

## WRITING TOOLS FOR NON-NATIVE USERS OF ENGLISH

S.M.A.Caldeira<sup>1</sup>, M.C.F. De Oliveira<sup>1</sup>, N.M.Fontana<sup>2</sup>,

C.Y.Nacamatsu<sup>1</sup> and O.N.Oliveira Jr.<sup>3</sup>

<sup>1</sup>Departamento de Computação e Estatística, ICMSC, USP

CP 668, 13560 São Carlos, S.P. - Brazil

<sup>2</sup>Departamento de Letras, Universidade de Caxias do Sul

CP 1352, 95070 Caxias do Sul, R.S. - Brazil

<sup>3</sup>Instituto de Física e Química de São Carlos, USP

CP 369, 13560 São Carlos, S.P. - Brazil

### Abstract

*We report on an ongoing project aimed at developing software tools to help non-native users of English in the writing of scientific papers. The environment, named AMADEUS (AMiable Article Development Environment for User Support), is based on an empirical resource which consists in providing linguistic input to the user in the form of expressions and templates stored in a sentences base, and is at an initial exploratory stage, in the sense that we are trying different user-interfaces and environments for developing the tools. One of the versions conceived for the environment is already implemented.*

**Keywords:** writing tools, non-native users of English, computer assisted writing

### 1) Introduction

The development of software writing aids requires research efforts on a number of different issues, as writers' needs may vary tremendously depending on their experience in writing, on their expertise on the subject they are writing about, and obviously on their knowledge of the language itself. Though some characteristics such as user-friendliness are always desirable, these tools must

generally be targeted at specific user-groups [1].

We have instituted a project aimed at developing software tools that may help non-native users of English in writing scientific papers. The work is essentially based on an (empirical) resource [2] which consists in providing linguistic input to the user in the form of expressions and templates stored in a sentences base. The work reported here may be considered as exploratory in the sense that we are trying different interfaces and environments for developing the tools. In this paper we address the efficacy of the resource proposed, which was analysed through the monitoring of writing exercises by Brazilian post-graduate students (Section II); and the implementation of the resource in software tools (Section III).

## II) Analysis of the Resource from a Linguistic Perspective

The difficulties faced by writers when preparing texts in a foreign language seem to arise due to at least two main factors. The first one is sociolinguistic diversity, as languages differ considerably with respect to phonology, syntax and lexicon, as well as at the pragmatic level [3]. Consequently, different languages use different rhetorical patterns and elements to organize discourse [4]. The second factor which is likely to affect the learner's written outcome has to do with the relationship that holds between comprehension and production: it is now largely accepted that comprehension is beyond production. The combination of these two factors appears to be connected with a phenomenon known as transfer in the linguistic area, whose influence on L2 data has been investigated by several linguists. Here we follow Schachter [5] in her cognitive approach to language transfer. As transfer can often lead to transfer errors, it is our belief that

its negative effects could be prevented, to a certain extent, by providing the non-native writer with expressions from authentic texts from books and papers written by competent native speakers of English to help him/her in organizing their texts more effectively and naturally.

In order to test the efficacy of the resource proposed and also to provide feedback to the process of developing the software tools we monitored writing exercises done by post-graduate students. This was carried out as part of a course on technical writing held in the Physics Department of University of São Paulo in São Carlos. The results are extremely encouraging for using the resource minimized effects of L1 (Portuguese) transfer on L2 (English), and also enabled students to get started in their writing tasks. Indeed, getting over the writer's block has been mentioned as one of the most important problems faced by writers [1].

### III) The *AMADEUS* Environment

*AMADEUS* will comprise composition tools to assist users in the writing of technical articles in English by offering linguistic input in the form of contextualized expressions. It is being developed within the XView (X Window-system-based Visual/Integrated Environment for Workstations) a user-interface toolkit to support interactive, graphics-based applications running under the X-Window system in a Sun Workstation. XView features an object-oriented style interface following the OPENLOOK Graphical User-Interface (GUI) specification. We are following a prototyping approach in the development of the environment, and the X-Windows and Emacs-based tools to be discussed shall provide starting points for further developments.

The expressions from the sentences base may be accessed by one of three ways:

1. **Taxonomy** - The expressions are indexed according to a taxonomy which is based on the main topics likely to appear in a paper, and are accessed by selecting an item in the taxonomy.

2. **Communicative Goals** - The search can be performed by selecting a communicative goal (describe, justify, compare, contrast, etc.).

3. **Keywords** - Here the search is carried out by selecting a keyword that appears frequently in texts of a given area. It can also be used for linguistic items which are usually troublesome. For instance, if the phrasal verbs *carry out* and *carry on* are chosen to be keywords, a search into the database will show the correct contextualized uses for these verbs.

Users can be classified according to two main axes, namely their background knowledge of the English language and their experience in writing papers. In order to 'meet different users' needs we have conceived three versions for the writing environment:

i) **Tutorial** - The whole writing process is monitored by the system; a rigid frame for the organization of the paper is offered, hints are given as to what type of material should be included, and then there is an automatic selection of expressions which guides the user throughout the writing exercise. This automatic selection is driven using expert rules. No acquisition mode for entering new expressions is made available so as to avoid inexperienced writers adding inadequate material. This version is the most demanding in terms of computational development, and has not been implemented yet.

ii) **Support** - This version is similar to the Tutorial except that it offers hints as to how better structure the paper rather than

offering a rigid frame, and allows for acquisition of new material to be entered into the sentences base (customization).

iii) Reference - This version requires the least computational development effort and is already implemented. The user consults the database (accessing the expressions by one of the 3 possible ways mentioned above) and selects the expressions on his/her own. The tool can be customized since it includes the Acquisition Mode by which new expressions can be entered into the sentences base.

Fig. 1 shows a screendump displaying the main features supported by the already implemented Reference version of the environment. Two windows are provided, one as a working area while the other displays expressions from the sentences base. Clicking the option Main Menu the user is returned to the menu which displays the options Composition and Acquisition. The options Article and Keywords correspond to the modes 1 and 3, respectively, for accessing the expressions. The Communicative Goals menu is always displayed on the screen because it is the most general mode of accessing the expressions.

An alternative way of implementing the resource is also being attempted which uses the GNU Emacs text editor, a public domain software package. Emacs was chosen because it provides hypertext facilities, it is customizable and furthermore, it is extensible which means that one can go beyond customization by writing entirely new commands, programs in the Lisp language to be run by the Lisp interpreter provided by Emacs. In our implementation use is being made of a special Emacs function called Outline-mode, which has built-in routines that hide and display chunks of text which can be classified hierarchically as items and sub-items of a menu or as normal text. The two windows available in Emacs are used for the text being edited and the database of expressions, and new

routines were added to the Outline-mode in order to allow the transfer of expressions from the sentences base to the text. The facilities already available are similar to the Reference Version of the tool implemented in the XView environment. Using this tool requires users to be familiar with Emacs, by no means a simple word processor to use. However, we are developing a friendlier interface which will allow users to employ the tool with a minimum background knowledge of how Emacs works.

What we had in mind when we decided to investigate an alternative implementation for the resource was to allow for customization and portability of the tool, and also investigate the possibility of introducing some degree of "intelligence" into it. Emacs seemed a natural choice for its flexibility. For instance, texts in Emacs can be generated in the Text Mode (ASCII) or in other modes so as to allow integration with wordprocessors such as TEX. Because Emacs is entirely re-programmable new functions can be added either for the introduction of expert rules or for extending the portability of a given tool. The obvious disadvantage of the Emacs alternative is its poor graphics capability (compared to XView), but at the moment it is not possible to predict whether this disadvantage will be overpowered by its other welcome features.

#### IV) Concluding Remarks

We should stress that the tools being developed are not aimed at generating text in the more strict sense of the so-called Natural Language Generation (NLG), for the user rather than the system will be ultimately responsible for the quality of the text. But we can, nevertheless, seek help in the techniques usually employed in NLG. In this context, systems [6,7] appeared recently

in the NLG literature which utilize essentially the same strategy discussed here. They are based on statistical analysis of large corpora in order to identify standard linguistic constructions in texts of a given field. It may well be, nevertheless, that the studies we are undertaking, related to language acquisition by a non-native user, may also be of some use to the NLG community.

#### References

- [1] N. Williams, "Writers' Problems and Computer Solutions", CALL Volume 2: Computer Assisted Composition, Intellect, Oxford, pp. 5-25, 1990.
- [2] O.N. Oliveira Jr., S.M.A. Caldeira, and N. Fontana, "Chusaurus: A Writing Tool Resource for Non-Native Users of English", in Proceedings of the XI International Conference of The Chilean Computer Science Society, pp. 59-70, 1991.
- [3] N. Wolfson, "Perspectives: Sociolinguistics and TESOL" Newbury House Publishers, N.York, 1989.
- [4] C. Henner-Stanchina, "From reading to writing acts", in Discourse and learning, Riley, P. (ed.), Longman, Burnt Mill, 1985
- [5] J. Schachter, "A New Account of Language Transfer", in Language Transfer and Language Learning, Gass, S. & Selinker, L. (eds.), Newbury House Publishers, Rowley, pp. 98-111, 1983.
- [6] P.L. Miller and G.D. Rennels, "Prose Generation from Expert System - An Applied Computational Linguistics Approach", AI Magazine, pp. 37-44, Fall 1988.
- [7] F.A. Smadja and K.R. McKeown, Automatically Extracting and Representing Collocations for Language Generation, 28th ACL Meeting, Pittsburg, PA, 1990

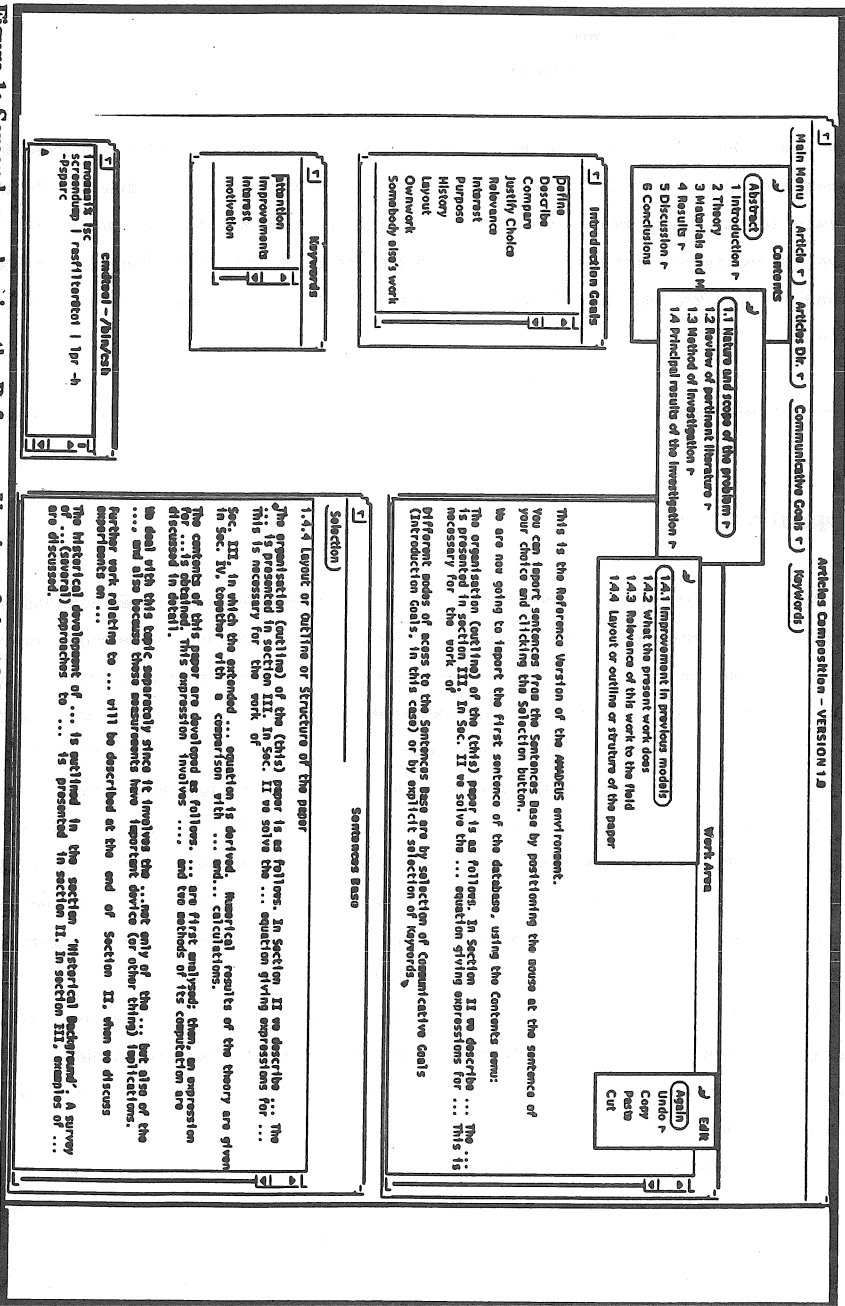


Figure 1: Screenshot showing the Reference Version of the AMADEUS software environment.