Communicating is not easy!
Can the computer help second language learners and teachers?
Ruddy Lelouche
Département d’Informatique, UNIVERSITÉ LAVAL, Québec G1K 7P4, CANADA
Telephone: (+1-418) 656 2131 (2597) Fax: (+1-418) 656 2324 E-mail: Ruddy.Lelouche@ift.ulaval.ca

Abstract
Second language learners have a great difficulty in apprehending usage rules that are used by native speakers to express their intentions or communicative goals in a way appropriate to the situation at hand. In response to this difficulty, we have built a computer program to provide them with the relevant knowledge. The main goal of this paper is to show how our system prototype generates forms appropriate to a given situation, how it analyses submitted linguistic forms, and how the interfaces provided by our prototype, although quite simple, can be used for learning, by teachers and by students alike.

Keywords: Communicative intention, computer-user interface, pragmatics, language teaching, language learning.

Introduction
Because second language (L2) learners must acquire a communicative competence [2], L2 teachers place more and more emphasis on the importance and variety of sociolinguistic factors affecting message and form [1]. However, learners do not find always easy to take these factors into account as they respond to and express themselves in a second language.

Our goal is to use a computer to address this difficulty. More specifically, we want our computer program to provide learners with the necessary knowledge to select the language form(s) considered to be the most appropriate vehicle for receptive and expressive use, and to fulfill a language function in a given communicative situation.

We have shown why identifying usage rules used by native speakers is difficult for L2 learners and how our system addresses that problem [4]. Here we briefly discuss how our prototype generates forms appropriate to a given situation (section 1) and analyses user-submitted linguistic forms (section 2). Finally, we show how the corresponding computer–user interfaces can be used also for learning, by both students and teachers (section 3).

1 Production generation
In the generation case, the problem is the following. Given a particular communicative situation and a particular communication intention, one (i.e. the student, or the computer) is to produce one or several appropriate linguistic forms. We call Réaliseur the system module responsible for generating linguistic forms on the basis of the provided exchange style. About the general 3-engine architecture of PILÉFACE, see [3].

Our prototype Réaliseur interface shown in figure 1 displays the five “linguistic” variables making up the present exchange style (“style d’échange”) and their possible values. In this example, we assume that the communicative situation led to an exchange style in which (checked values): the level is “respectueux” (respectful) and/or “poli” (polite); the personalization index is “impersonnel” (impersonal); the tonality is “décorticée” (relaxed); the insistence is “normale” (normal) and which implies using “vous” (the polite addressing form).

Using this information, the Réaliseur then generates the possible productions “Bonjour, monsieur!” and “Bonjour”, which are rather conservative forms, because of the exchange style of the situation.

2 Production analysis
In the analysis case, the system must be able to analyse a given linguistic form (e.g. submitted by a language learner) in order to diagnose whether it is an acceptable expression of the given intention in the given situation. We call Analyseur the module that effects this diagnosis, from the exchange style derived from the situation.

In figure 2, the communicative situation is supposed to have led to an exchange style in which (checked values): the level is “poli” (polite), the personalization index is “affectueux” (affectionate), the tonality is “décorticée” (relaxed), the insistence is “normale” (normal) and the addressing style is “tu” (colloquial addressing in French).
The user submitted the greeting realization “Bonjour, mon lapin!” (Hello, my dove!). Giving its diagnosis to comment on this realization, the Analyseur finds inappropriate the use of “lapin”, considered as “familier” (familiar), for an exchange style level defined simply as “poli” (polite).

![Interface screenshot](image)

**Figure 2. — Analysis example (development interface).**

### 3. Using these interfaces for learning

The above examples will hopefully convince the reader of the PILÉFACE capabilities to generate or to diagnose utterances. That refers to the problem-solving part of the domain. But what about learning? Here again, these interfaces do allow learning interactions, and we now show how, first on the teacher’s side (or more generally on the development team’s side), and then on the student’s side.

### 3.1 On the teacher’s side

We should first emphasize that the screen snapshots shown in figures 1 and 2 are not the ones presented to the student. Indeed, they are interfaces presented to the developers to check the functioning of the Réalisateur and of the Analyseur. Similar interfaces exist to demonstrate and check the capabilities of the first two inference engines [3], i.e. the Extraiteur and the Formalisator.

The main interest of these interfaces above is that they deal with only one module, were it the Extraiteur or the Formalisator (not presented here), the Réalisateur or the Analyseur (presented in sections 1 and 2). That allows the developers to concentrate on one module at a time, i.e. on a specific inference engine of the system. In each case, the interface allows the developers (good French speakers and language teachers) to experiment with the input parameters of the module being tested. One of the most promising ways is to use a learning-by-experimenting approach: the experimenter is varying one input parameter at a time, and then observes the resulting changes in the output parameters of that module. If there is a discrepancy between the displayed output value and what a normal behaviour should yield, the implied parameter directs the user at once to the erroneous or missing rule.

### 3.2 On the student side

Naturally, in the *interfaces presented to the learner*, these intermediate parameters should be ignored, since there is no certainty that PILÉFACE actually mimics faithfully what happens in the native speaker’s mind! Figure 3 shows a possible student-system interface for an ideal system (the one we are striving at!) where, in addition to the analysis of the situation and of the student’s answer, problems like adaptation to the student’s level and profile would eventually be dealt with in a user-friendly manner: The lack of space forbids us to extensively comment on that figure, and we refer the reader to [4] for details.

![Interface screenshot](image)

**Figure 3. — Student–PILÉFACE interface example.**

### Conclusion

We gave examples of generation and of analysis of linguistic forms, as they are performed by our prototype, and examples of system–learner interactions, as we would like them to be performed by a real tutoring system. We also showed that the corresponding interfaces can also be used for learning by the teachers and the students likewise. The computer, owing to its patience and its permanent availability, can be a good learning aid indeed!

Naturally, this approach needs to be further verified by extending it to other communication intentions. Such intentions will be more interesting to work with, and will give us chances to check the generality of our approach.

### References


