

# Description and Evaluation of a Portuguese AAC System

*Luis Garcia*

ESTIG  
Afonso III, 1, 7800 BEJA, PORTUGAL  
luisbgarcia@estig.ipbeja.pt

*Luís de Oliveira*

INESC-ID/IST  
R. Alves Redol, 9, Lisboa  
lco@inesc-id.pt

## Abstract

This paper describes the development and evaluation of an Augmentative and Alternative Communication (AAC) System for the European Portuguese. The system works on top of the Windows operating system and it can interact with any application that requires keyboard input. The system is targeted to motor impaired users and offers two rate enhancement techniques: word prediction and abbreviation expansion. People with difficulties using a computer keyboard can use our system's on-screen keyboard with different access methods. Aiming at reinforcing the user's interaction, our system also includes an interface agent and a speech synthesizer. Each component of the system can be configured to the specific requirements of each user. Tests were carried out qualitatively, evaluating the performance with real users, and quantitatively, by using a simulator included in our system. These tests showed that our system can save up to 50% of user typing effort.

## 1 Introduction

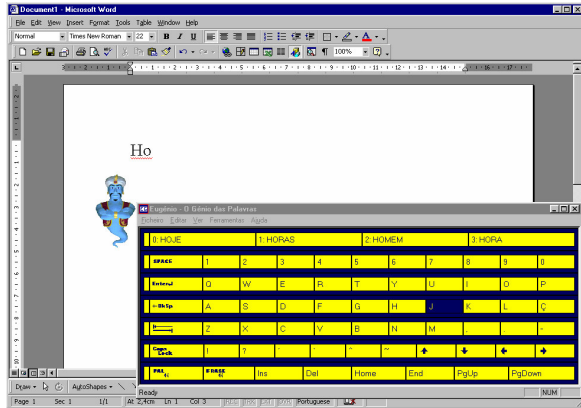
Certain physical or cognitive disabilities can total or partially reduce the human communication and several technologies to support and enhance the communication process have been developed. Among these technologies we can find the word predictors, whose main function consists in writing acceleration, where the vast majority is targeted for the English Language. Although many accessibility tools can be easily converted for other languages, that is not the case of Augmentative and Alternative Communication Systems (AACS) where this localization is always a problem. For instance, some word prediction systems for the English Language don't allow the use of accentuated words, that makes them impossible to use for certain languages like Portuguese. Besides these reasons, the different techniques used by these systems still need substantial improvements so that they can be really effective. In a study, Lesh, Moulton, Higginbotham & Alsofrom (2002) argue that there are still some improvements to be made in word prediction systems.

Our main objective in implementing a word predictor for the Portuguese language was at first the development of a statistically based system that, together with other already available tools like on-screen keyboards and speech synthesizers, could be used to create AAC solutions for a wide scope of users. Some experiments showed us the difficulty of integrating different sources into an AAC system. Therefore we extended the scope of our project and decided to develop a more general system, incorporating all the necessary components for the development of multiple AAC solutions for Portuguese. Figure 1 shows this system working with Ms Word.

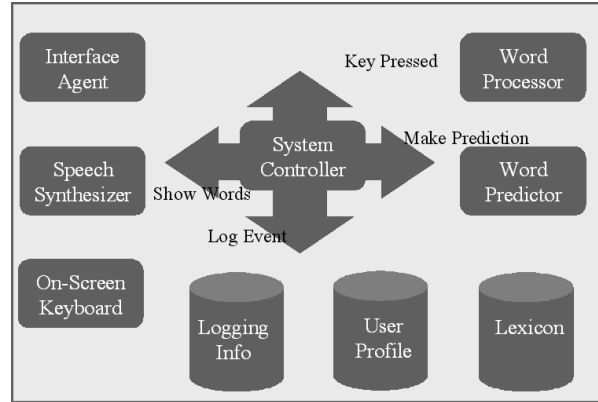
## 2 System Architecture

In conceptual terms our system is based on the structure proposed by Cook & Hussey (1995) for an AACS. This structure is composed by three different elements: (1) Processor; (2) User Interface; (3) System Output. The User Interface (UI) makes it possible to access the several system functionalities. The Processor implements the functionalities that make possible the composition and transmission of messages. As a result it is generated an Output that carries the message to the listener.

The IU is essentially composed by three elements, an On-Screen Keyboard, a Speech Synthesizer and an Interface Agent. In the Processor of our system we can highlight five components: Word Processor, Word Predictor, Lexicon, User Profile and the System Controller. All these components were implemented with C++ objects. The system was developed in the Visual C++ environment and with the MFC support.



**Figure 1:** The AACS "Eugénio - o Génio das Palavras"



**Figure 2:** Eugénio architecture

The Controller of the System is responsible for the coordination of all other components. For such, after system initialization, the System Controller starts waiting for the signalling of events from the other components. As a response to the arrival of a message it can develop a series of actions over some objects of the system. These events are signalled through the Windows message mechanism. Figure 2 shows a simplified architecture of the system. For instance, when the Word Processor informs the System Controller that the user has pressed a key this component asks the Word Predictor for a prediction. To accomplish this task the Word Predictor needs the user text from the Word Processor and the language knowledge contained in the language model. When the prediction operation is successful it supplies the word prediction list to the System Controller that asks the On-Screen Keyboard to present the result to the user. The manipulation of the several objects is done through calls to the public member functions.

### 3 Word Prediction

To generate a list of suggested words, the word predictor must have some knowledge about the language. Three different approaches have been applied to language modelling in word prediction systems: (1) Rule-based modelling; (2) Statistical-based modelling; (3) A mixture of the previous two (Boissiere, 2003). The language model of our system uses only statistical-based knowledge. The development of a good rule-based system for the Portuguese Language, or for any other language, is a complex process, very slow and it requires the participation of specialists. These systems are not adaptable to the user's vocabulary and writing style. The use of a less tuned rules system could impose too much restrictions on the candidate words, removing from the suggestion list correct words for the user. The statistical approach allows the automatic processing of large corpora to produce the required language model. Careful corpora selection also enables the generation of domain specific models for specific users: children, lawyers, engineers, etc.

#### 3.1 Prediction Algorithm

Based on the already written text, the prediction algorithm of our system selects a group of candidate words from the Lexicon, which are classified according to the available statistical information about the language. The words with the best score are presented to the user as a suggestion list.

In the first stage the word predictor obtains a group of  $N$  candidate words,  $w_1, w_2, \dots, w_N$ , through the use of a set of restrictions to the lexicon including the prefix of the word the user is writing. When the user hasn't typed any letter from the word the system considers for suggestion all the words that frequently follow the previous word. When it is used as a language teaching tool, the system also allows the introduction of additional restrictions: it is possible to avoid the suggestion of words with certain letters, or sequences of letters that weren't learned yet. By the limiting the words size it is also possible to force the system to propose small words, and so probably less complex for children in an early learning phase.

After the selection of the candidate words the predictor classifies each one using the statistical language model: it attributes a  $c_i$  score to each one of  $N$  candidate words  $w_i$ ,  $(w_1, c_1), (w_2, c_2), \dots, (w_N, c_N)$ . To obtain the score for each

word our system uses several statistical factors of different nature. Each factor contributes with a partial score for the final evaluation of each word. The final score for the word  $w_i$  is the result of a linear combination of a set of partial scores as expressed in Equation 1.

$$c_i = \alpha_1 \times p(w_i) + \alpha_2 \times p(w_i | w_{prev}) + \alpha_3 \times p(psi_i | ps_{prev}) + \alpha_4 \times p_u(w_i) + \alpha_5 \times p_u(w_i | w_{prev}) \quad (1)$$

The use of a linear combination for the final score of each candidate word presents several advantages. Similar approaches are used by (Carlberger, 1997) and (Fazly, 2002). The contribution of each factor is weighted through the set of  $\alpha$  coefficients and, in this way some factors can be granted higher importance than others. Our system allows the definition of different user profiles, each one with its own set of weights for the different classification factors.

### 3.2 Lexicon

To estimate the values of the several probabilities for the Portuguese Language a newspaper corpus, composed by 156.876 sentences taken from the newspaper “O Público”. The selected sentences are the first two paragraphs of each article published in this newspaper from 1991 to 1994. A newspaper corpus like this one does not fully represent a given language, for instance verbal forms in the 1st person are not common, but it uses a writing style more common than other text corpora like books. In our system the lack of coverage of the initial language model is also compensated by the adaptability to the user own usage of the language.

The real frequency or probability of a word in a language is hard to assess. In what follows, when we refer the term probability of a word or word pair we are indeed referring to the an estimate of this value computed on a given corpus.

Through the processing of our corpus we counted, 65.820 different words and 522.792 word bigrams, and respective frequencies. With these values it is possible to calculate the general probabilities of Equation 1. Word probabilities for a specific user are continually updated with system usage. We also checked that most of the word bigrams presented a low number of occurrences. Through the elimination of all of the bigrams with number of occurrences less than 3, we reduced the bigrams number to 80.850, about 15% of the initial value. In the part-of-speech analysis we used 13 word classes (e.g. nc, np, v, adj). The part-of-speech tagging was performed automatically (Ribeiro, Oliveira & Trancoso, 2002). In our system we only use currently non-ambiguous words (32.933) for which we computed the part-of-speech uni-gram and bi-gram probability.

## 4 Interaction with an External Application

As stated earlier, our system’s goal is to provide text input to the widest possible range of applications on the Windows operating system. To support message composition for an application the AACS must handle each symbol selected by the user to the external application, but it must also be able to preserve contextual information for the prediction process. In some situations it is also necessary that the external application notifies the occurrence of certain events. For instance, an AACS may need to know about the introduction of a new character in the text to do a new word prediction. Some applications already have mechanisms through which they can expose their functionality and data. Ms Word is one of these applications and it makes available the access to their internal objects through OLE Automation. Given the widespread use of this processor in schools and rehabilitation institutions we decided to offer some privilege to the interaction between our system and this application. So it was developed a specific interaction mechanism for Ms Word. We also developed a more generic mechanism that allows the interaction between our system and any other application of the Windows operating system. This mechanism is however more limited than the first.

### 4.1 Interaction with Ms Word

Ms Word has a programming interface, based on OLE Automation, that allows the access to the internal objects. In Ms Word the exposed objects are organized in an hierarchical structure named Word Object Model (WOM). In this

structure the *Application* object is in the top of this hierarchy, being followed the *Documents*, *Paragraphs*, *Words*, *Characters* and other. To access this structure the AACS has to establish a connection with Ms Word through the OLE Automation. The Ms Word OLE Automation server should accept the connection and send to the AACS a reference to the *Application* object, through which it can be made the access to the remaining elements. This way the AACS can access the text the user is writing and it can manipulate it for reading or writing purposes.

In this interaction process was however a difficulty related with the events signalling. The first version of Ms Word with which our system worked (Ms Word 97) signalled to the outside a small amount of events and none of these was a text change signal. Therefore it was needed the creation of a hook to monitor the text input in Ms Word. A hook is a Ms Windows mechanism that allows an application to inspect the arrival of certain messages in another application. In our case the hook monitored the arrival of keyboard messages in the Ms Word. At that time the hook called a function that sent a message to the AACS signalling the event. Knowing the occurrence of this event the AACS accesses WOM to get the already written chars, makes the prediction and finally shows predicted words list. If the user wants to complete a word he or she should press the associated key (e.g. F1, F2, F3,...F9). If the AACS knows about the selection of one of these keys it accedes WOM again and completes the word that the user asked for.

## 4.2 Interaction with a Generic Application

As the generic application does not allow direct access to the text the user has already entered, this information has to be preserved inside the AACS. By inspecting the messages generated by the keyboard, the system maintains a copy of the data necessary for word prediction. The keyboard messages are detected by the hook and sent to the word predictor to update its state in accordance. Cursor movement messages associate with large contextual changes (cursor up or down, page up or down, etc.) clears the prediction memory to avoid out of context suggestions.

When a suggested word is accepted by the user, a sequence of keyboard events is sent to the to the application as if the user has typed it. For the process to be completely invisible to the external application, for each word letter a sequence of messages *WM\_KEYDOWN* and *WM\_KEYUP* is sent stating that a key was pressed and then released. This way the presence of the word predictor is hidden from the external application. Lieberman (1998) calls this approach "marionette strings" because the AACS is given a set of "strings" corresponding one-to-one with user actions in the interface, and can "tug" on the strings to make the program perform.

## 5 User Interaction

The IU is essentially composed by three elements, an On-Screen Keyboard, a Speech Synthesizer and an Interface Agent. The On-Screen Keyboard can be used by people with difficulties in using a computer keyboard. The elements present on this component can be selected with a pointer device (Direct Selection). For persons with difficulties in interacting with these devices the system can automatically scan the matrix items representing the keys awaiting the pressure of a switch as user sign for the selection of the current item.

When required, the Speech Synthesizer can send speech feedback to the user. This audio feedback is extremely important for letters and predicted words selection, abbreviation expansion and even text revision. When the system is being used in a conversation, it will be the element responsible for the transmission of the message to the listener. In our system, it was above all necessary the use of a speech synthesizer for the European Portuguese. As the speech synthesizer DIXI+ is SAPI 4 (Speech Application Interface version 4) compatible, we introduced in our system the ability to interact with any SAPI 4 speech synthesizer.

To increase the interaction capabilities of our system an interface agent was integrated. This agent was implemented using the msagent animation package. By using this agent we intended to increase the ability of the system to be used by children for the acquisition of reading and writing capabilities. In the future we intend that this assistant can adapt to the user and become gradually more competent helping the user (Maes, 1994).

## 6 System Evaluation

AAC users take longer times to attain reasonable performance levels working with these systems. Therefore benefits must be measured through long term studies (Newell et. al, 1992). To start developing long-term studies we are

implementing some logging capabilities in our system. The logging format is compatible with the Universal Logging Format purposed by Leshner, Moulton, Rinkus, Higginbotham (2000). Meanwhile we decided to develop a simulator that acts as an ideal user of the system to measure the performance of our word predictor and the influence of each scoring factor used for word ranking in the prediction algorithm. Simulations were carried out with three compilations of texts from three distinct sources: (1) school texts from a young student with special needs; (2) email texts from an adult; (3) articles from the "Diário do Minho" newspaper. Our word predictor presented performance measures similar to systems of other languages (e.g. keystroke savings of 50%)(Higginbotham, 1992)(klund & Novak, 2001). We verified that the introduction of some factors in the prediction process has positively influenced the system efficacy, however changing the weight of each factor did not introduced significant changes to the overall performance measures. Factor weighting and its optimization may deserve further study.

## 7 Conclusions and Future Work

A complete AACCS was developed for the European Portuguese. This system was called "Eugénio – O Génio das Palavras" and the version 2.0 is available in the Internet, in <http://www.l2f.inesc.pt/~lco/eugenio>. The contacts that we have been receiving have been quite gratifying and they indicate us that this system is being used by a large number of people. The quantitative tests carried out indicate that the system performance is comparable with systems for other languages. As "Eugénio" is a modular system we intend to continue improving all their components and building alternative ones. We also plan to make each system component more autonomous through a more careful design of their interfaces.

## 8 Bibliography

- Boissiere, Philippe (2003). An Overview of Existing Writing Assistance Systems. First French-Spanish Workshop on Assistive Technology, Paris.
- Carlberger, J (1997). Design and Implementation of a Probabilistic Word Prediction Program. In Master's Thesis Dept. of Speech, Music and Hearing, KTH, SE-100 44. Stockholm. Sweden
- Cook, Albert M. & Hussey, Susan M. (1995). Assistive Technologies: Principles And Practice. Mosby. Missouri.
- Fazly, Afsaneh (2002). The use of syntax in word completion utilities", MSc thesis, Department of Computer Science, University of Toronto.
- Higginbotham, D. Jeffery (1992). Evaluation of Keystroke Savings across Five Assistive Communication Technologies. AAC Augmentative Alternative Communication.
- Klund, J. & Novak, M. (2001). If Word Prediction Can Help, Which Program Do You Choose? from <http://trace.wisc.edu/docs/wordprediction2001/>
- Leshner, Gregory W., Moulton, Bryan J., Higginbotham, Jefferey & Alsofrom, Brenna (2002). Limits on Human Word Prediction Performance. CSUN 2002, California State University, Northridge.
- Leshner, Gregory W., Moulton, Bryan J., Rinkus, Gerard, Higginbotham, Jeffery (2000). A Universal Logging Format for Augmentative Communication. CSNU 2000.
- Lieberman, Henry (1998). Integrating User Interface Agents with Conventional Applications. Proceedings of the ACM Conference on Intelligent User Interfaces. San Francisco.
- Maes, Patie (1994). Agents that Reduce Work and Information Overload. Communications of the ACM.
- Newell, A. F., Arnott, J. L., Booth, L., Beattie, W., Brophy, B. & Ricketts, I. (1992). "Effect of the "PAL" Word Prediction System on the Quality and Quantity of Text Generation," AAC Augmentative and Alternative Communication, vol. 8, pp. 304-311.
- Ribeiro, R., Oliveira, L., Trancoso I. (2002). Morphosyntactic Disambiguation for TTS Systems. Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002). Las Palmas. Spain.