract

This paper presents IDAZKIDE, a prototype of an intelligent language learning environment (ICALL) for learners of Basque. The philosophy of the system is to make different Natural Language Processing tools simultaneously accessible to students to help them (mainly at morphological level) write in Basque as well as to give adequate advice taking into account some characteristics of the student gathered in a student model.

Keywords: NLP, ICALL, Corpora, Student Model, Morphology, Second Language Acquisition.

1 Introduction

The main goal of this article is to explain a system that helps students write using NLP tools. We collect some characteristics of the students in a student model and use them to adapt the environment. This environment provides a graphical user interface that allows the learner, among other operations, to look up in a dictionary, to analyse morphologically a word-form, to check if it is correct and ask for proposals.

In the near future, when the student model already developed (see section 3) is integrated in the interface, the system will correct texts written by learners according to their knowledge level in Basque. In HITES (a system for modelling language learners' interlanguage), we represent each language level by means of the corresponding interlanguage model. The linking between IDAZKIDE and HITES has already been designed but the implementation of the communication between both systems is still in project. In this article we will only explain IDAZKIDE. Information about HITES can be found in (Maritxalar et al. 97).

IDAZKIDE can be considered an interesting tool for two reasons: first, Spanish-speaking learners of Basque usually find several problems in learning the morphology of Basque due to the fact that it is an agglutinative language and second, most learners do not have the necessary knowledge about the rules for the normalisation of the language periodically published by the Basque Language Academy (Euskaltzaindia); it must be taken into account that, as the normalisation process of Basque language started at the end of the sixties, there are still native speakers of Basque who are in an alphabetisation process: IDAZKIDE is also useful for this type of learners.

For a better understanding of the usefulness of our tool, first we will do a brief description of the framework of the project. The most interesting modules for our purposes, i.e. the student model, the NLP tools module and the interface, will be described in sections three, five and six respectively. In the sixth section we will also demonstrate how IDAZKIDE works. Finally, we will comment on some conclusions and ideas about future work.

2 The framework of IDAZKIDE

In this section we will explain the linguistic context of IDAZKIDE by means of a description of Basque, and then explain the relationship of IDAZKIDE with HITES and IRAKAZI (a module for teachers).

2.1 Brief Description of Basque Morphology.

Basque is an agglutinative language; that is, for the formation of words the dictionary entry independently takes each of the elements necessary for the different functions (declension case included). More specifically, the affixes corresponding to the determinant, number and declension case are taken in this order and independently of each other.

One of the principal features of Basque is its declension system with numerous cases, which differentiates it from the Romance languages in contact with it. The inflections of determination, number and case appear only after the last element in the noun phrase. This last element may be the noun, but also typically an adjective or a determiner. In Figure 1 there is an example of the analysis of *semeArEN etxeAN* ('in the house of the son').

seme	A	r	EN	*etxe*	A	N
noun 'son'	det. 'the'	epenthetical element	genitive case 'of'	Noun 'house'	Det. 'the'	inessive case 'in'

Figure 1: Analysis of *semeArEN etxeAN*

As prepositional functions are realised by case suffixes inside word-forms, Basque presents a relatively high power to generate inflected word-forms. We can have combinations of number, determination, and case marks, not capable of further inflection, but we could also have word-forms ended with one of the two possible genitives or with a sequence composed of a case mark and a genitive mark. If the latter is the case, a set of morpheme combinations could be added again recursively generating a complete set of forms. It could be theoretically extended *ad infinitum*; but in practice, it is not usual to find more than two levels of this kind of recursion.

e.g.	*semeA*	(the son)
	semeArI	(to the son)
	semeArEN	(of the son)
	semeArEN etxeA	(the house of the son)
	semeArENA	(the one (house) of the son)
	semeArENArI	(to the one (house) of the son)

All the features of Basque have been taken into account in the NLP tools in IDAZKIDE making it robust and rich.

2.2 The Interlanguage Level Models

IDAZKIDE has been developed in a framework where other two different systems have evolved: HITES and IRAKAZI. HITES interacts with a psycholinguist who gives hints to the system in the construction of interlanguage models for each language level. IRAKAZI works with the language teacher so as to give more accurate information about particular learners. We create two types of models: the interlanguage level models and the student models having as a basis the information collected by both systems and the details given by the learner in IDAZKIDE. The three systems we have mentioned share those models.

The interlanguage level models adopt a differential perspective (Self 92). They are multiple models. As Self says: *A novice's knowledge may be a conceptualisation fundamentally different from the expert's. Hence the former cannot be represented by small perturbations of the latter.*

When defining the interlanguage, we consider both student's standard and deviant linguistic structures (Díaz de Ilarraza et al. 98) and the context in which they appear. Depending on the knowledge level of the student, the same structure can be deviant in a high knowledge level and standard in a low level where the student hasn't learnt it. That's one of the reasons of speaking about deviations instead of errors. Besides, *error* is a term referred to the standard language. Based on empirical work previously performed (Andueza et al., 95) we concluded the necessity of having, at the beginning, a standard set of user models for each language level. We built the first versions of these models from the analysis of a corpus made up of about 300 texts from two different schools specialised in the teaching of Basque. After defining the interlanguage by levels, we will use it in IDAZKIDE to help the students by giving them the most adequate information for their level.

The student models are descriptive models whose characteristics are used by the three systems mentioned.

We differentiate two submodels inside the student model:

- The kernel is composed of the corresponding interlanguage level model and general characteristics of the student relative to the learning process.
- The student model of the session gathers the characteristics of the student that are important for the interaction with the learner in IDAZKIDE.

In section 3, we will explain the student model of the session. Information about interlanguage level models of the kernel can be found in (Maritxalar et al., 97).

3 The Student Model

IDAZKIDE gathers the students' model in order to guide their learning process and give information according to their preferences during the writing. The main feature of the student model is that we find inside the system only a part of the model. That means that another part of the model is outside: the kernel of the student model. This model, as well as the

interlanguage level models, is shared by different systems. In the case of IDAZKIDE, interactions with the user will actualise and adapt the model of the language level towards a personal one.

3.1 The student model of a session

The knowledge we consider interesting in the model of a session is:

- Texts written by the learner.
- Tools used by the learner when examining words (in or outside the texts).
- Modifications made by the learner in the kernel of the student model.
- Questions answered by the learner at the beginning and at the end of the session.
- The student's reaction when the system gives her/him hints.

Inside the student model of the session we collect information about the learner detected during the interaction with the system. This is the dynamic part. Moreover, the model has a permanent part, and at the end of the session the system updates its information, taking into account the information gathered in the dynamic submodel.

3. 1. 1 The permanent submodel

In the permanent submodel we collect information about the last sessions. As we will see in the definition of the objects we identify the knowledge and the learning strategies detected in the sessions.

```
    {{Sesion_permanent_model
      (is-a student_learner_concept)
         (multislot conclusion_unit: instance-name of the
class
                  permanent_unit)
    }}
    {{Permanent_unit
      (is-a student_learner_concept)
         (slot use_factor: float)
    }}
    {{Permanent_knowledge
      (is-a Permanent_unit)
         (slot type: symbol) -> allowed-symbols: lemma;
rule.
         (slot value: instance-name of the class
Interlanguage-
                  structure)1
    }}
    {{Permanent_strategy
      (is-a Permanent_unit).
         (slot type: symbol) -> allowed-symbols: cognitive;
                                metacognitive
         (slot identifier: instance-name of the class
                  Student_learning_strategie)
    }}
```

Figure 2: Definition of the permanent student-model of the session

3.1.2 The dynamic submodel

The life of this submodel finishes when the session ends. Next, we will see the units the student can examine during a session: *word_unit* or *text_unit*.

```
    {{Sesion_dynamic_model
      (is-a student_learner_concept)
         (multislot theme_unit: instance-name of the class
                  Dynamic_unit)
    }}
    {{Dynamic_unit
      (is-a student_learner_concept)
         (slot identifier: symbol)
         (slot type: symbol) -> allowed-symbols: word;
text.
    }}
    {{Text_unit
      (is-a Dynamic_unit)
         (slot identifier: string)
         (multislot deviations: instance-name of the class
                  Deviation2)
         (multislot consulted_words: instance-name of the
                                class Word_unit)
    }}
    {{Word_unit
      (is-a Dynamic_unit)
         (slot value: string)
         (multislot texts: string)
         (multislot tools: instance-name of the class Tool)
         (slot session_appearances: integer)
    }}
    {{Tool
      (is-a IDAZKIDE_concept)
         (slot identifier: string)
         (multislot learning_strategies: instance-name of
the
         class Learning_strategy)
    }}
```

Figure 3: Definition of the dynamic submodel

The slot *tools* of the object *Word_unit* identifies the tools used by the learner when examining the word. We must not forget that these tools give us information about the learning strategies used by the learner.

In the case of the *Text_unit* object we gather the deviations detected by the system. If the learner decides to examine any deviant word by means of the tools of the interface, an instance of *Word_unit* is created for such deviation.

4 Architecture of IDAZKIDE

IDAZKIDE is not an intelligent language *teaching* environment, but an intelligent

1 See (Maritxalar et al., 97)

2 Subclass of *Interlanguage_structure*

language *learning* environment: the student learns from the system as well as the system does from the student. The main aim is to help the students during their learning process giving them hints. To do that, it includes information about the learner's characteristics and language knowledge gathered in the student model. Moreover, we include in the system, i.e. in the linguistic tools, information about the language knowledge of students at the same language level. When students use the tool some information about their preferences is gathered from that interaction. In this way, the system will adapt the next interactions with that student.

Note that the system doesn't have teaching material previously stored as most of the CALL systems do (Zubiri 90), but it has as a basis wide-coverage NLP tools. The independence of these tools is very interesting in IDAZKIDE. The linguistic tools can be used apart from the interface. Separating the linguistic tools from the user interface in this way you make it possible to experiment with different user interfaces. So, IDAZKIDE is composed of the following modules: a set of wide-coverage linguistic tools, an adaptative user interface (Downton, 91), a student model, an advisor and a scheduler.

The system gives the possibility of using three types of learning: guided learning, free exploration and mixed free-guided learning, allowing the students to navigate as they want.

The scheduler decides which linguistic tool must run when using guided learning or mixed free-guided learning. In the guiding process, the scheduler takes into account the students' exploration, i.e. the tools they are using, and the different characteristics gathered in the student model. When using free exploration the student decides which linguistic tool the scheduler must call.

The advisor adapts the answer of the system for the student using the information of the student model. However the scheduler is which really activates the advisor.

The architecture of the system is shown in Figure 4:

Figure 4: Architecture of IDAZKIDE

5 Integration of Linguistic Tools: NLP in IDAZKIDE

The work we present in this paper has been developed inside a general project for NLP of Basque. The linguistic tools used in IDAZKIDE have been designed and implemented taking as a basis the tools previously developed in our research group. Besides, the tools don't use a list of words as a basis for analysing, correcting or generating word-forms, but a lexical database of about 70,000 entries from which the system generates or analyses the word-forms. The NLP tools to which the students can access are the following.

- XUXEN3: a spelling-checker corrector (Aduriz et al., 97a; Agirre et al., 92), which detects deviant words in a text, or given a word-form shows if it is correct or not. The adapted corrector will correct students' text depending on their knowledge level of Basque and their characteristics. For example, it won't mark those words that are out of the language level corresponding to the student. This idea won't be able until the communication with HITES is implemented.
- EDBL3: a lexical database for Basque (Agirre et al. 95; Aduriz et al. 98) with 70,000 entries that offers, among others, the possibility of consulting the category and subcategory of words together with examples of use. It gives lexical information about the word even if it appears inflected in the text. Before consulting the word the system lemmatises the word in order to find the corresponding entry in the lexical database. We must consider that we need a lexical database for each language level, in consequence, new attributes will have to be in the database in order to store this kind of information.
- A dictionary where the student can look up the definition of the chosen word. In this case, as in EDBL, the tool lemmatises the

3 Registered Trademark

word given by the student before looking up it.

- A correct word-form proposer that, given a deviant word-form, assists the student offering similar correct proposals taking into account particular characteristics of the student, as language level, objectives in the session, etc.
- A morphological analyser (Alegria et al., 96) that could be useful for students who want to know the grammatical rules used in Basque morphology in a straightforward manner. That analyser is based on the Two-Level Morphology (Koskenniemi 83), and tries to give the analysis of the given word even if the word is unknown for the system. In the analyser the control and knowledge are independent, and, thus, it is possible to adapt it for new goals, a learning system, in our case. We have mainly changed the knowledge and, we have also made some changes in the control. As an example, the initial analyser didn't detect the applied rules in the analysis. The adapted one, however, gives for each interpretation the rules that have been applied. In that way the students could be able to realise that they have used a deviant language structure. If they don't know why it is deviant, they can ask for proposals getting feedback on their work.
- A generator of correct word-forms with which the students can get all possible declension-forms associated to a given lemma. The tool also declines a lemma, given the desired case for its declension and the number. The generator we have developed has as a basis our morphological analyser that makes analysis and generation. This is one of the most interesting features of the Two-Level Morphology that has been used as theoretical basis in the analyser.

6 The Interface in a Session with IDAZKIDE

In this section we will show you some of the most interesting options to which a student can access in a session using IDAZKIDE. Before that we will present some ideas about the interface.

Two different platforms are offered in the interface to work with the tools. One in which the students can pose their questions about a specific word and check its morphology and another one for doing the same with the entire text. Both platforms can offer guided learning, free exploration or a combination of both. The first two update the student model, so in this way some advice could be given. That doesn't happen navigating freely through the interface. In the student model, we basically strive to model the Basque language knowledge of the students without forgetting the modelling of the students' preferences and, their behaviour in different work sessions. We understand as behaviour the type of deviations the students make and the use they make of the tools. Bearing this in mind, the system will adapt the advice for each student depending on her/his profile, making our interface adaptative. Thus, the system can be used with a double purpose: to check freely any students' doubt or to correct their texts.

IDAZKIDE has a Motif-style graphical user interface. Since this is a standard for many X Window System applications, the time needed to learn how to interact with IDAZKIDE is minimised. Moreover, we have designed the interface in a way in which the student can access to a specific tool window to solve a particular doubt without starting from the main screen.

After having seen some features of the interface, next we will simulate a typical session of a student using IDAZKIDE.

When the students start the work session, they find the screen shown at the top of Figure 5. In that screen the students have the possibility of working with a previous written text or with a text that is being edited at that moment. They can use the options of a typical text editor such as file management (open, close, save, save as…) and edition (cut, paste…). With the chosen text, the system could help them write it using the NLP tools above mentioned: corrector, generator, dictionary, proposer, analyser and lexical database. Apart from the linguistic tools, students can find three helping tools: a note-taking tool, a summarisation tool and a self-evaluation tool.

The generator and the proposer are of special interest due to the features explained in subsection 2.1.

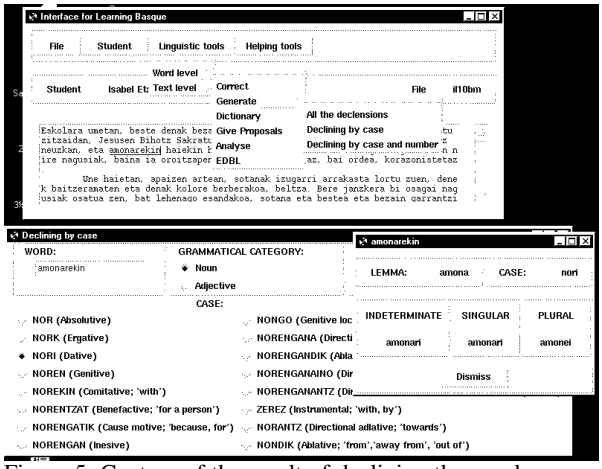

Figure 5: Capture of the result of declining the word *amona*.

The options in the generator to decline a word-form are shown in Figure 5. In this example, the student wants to decline the word *amonarekin* 'with the grandmother' in a specific case, for example 'dative' (Declining by name) but s/he doesn't know which the lemma of the word is. The system is able to find the lemma of the word (*amona* 'grandmother') using the lemmatiser and then decline it in the asked case. The result of the action is shown in the small window in Figure 5 (*amonari* 'to any indeterminate grandmother', *amonari* 'to the grandmother' and *amonei* 'to the grandmothers'). Due to the agglutinative characteristic of Basque, among others, we have found a problem when choosing the lemma for *amonarekin*. In the example below, we can see that there is more than one possibility for that word:

amonA + arekin 'with the grandmother'
ama + on + arekin
amA + on 'of us, the mothers'
amA+ on + arekin 'with the one of us, the mothers'

In the first option there is only one case suffix and in the second there are two. After some experiments with students we have concluded that when the student has little knowledge of the language, it is more probable that they use the longest lemma because it will probably have less suffixes. However, this heuristic must be better studied and revised.

There are another two options to use the generator. First, as well as in the option explained before, the students can select not only the case but also the number (indeterminate, singular or plural) (*Declining by case and number*). Second, given a word form, declined or not, the students can get all the possible inflected forms for the first level of declension (see subsection 2.1)(*All the declensions*).

We have said that the generator is useful for those students who don't know how to decline a word. They can use the proposer, among other purposes, to find the correct lemma starting from a deviant one (declined or not). Next, we will explain how the proposer works.

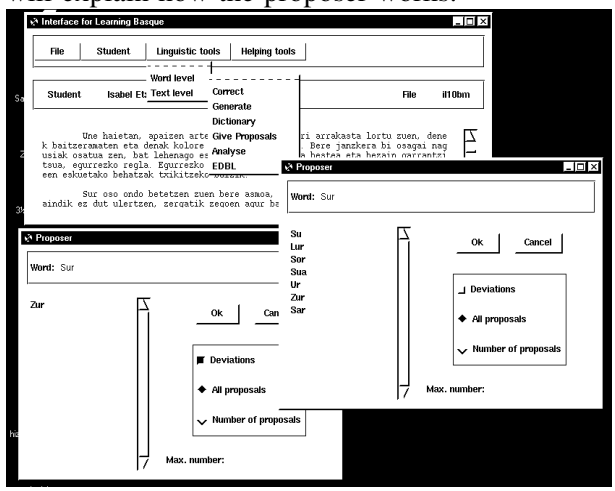

Figure 6: Capture of the result of proposals for the word **sur*

In the example above (fig. 6), the student doesn't know with certainty how to write the word wood (*zur*). S/he knows that it is written similarly to **sur*, so s/he can call the proposer with that word. S/he will get all the possible proposals for that word or limit the number of answers. In order to limit the answers s/he can specify a maximum number (with that option the system will propose the words with more similarities) or select the option for deviations. In the example above, when asking for all the proposals, the student has got seven words, but when asking only for typical deviations, s/he has got one word. That occurs because the unique word between the seven with typical deviations is *zur* (replacing z with s, a very typical deviation in Basque).

7 Conclusions and Future Work

In the article we have explained IDAZKIDE, a prototype of an ICALL environment for learners of Basque. It runs on UNIX and the interfaces have been implemented using TCL/TK while the tools are coded in C++. Some adapted NLP tools have been integrated in the system and are accessible to the students in order to help them write. They were developed in the context of a wide framework of NLP research on Basque and each tool is independent of the others. We have seen in our experiments that **systems for learning languages must have a strong basis in the computational linguistics field**.

We have also concluded **that the study of learning strategies could give hints for the design of interfaces**. The appearance of the interfaces could be adapted to the strategies that each student follows, but this area, the human-computer interaction field, is out of our research by the moment.

IRAKAZI, a system for language teachers has been already developed. At the moment, we include in it information about characteristics of the student important for the learning process (age, mother tongue, level of language knowledge, motivation and goals for learning Basque, preferences when learning…), and language structures used by the student in texts. In the future, we can use that information in **the diagnosis process**.

We must not forget the **work at sentence level** done inside our research group (Aduriz et al. 97b; Gojenola et al. 94). We have experimented with two different approaches: the Constraint Grammar (Karlsson et al. 95) and PATR-II. However, by the time, we must still study how to adapt them to IDAZKIDE.

During a two-year period we have been working with a pair of schools specialised in the teaching of Basque. They seem to be very interested in that kind of tools, which help in the learning of the language without imposing its own method, which could be quite different from the method used in the school. We consider that the field of the schools specialised in the teaching of Basque and primary and secondary schools in which the teaching is done in Basque is quite a big area of potential users. Another one is the public administration.

To make IDAZKIDE a commercial product we need to improve the speed of the tools. We are trying to do so using new finite state tools (we have already used these tools in the spelling/corrector making it 100 times faster).

Finally, we must say that we are analysing two possible ways in which we can market our tool: via Internet or as a specific product. We think that our decision must be deeply studied due to the fact that our tools use a lot of different software packages and are distributed in several machines running UNIX.

Acknowledgements

This work is being developed in the framework of a research project sponsored by the University of the Basque Country and the Government of the Province of Gipuzkoa. The authors would like to acknowledge both institutions.

References

(Aduriz et al. 98) I. Aduriz, I. Aldezabal, O. Ansa , X. Artola, A. Díaz de Ilarraza, J.M. Insausti, *EDBL: a Multi-Purposed Lexical Support for Treatment of Basque*, Proceedings of the First International Conference on Language Resources and Evaluation, Vol. II. pp 821-826, Granada. May 28-30, 1998.

(Aduriz et al. 97a) I. Aduriz, I. Alegria, X. Artola, N. Ezeiza, K. Sarasola, M. Urkia*, A spelling corrector for Basque based on morphology,* Literary & Linguistic Computing, Vol. 12, No. 1. Oxford University Press, Oxford, 1997.

(Aduriz et al. 97b) I. Aduriz, J.M. Arriola, X. Artola, A. Díaz de Ilarraza, K. Gojenola, M. Maritxalar, *Morphosyntactic disambiguation for Basque based on the Constraint Grammar Formalism*, Recent Advances in NLP (RANLP97), Tzigov Chark (Bulgary), 1997.

(Aduriz et al. 96) I. Aduriz , I. Aldezabal, I. Alegria, X. Artola, N. Ezeiza, R. Urizar, *EUSLEM: A lemmatiser/Tagger for Basque*, EURALEX'96, Göteborg (Sweden), 1996.

(Agirre et al. 95) E. Agirre, X. Arregi, J.M. Arriola, X. Artola, A. Díaz De Ilarraza, J.M. Insausti, K. Sarasola*, Different issues in the design of a general-purpose Lexical Database for Basque,* First Workshop on Applications of Natural Language to Databases, Versailles, 1995.

(Agirre et al. 92) E. Agirre, I. Alegria, X. Arregi, X. Artola, A. Díaz de Ilarraza, M. Maritxalar, K. Sarasola, M. Urkia*, Xuxen: 1 Spelling Checker/Corrector for Basque based in Two-Level Morphology,* Proceedings of ANLP'92, 119-125, Povo Trento, 1992.

(Alegria et al. 96) I. Alegria, X. Artola, K. Sarasola, M. Urkia, *Automatic morphological analysis of Basque,* Literary & Linguistic Computing Vol. 11, No. 4, Oxford University Press, Oxford, 1996.

(Andueza et al., 95) A. Andueza, A. Díaz de Ilarraza, M. Maritxalar, J. Martiarena, I. Pikabea, *Hizkuntza baten Ikaskuntza Prozesuari buruzko landa lana. Sistema Informatiko adimendun baten oinarria,* Internal Report, UPV/EHU/LSI/TR 8-96, Computer Science Faculty: University of the Basque Country, 1996.

(Díaz de Ilarraza et al., 98) Díaz de Ilarraza A., Maritxalar M., Oronoz M. *An Implemented Interlanguage Model for Learners of Basque*, Language Teaching and Language Technology. Swets and Zeitlinger (Publisher). Sake Jager, John Nerbonne and Arthur van Essen editors. pp 149-166. Lisse. 1998.

(Dowton 91) A. Downton, *Engineering the human-computer interface,* The Essex Series in Telecommunication and Information Systems, McGraw-Hill, London, 1991.

(Gojenola et al. 94) K. Gojenola, K. Sarasola, *Aplicación de la Relajación Gradual de Restricciones para la Detección y Corrección de Errores Sintácticos,*. Boletín de la SEPLN, Córdoba, 1994.

(Karlsson et al. 95) F. Karlsson, A. Voutilainen, J. Heikkilä, A. Anttila*, Constraint Grammar: a language-independent system for parsing unrestricted texts,* Mouton de Gruyter, 1995.

(Koskenniemi 83) K. Koskenniemi, *Two level morphology:a General Computational Model for Word-Form Recognition and Production.* University of Helsinki, Departament of General Linguistics, Publications nº 11, 1983.

(Maritxalar et al. 97) M. Maritxalar, A. Díaz de Ilarraza, M. Oronoz, *From Psycholinguistic Modelling of Interlanguage in Second Language Acquisition to a Computational Model,* Computational Natural Language Learning (CoNLL97), In conjunction with ACL'97/EACL'97 Madrid, Spain, 11-12th July 1997.

(Maritxalar et al. 93) M. Maritxalar, A. Díaz de Ilarraza, *Integration of Natural Language Techniques in the ICALL System Field: The treatment of incorrect knowledge,* Internal Report. UPV/EHU/LSI/TR 9-93, Computer Science Faculty: University of the Basque Country, 1993.

(Self 92) J. Self, *A framework for Learner Modelling,* Interactive Learning Environments Vol. 2, Issue (2), 111-137, 1992.

(Zubiri 90) I. Zubiri, *Ikas-Txip: euskara ordenadore bidez,* Zutabe 24, Donostia, H.A.B.E., 9-55, 1990.