Gouvernement du Canada
Ministère des Communications

Government of Canada
Department of Communications

Le Centre canadien de recherche sur l'informatisation du travail
Canadian Workplace Automation Research Centre

IRMA

An Agricultural Market Report Interpreter

Jean-Luc Cochard
Pierre Isabelle
Michel Simard

Canada

IRMA

AN AGRICULTURAL MARKET REPORT INTERPRETER

Jean-Luc Cochard
Pierre Isabelle
Michel Simard

Computer-Assisted Translation Group

Canadian Workplace Automation
Research Centre
Department of Communications

Laval

May 1990

Cat. No.   Co 28-1/49-1990 E
ISBN   0-662-17888-2

The authors are solely responsible for the opinions expressed in this report.

*   Ce rapport est aussi disponible en français.

Acknowledgements

We wish to express our thanks to:

CWARC, which made this project possible through its funding.

Raymond Descout, head of the Speech Technologies Group, who put forward the idea for the IRMA project and personally assisted in its organization and relations with Expotec.

Russ Short, for his help in programming the IRMA graphic interface.

Elliott Macklovitch, for his help in preparing the script and his valuable comments when this report was written.

Pierre Hamel, François Brochu, Richard Cheng and Philippe Galipeau, for designing the machine communication software.

Michel Beaudet and Pierre Brazeau, for their persistent efforts in solving hardware and software problems.

Marc Dymetman, for reading this report and providing his comments on it.

All the other members of the Computer-Assisted Translation Group, for their comments and suggestions during the different IRMA development phases.

All the demonstrators at the IRMA kiosk and Expotec authorities, for their belief in this project.

And, finally, Sun Canada, Quintus-Prolog and Zenith Canada, who kindly provided the hardware and software essential for this project.

Table of contents

Abstract

IRMA is a computer system that reproduces the skills of a human interpreter. It can recognize speech dictated into a microphone, translate that text from English into French or vice versa, and, finally, provide an oral version of the translation.

That entire process is made possible through the combination of commercial *speech recognition* and *speech synthesis* systems, and an experimental system for the *computer-assisted translation* component – the CRITTER system.

The parallels between IRMA and a human interpreter do not extend beyond a certain similarity of functions performed by the two. IRMA's interest lies mainly in its ability to illustrate, through a highly sophisticated interface, the operation of its internal mechanisms.

This report gives a detailed description of each of the system components and the principles underlying the design of the user interface.

Fields of interest: computer-assisted translation, speech recognition, speech synthesis, graphic interfaces, Prolog.

Introduction

1

The computer system discussed in this report is called IRMA, which stands for INTERPRÈTE DE RAPPORTS DE MARCHÉS AGRICOLES, or Agricultural Market Reports Interpreter, in English. "Interpreter", here, refers to someone who translates between two languages simultaneously. Just like a human interpreter, the IRMA computer system recognizes sentences spoken in French or English, translates them into the other official language and puts that translation into speech.

The comparison between IRMA and the human interpreter, however, does not extend beyond that functional equivalence. IRMA is no more than an experimental system, which cannot compete with humans in the field of interpretation. Each of the links that make it up is much weaker than the equivalent human skills. IRMA simply illustrates the current possibilities in certain fields of artificial intelligence: speech recognition and synthesis, and computer-assisted translation (CAT).

Figure 1 illustrates the IRMA system. On the left of the figure, the microphone captures vocal input and symbolizes the speech recognition system. On the right, the loudspeaker, symbolizing speech synthesis, issues sounds. Between the two, receiving the data from the microphone and transmitting results to the loudspeaker, is the CRITTER computer-assisted translation system, represented as a "black box".

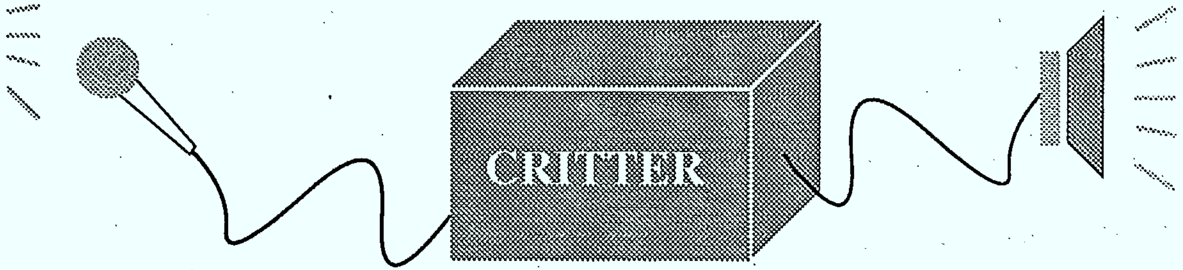


Figure 1 The IRMA computer system.

1.1. Background of the IRMA project

IRMA was the brainchild of joint efforts by the Computer-Assisted Translation (CAT) group, part of the Advanced Technology Directorate (ATD), and the Speech Technologies Group (STG), part of the Integrated Systems Directorate (ISD), both groups at the Canadian Workplace Automation Research Centre (CWARC).

The catalyst for the design of IRMA was a proposal to take part in Expotec 89[1], a summer exhibition on the theme of science for the layman, held at the Old Port of Montreal. The theme for 1989 was "Zoom of the senses!", and the exhibition presented a panorama of technologies and scientific phenomena relating to sight, hearing and speech.

The collaboration between the STG and CAT groups had two facets: the former provided skills in the fields of speech recognition and synthesis, and in the design of communications routines and speech peripherals management, while the latter offered the CRITTER system and the development of a user interface for the IRMA software.

Just slightly more than two months elapsed between the decision to participate in Expotec 89 and the first official and complete demonstration of the workstation, during which the various speech peripherals had to be tested, demonstration scripts written, the user interface designed and implemented, the various hardware components linked together, the different communication protocols defined and installed, and the CRITTER system modified to meet certain special needs of the IRMA project.

1.2. Didactic objective and context of use

Our main aim in designing this application was to make science accessible to everyone. IRMA was intended as a didactic tool to show the difficulties inherent in the automation of translation. Illustrating that point is all the more difficult in that the average spectator does not understand just how complex and approximative a task that can be. Human translation, for him or her, is seen as an atomic operation, indivisible, and the layman has trouble conceiving of the amount of knowledge necessary.

The problem is much the same where speech recognition and synthesis are concerned, two fields in which human beings excel and in which computers are, comparatively, still at the level of baby talk. The real-life application of this type of peripherals in IRMA gave spectators the chance to assess the performance of present-day technology in those fields. The speech recognition system used in IRMA, in particular, is not at the leading edge of technology, but it is close enough. Nevertheless, the operation of the speech recognition and synthesis peripherals is highly spectacular, and had the advantage of capturing – and holding – spectators' attention during the more technical and demanding explanations on CAT.

Making science accessible to everyone is said to sharpen the critical sense of the audience, but it is not clear that this goal could be achieved in an exhibition like Expotec. The number of kiosks to visit and the average time spent at each one, closer to one minute than five, was not ideal for in-depth reflection on a subject as technical as CAT. Nonetheless, we are convinced that the IRMA exhibit was sufficiently informative to answer most of the questions of anyone who took the time to examine it more closely.

1.3. Computer-assisted translation and the CRITTER system

Translating documents was one of the first non-numeric applications suggested for computers following their introduction in the late 1940's. Why, then, after 40 years of enormous investment, is high-quality computer-assisted translation still an inaccessible

[1] Presentation by Lavalin Communications and the Old Port of Montreal

goal? The reason is that simple word-for-word substitution is insufficient to produce a good translation. Computers, like human translators, have to understand perfectly the text to be translated. Our attitude toward this problem can be summed up by saying that computer-assisted translation can produce quality results only when applied to very restricted fields. It is for that reason that IRMA translates only Agricultural Market Reports, as thorough knowledge of that field has been programmed into the system.

As Figure 1 shows, the translation phase per se is handled by CRITTER, a computer-assisted translation system under development at CWARC [1]. That software was not designed for use as a translation system for IRMA; rather, IRMA should be seen as a possible application based on the skills of CRITTER. At present, the CRITTER system can automatically translate real-life documents – Agricultural Market Reports issued by Agriculture Canada – from English to French and vice versa.

As a central component of IRMA, the CRITTER system has been slightly modified to deal with a considerable problem for which it was not initially designed: overcoming the ambiguity of heterographic homophones (see Section 3.2). This problem, which surfaces during the use of the speech recognition peripheral, is generally difficult to manage, and CRITTER's ability to handle it satisfactorily was a happy surprise.

1.4. Speech processing

Speech recognition is another task in which the computer attempts to imitate human abilities. The human skill that we are trying to reproduce in this way is deceivingly simple, in that it is simply a matter of identifying the words spoken. The complexity comes with the realization that people automatically make links when they utter a sequence of words, that the intonations of different speakers are necessarily different, that the listening environment is rarely ideal, and so on.

At present, the most common commercial systems can recognize isolated words, clearly articulated by speakers for whom the system is "trained". This is a clear illustration of the limitations of this type of application, and also explains why IRMA's vocabulary is so limited. Every system user must undergo training, during which he or she must very carefully repeat every word in IRMA's vocabulary, four or five times.

At first glance, speech synthesis poses fewer problems than speech recognition, and hence is much less restricting. The fact remains, however, that if a "human" quality of elocution is to be attained, speech synthesis becomes a very complex operation. IRMA uses two speech synthesis systems, one for English and one for French, but the two are unfortunately not comparable in terms of elocution skill.

1.5. User interface

All these components of the IRMA system, i.e. the CRITTER system and the speech peripherals, are connected through a *user interface*. In addition to managing the flow of data among the various components, that interface clearly and simply displays the various functions available in the application. Thus we will be discussing mainly the interface in the following chapters, not in its capacity as a link among the components, but rather because it gives a picture of the product, offers concrete evidence of the designers' aims, and plays an essential role in making the technology accessible to everyone.

Accordingly, the organization of the IRMA console and the available functions are described in the following chapter, *IRMA's capabilities* (Chapter 2). The visual appearance of IRMA's user interface meets a series of criteria and principles that we have already discussed, and which are summarized in Chapter 3, *Design concepts*. And to round off our general look at IRMA, we will then deal with a few highly technical aspects in Chapter 4, *Technical details*, such as the hardware necessary and communications protocols between the interface and peripherals.

IRMA also served as a testing ground for the use of other speech recognition systems besides the version on display at Expotec. Those experiments are summarized in Chapter 5, *IRMA variants*. And to conclude, in Chapter 6 we offer a critical reflection on what could be termed the IRMA experience, in which we attempt to highlight errors and incorrect choices made, but also the positive reactions and repercussions of IRMA on the future work of the Group in the area of graphic interfaces. One concrete benefit is the establishment of a joint project with Bourbeau Pinard, Inc., for the creation of a Microcomputerized Phonetic French Dictionary (DMFP), as an extension of the DMF [2].

To complete this relatively comprehensive overview of the IRMA project, the Appendices include the script distributed to the demonstrators to assist them in explaining the operation of the IRMA workstation, and a series of screen copies from the IRMA console, which illustrate that script and give an idea of how the system operates.

IRMA's capabilities

2

2.1. General

IRMA kiosk

Figure 2 shows the IRMA kiosk, as it appeared throughout the summer at Expotec 89. The different visible IRMA components are as follows:

- a SUN 4/110 workstation, with its keyboard and mouse, to which we refer when discussing *the IRMA console*;
- a micro-headset on the demonstrator's head, for speech recognition;
- a loudspeaker, for speech synthesis; and
- an amplifier, built into the stand.

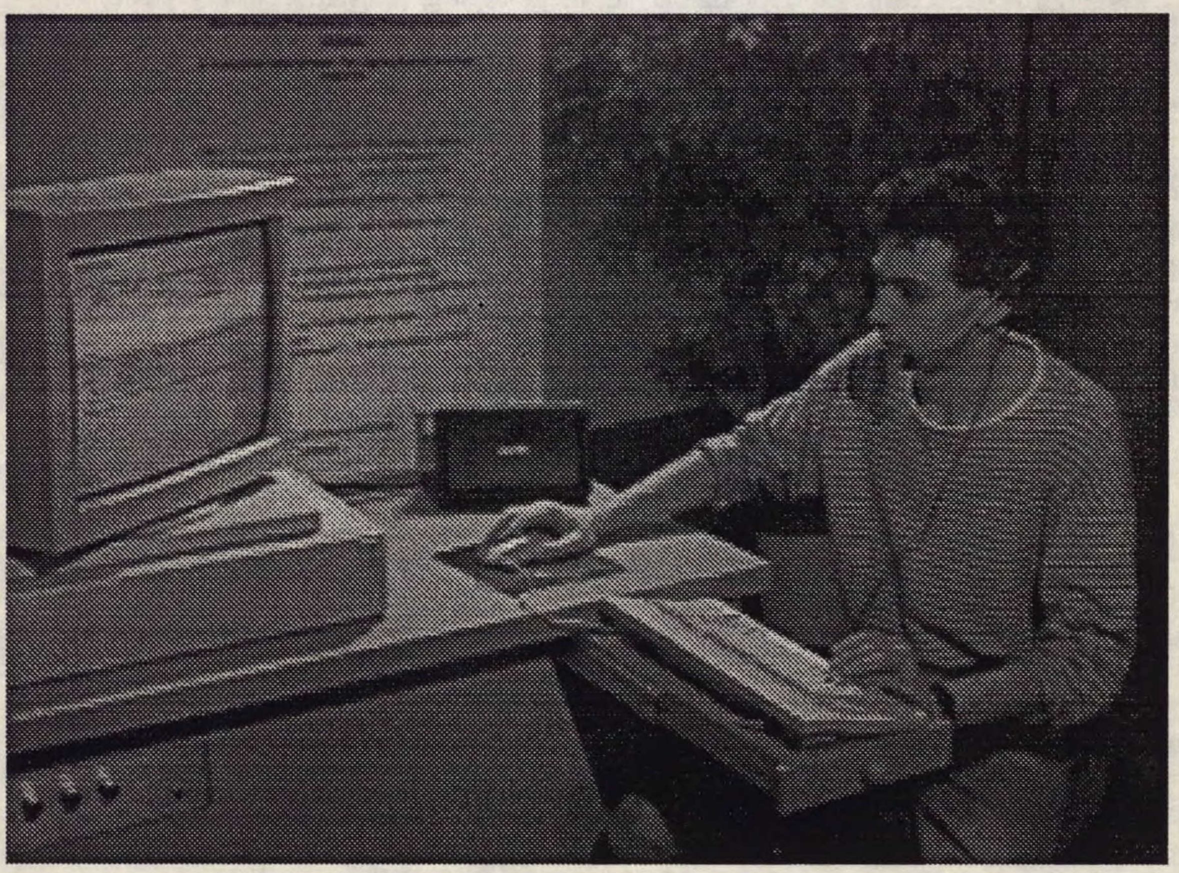


Figure 2 The IRMA kiosk, as it appeared at Expotec 89.

The stand contains a wide range of hardware, whose nature and purpose we will describe in Chapter 4. On the back of the stand is displayed a poster very briefly explaining IRMA's capabilities.

We will devote the rest of this chapter to an examination of the contents of the IRMA console, since it is essentially through this device that the demonstrator interacts with the system.

General procedure

Before examining the capabilities of the interface in detail, we will first present a brief description of the general procedure for using IRMA. The demonstrator dictates sentences into a micro-headphone connected to the speech recognition system, one word at a time, ensuring that every word in the sentence is included in the system's vocabulary. When a word is recognized, it is automatically displayed on the screen of the IRMA console, at which point the demonstrator may dictate the next word.

Once the entire sentence has been recognized, the demonstrator gives the translate command, and may decide whether or not to display the intermediate results of the translation process. When the translation is completed, the system automatically displays a written version of the translated text and, at the same time, provides an oral version.

If the intermediate results are not displayed, the complete translation of one sentence, including speech recognition and synthesis, takes only a few seconds.

Starting up IRMA

When the IRMA system is started up, the console screen looks like Figure 3. The screen is broken up into a number of areas called *windows*, separated by either two thin lines or one thick one. Every window has a specific role, and contains various textual or graphic objects, the latter referred to as *icons*. All the components are discussed in depth in the following sections. For the moment, we will restrict ourselves to a general overview.

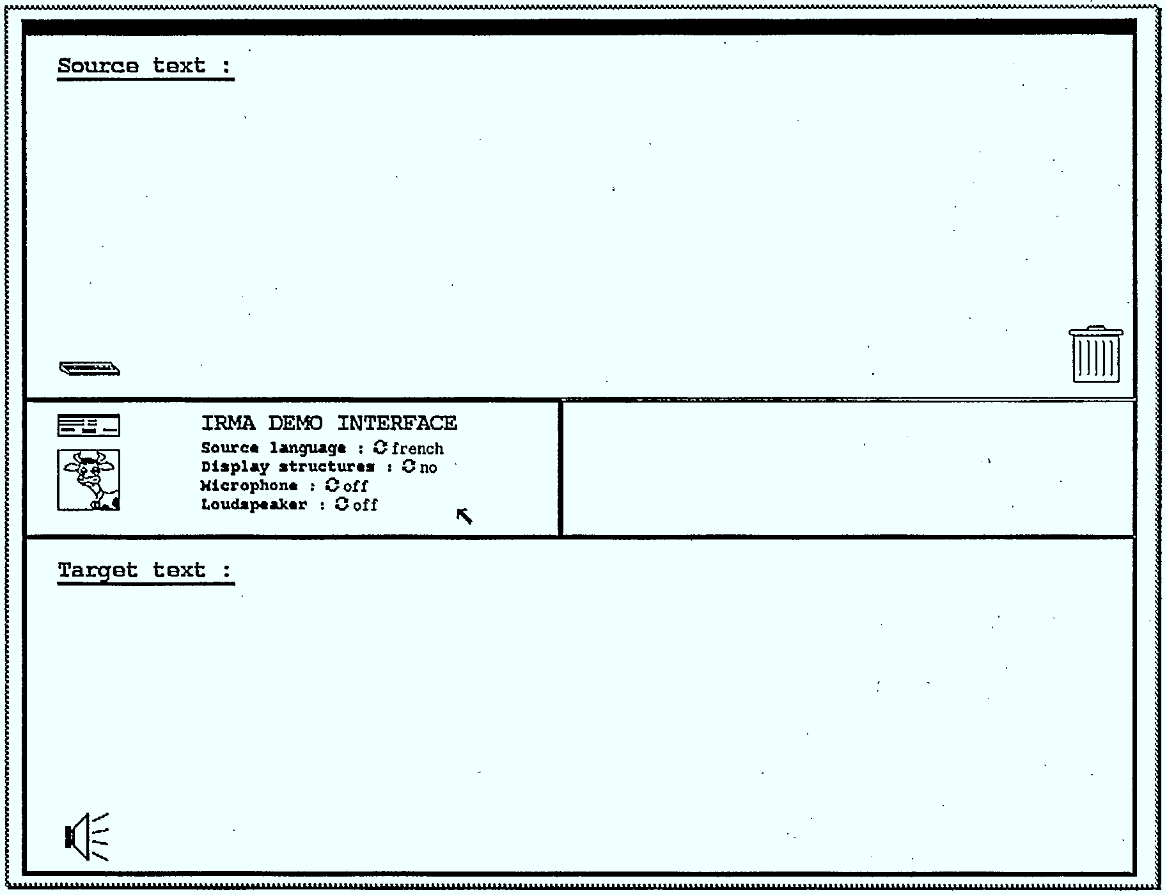
The window in the upper part of the screen containing the source text :[2] and two icons (the keyboard and the garbage can), is called the *speech editor*. It is in that window that the text for translation, which may or may not be dictated into the microphone, appears. The window in the lower part of the screen, containing the target text : and the loudspeaker icon, is used to *display translations*. Between those two windows are two other windows. The one that is initially blank plays a role in the display of *steps in translation*. The other, which reads IRMA DEMO INTERFACE, and contains a number of other symbols, is the central *control panel* of the interface. It is from this window that the user asks for a translation, turns the speech recognition and synthesis peripherals on and off, etc.

Comments

In the specific Canadian linguistic context, i.e. French-English bilingualism, it is important that IRMA be able to display text in French or English. One of the system parameters (see *Configuration parameters*, in Section 2.2) allows the user to select the display language. Note also that this feature called for some extra efforts in programming, with the result that the list of display languages could easily be increased.

[2] This character set is used throughout the report to identify objects on the IRMA console.

The French version of this report uses screen copies from the French-language version of IRMA.



Source text :

IRMA DEMO INTERFACE

Source language : ↻ french
Display structures : ↻ no
Microphone : ↻ off
Loudspeaker : ↻ off

Target text :

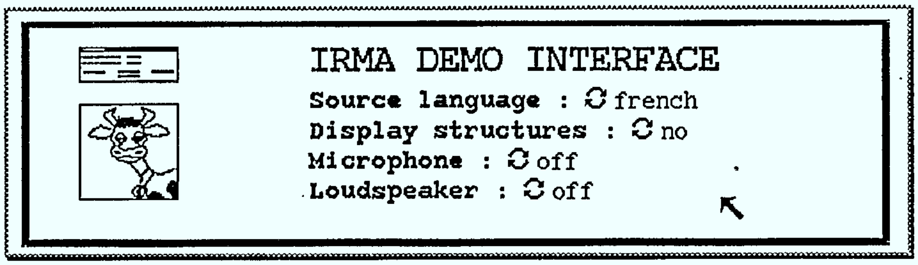
Figure 3 Initial IRMA screen.

2.2. Control panel

IRMA's control panel allows the user to perform three types of operations: toggle the operating parameters, request a translation, or exit the program. The operating parameters can be broken down into two categories: those that it may be desirable to change at any point during a demonstration, and which we will refer to as *operating parameters*, and those that are changed between demonstrations, if necessary, referred to as *configuration parameters*.

Operating parameters

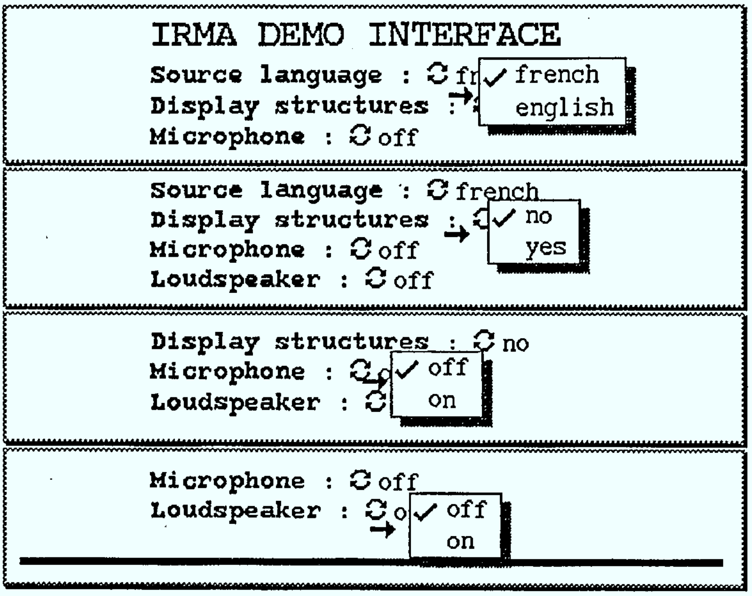
There are four operating parameters that appear directly in the control panel: `Source language`, `Display structures`, `Microphone`, and `Loudspeaker` (see Figure 4).



IRMA DEMO INTERFACE
Source language : ↻ french
Display structures : ↻ no
Microphone : ↻ off
Loudspeaker : ↻ off

Figure 4 IRMA control panel window.

Each of those parameters can be changed in two different ways: either by clicking the left-hand button on the mouse while the cursor is near the current value, or by clicking the right-hand button. In the first case, the current value is replaced by the next value in a circular list of possible values (when the end of the list is reached, the next value is the first value in the list). In the second case, the system displays the list of possible values, highlighting the current value. We decided on that technique in order to show the possible values of the different parameters.



IRMA DEMO INTERFACE
Source language : ↻ fr ✓ french
Display structures : english
Microphone : ↻ off

Source language : ↻ french
Display structures : ✓ no
Microphone : ↻ off yes
Loudspeaker : ↻ off

Display structures : ↻ no
Microphone : ✓ off
Loudspeaker : on

Microphone : ↻ off
Loudspeaker : ↻ o ✓ off
on

Figures 5.1 to 5.4 The different possible operating parameter values.

The Source language parameter (see Figure 5.1) refers to the language in which the text appearing in the speech editor is given. This parameter has an effect on speech recognition, since the system must know what language is being spoken so that it can select the correct vocabulary (see Section 4.2). Obviously, it also influences the translation process.

The Display structures parameter (see Figure 5.2) relates directly to the translate command (see *Translate command*, in Section 2.2). The value no means that the translation will be carried out without interruption, and the structures handled during the process will not be displayed. The value yes causes the translation command to work in such a way that the operation of the CRITTER system (see Section 2.5) is explained as it carries out the translation within IRMA.

The **Microphone** parameter (see Figure 5.3) determines the status of the speech recognition peripheral, i.e. whether it is **on** or **off**. Thus the demonstrator can alternately dictate a sentence for translation and offer explanations, without removing the headphone.

The **Loudspeaker** parameter (see Figure 5.4) determines the status of the speech synthesis peripheral. If the loudspeaker is **on**, the translated text is sent for speech synthesis after being displayed in the appropriate window. If it is **off**, the translated text is simply displayed.

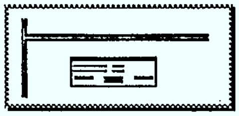
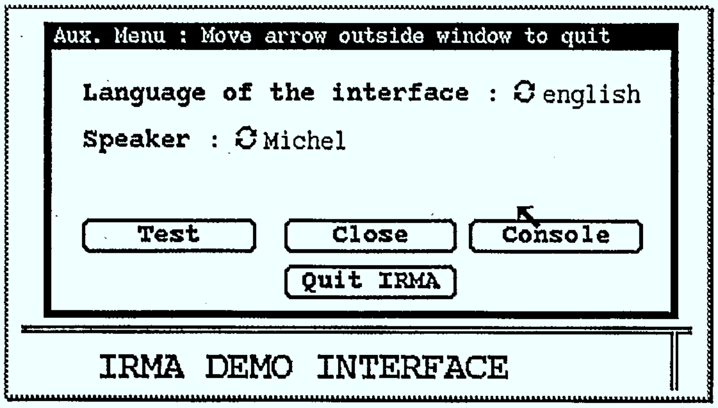


Figure 6 Icon allowing the user to display the configuration parameters.

Configuration parameters

The configuration parameters can be displayed by selecting the icon representing a small window, on the control panel (see Figure 6), with the left-hand mouse button. When the user makes that selection, the system displays a new, temporary, window above the control panel, and the cursor is automatically placed inside that window (see Figure 7). To close that temporary window, the user simply moves the cursor outside it.



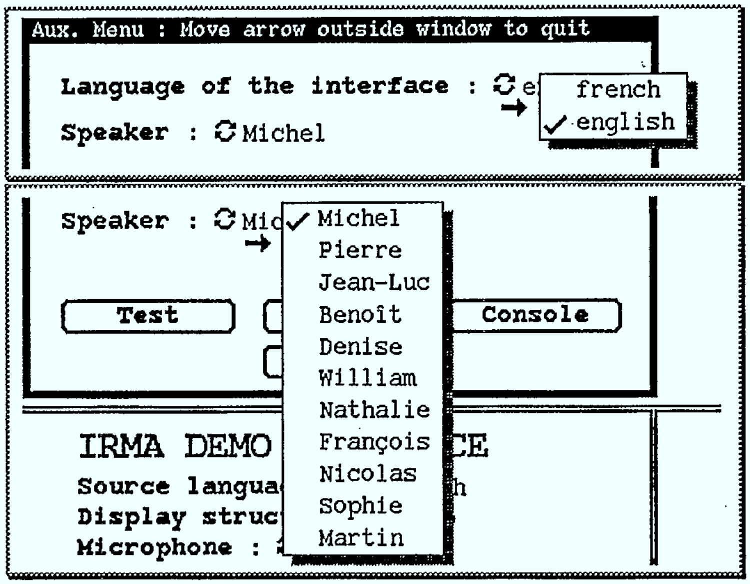
Aux. Menu : Move arrow outside window to quit

Language of the interface : english

Speaker : Michel

Figure 7 Configuration parameters window.

The **Language of the interface** parameter (see Figure 8.1) refers to the language in which the text is displayed in the interface, i.e. the various titles, names and values of parameters and the messages that may appear on the screen during certain operations. It is with this parameter that IRMA can be configured for use by Francophones or Anglophones.



Aux. Menu : Move arrow outside window to quit

Language of the interface : english

Speaker : Michel

Figures 8.1 and 8.2 Different possible values of configuration parameters.

The **Speaker** parameter (see Figure 8.2) allows the user to configure the speech recognition module for optimal performance. Since IRMA uses single-speaker speech recognition software, this parameter must be set correctly.

Translate command

The translation process is activated by selecting the cow icon, with the left-hand mouse button. The contents of the speech editor are then submitted for translation by the CRITTER system.

When the user sends a sentence to the CRITTER system, the cow icon's appearance varies, indicating the stage reached in the translation process. The following is an explanation of the various images.

Initial (default) stage: waiting for a translation request.

Linguistic processing stage: executing a CRITTER system step.

Processing phase in the interface: either the translation or the graphic structures are displayed.

Current operation failed; this icon is displayed for approximately two seconds, and then is replaced by the default icon.

In these picture codes, the cow illustrates the steps in a translation by the CRITTER system by chewing and swallowing a sheaf of hay. If the translation fails, the cow's vexed appearance can be taken to indicate indigestion!

The translate command has two separate effects when the structures are not to be displayed. The first, obvious, effect is to produce a translated sentence. The second is to render the original sentence less ambiguous. The use of a speech recognition peripheral to enter the source text gives rise to spelling ambiguities owing to heterographic homophones. The representation of those ambiguities is described in Section 2.3. A secondary effect of the translation is to select the correct one of the many possibilities displayed. Consequently, the original sentence can be displayed in the speech editor.

Exits

The IRMA system has three "exits", corresponding to three specific situations in which the user may wish to leave IRMA (see Figure 9).

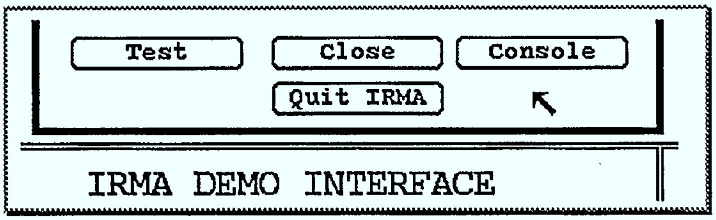


Figure 9 IRMA exit commands window.

The first and simplest situation is when the user wishes to exit IRMA completely. In that case, the `Quit IRMA` button should be used. IRMA will ask for confirmation by displaying a new window (see Figure 10) and allowing the user to reply with `Yes` or `No` to the question `Quit?`

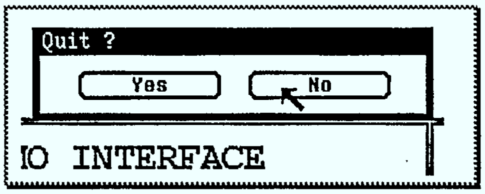


Figure 10 Request for confirmation when the user selects `Quit IRMA`.

IRMA also allows the user to interrupt work in IRMA by freeing the screen up for the display of other windows. This operation is accomplished by selecting the `Close` button, and the IRMA screen is replaced with the cow icon (see Figure 11). To re-open the IRMA screen, the user need only select that icon with the left-hand mouse button.

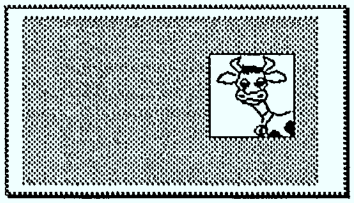


Figure 11 Section of screen after the IRMA application is temporarily closed.

The third possibility, using the `Console` button, is to temporarily halt IRMA by returning control the recognition PC, in order to train the system to recognize new words. This very special operation is discussed in greater detail in Section 4.2. If the user selects this button on the console, IRMA asks for confirmation, displaying a new window (see Figure 12) and requesting a `Yes` or `No` answer to the question `Control goes to the PC. OK?`

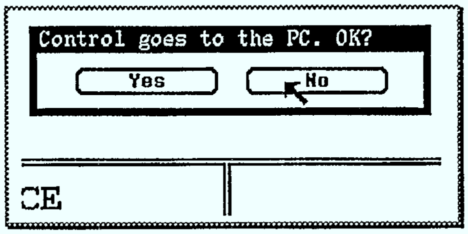

Control goes to the PC. OK?

Figure 12 Request for confirmation of the `Console` command.

The use of the other button, `Test`, is discussed in Section 4.2.

2.3. Speech editor

The speech editor is the name given to the window that displays the text recognized by the speech recognition peripheral (see Figure 13). That text consists of the separate units, i.e. words and punctuation symbols, identified through speech recognition. In addition to displaying the units, the user can also manipulate them, which is why this window is referred to as an editor.

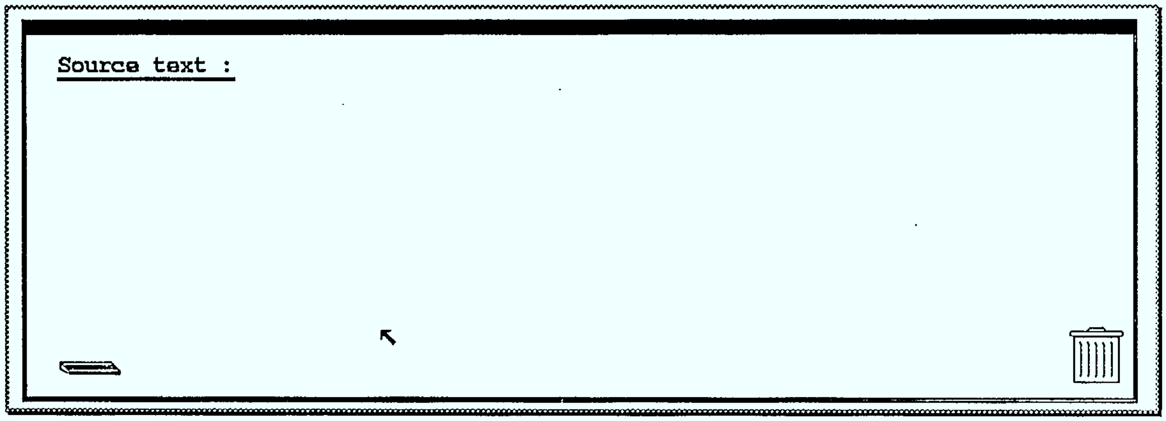
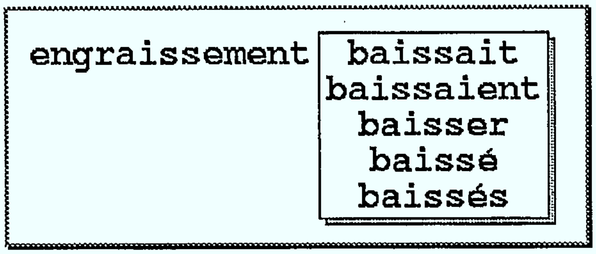


Figure 13 IRMA speech editor.

Types of units displayed

Two types of units can be displayed by the speech editor: words (including punctuation symbols) and heterographic homophones (see *Loss of information between spoken and written language*, in Section 3.2). The words are displayed as usual, with the homophones displayed in a column on a separate background (see Figure 14).



engraissement

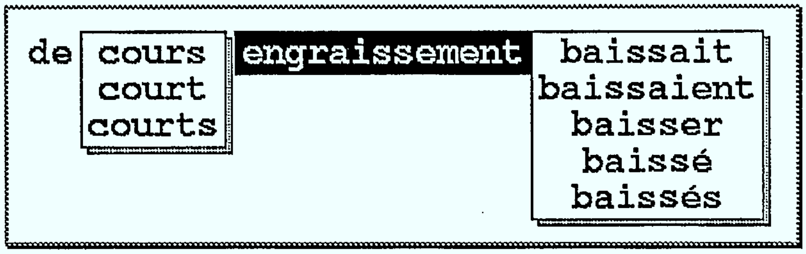
baissait
baissaient
baisser
baissé
baissés

Figure 14 Part of a sentence, illustrating the two types of units that can be displayed.

Manipulating units

Units can be manipulated in three ways in the speech editor: adding a new unit, replacing a unit, and deleting a unit. All those functions are handled with a single elementary operation, called selecting a unit.

A unit is selected by clicking the left-hand button on the mouse, while the cursor is close to or on the unit. The unit is displayed in reverse video to indicate that it has been selected (see Figure 15). To de-select a unit, the user need only select another unit or click the left-hand cursor button while the cursor is "nowhere".



de

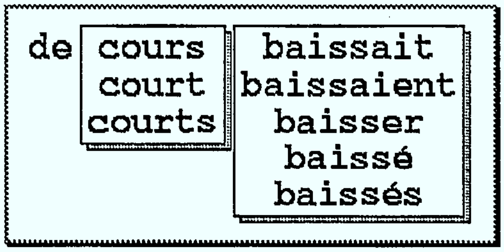
cours
court
courts

engraissement

baissait
baissaient
baisser
baissé
baissés

Figure 15 Display of a selected unit, the word `engraissement`.

If no selection is made, which is the default situation, the dictation of a word is interpreted as adding a unit to the end of the line. If a unit has been selected, the word dictated is considered to replace that unit. If a unit is selected twice in a row, it is deleted (see Figure 16). Those few simple operations are all there is to manipulating units in the speech editor.



de

cours
court
courts

baissait
baissaient
baisser
baissé
baissés

Figure 16 Deleting the word `engraissement` by selecting it twice.

Keyboard and garbage can

The speech editor offers another way, besides dictation, to input units: the keyboard. The keyboard is turned off by default in the speech editor, and so one cannot simply type the words out and watch them appear on the screen. The keyboard must first be turned on, by selecting the corresponding icon (see Figure 17).

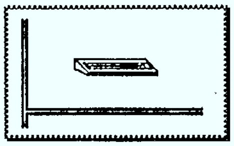
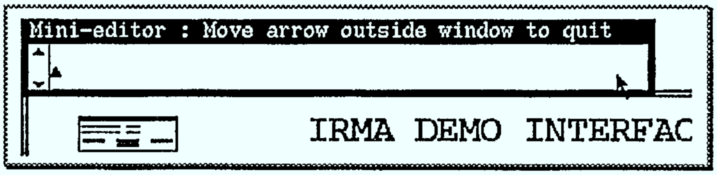


Figure 17 Keyboard icon in the speech editor.

When this icon is selected, it is temporarily hidden by a window displaying the text that is being entered at the keyboard, unit by unit (see Figure 18). The cursor is automatically placed inside that window when it opens, allowing the user to begin typing immediately. When the end of a unit is detected in the sequence of characters entered at the keyboard, i.e. a space between two words or a punctuation symbol, the unit is sent to the speech editor as if it had been input through the speech recognition peripheral. To terminate the keyboard input, the user simply moves the cursor outside the input window.



Mini-editor : Move arrow outside window to quit

IRMA DEMO INTERFAC

Figure 18 Temporary keyboard input window.

Note

When text is input through the keyboard, there is no way to display homophones, since they are specific to speech.

To speed up the IRMA demonstrations and since speech recognition was not used constantly throughout the demonstration, we pre-recorded several sentences. When the **Source language** is **French**, the sentences associated with function keys F1 to F4 are as follows:

F1 les prix augmentaient de 3 dollars lundi, demeurant soutenus pour le reste de la semaine.

F2 les vaches grasses se vendaient à rabais, alors que les meilleures génisses augmentaient de 3 à 4 dollars.

F3 les sujets maigres de un an se vendaient à prix soutenus.

F4 les cours des bovins de court engraissement baissaient de cinq dollars.

When the **Source language** is **English**, the pre-recorded sentences are as follows:

F1 prices increased three dollars Monday, remaining steady for the rest of the week.

F2 fat cows were discounted, while the top heifers were up $3-$4.

F3 lean yearlings sold at steady prices.

F4 feeder prices decreased $5.

When the processing of a sentence is finished and the user wishes to move on to the next sentence, or when the contents of the speech editor are so incorrect that no manipulation is worthwhile, the user can select the garbage can icon (see Figure 19), which clears the speech editor and the translation display window (see Section 2.4).

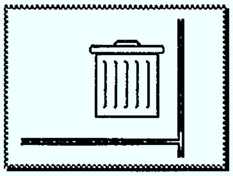
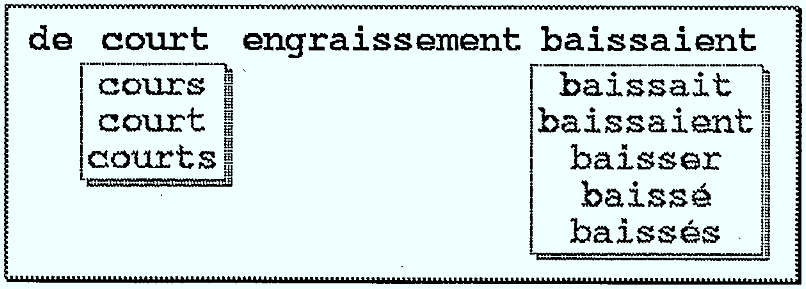


Figure 19 Garbage can icon in the speech editor.

Secondary effect of translation

When IRMA carries out a translation, it must choose between the various groups of homophones that may apply to the source-language sentence. To illustrate those choices, the system replaces the groups of homophones on their separate backgrounds with conventional units, by moving the backgrounds of the homophones on the screen and displaying one of the units that appeared in the previous list (see Figures 15 and 20).

This operation is carried out either before the translation is produced, when the `Display structures` parameter is disabled, or at the end of the analysis when `Display structures` is enabled.



de court engraissement baissaient

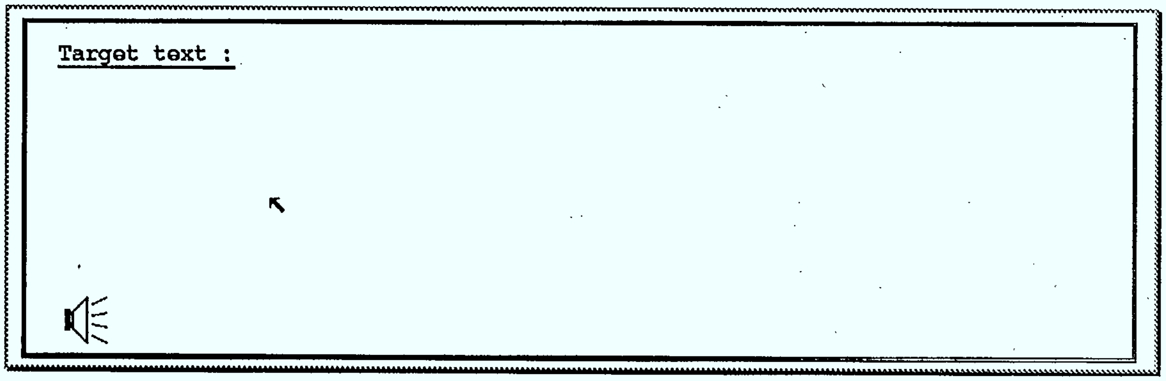
cours
court
courts

baissait
baissaient
baisser
baissé
baissés

Figure 20 Clarification of ambiguities in a sentence fragment.

2.4. Display of translations

The translation display window is used only to display the written version of the translation produced by CRITTER (see Figure 21). Depending on whether the loudspeaker is on or off, the sentence may or may not be sent to the speech synthesizer.



Target text :

Figure 21 Translation display window.

By selecting the loudspeaker icon in this window, the user can have the sentence on the window read out by the speech synthesizer, provided that the `Loudspeaker` is first turned `on`. No other operation can be carried out in this window, since only the garbage can icon in the speech editor can be used to clear the text from this window, and no text manipulation is possible.

2.5. Steps in translation

When `Display structures` is turned on in the control panel, the window on the right of the control panel displays three new icons, containing the initials of the key terms for this type of operation: analysis, transfer and synthesis (see Figure 22).

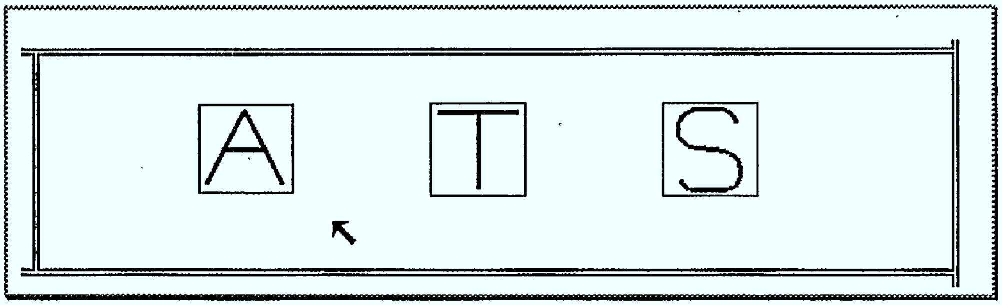


Figure 22 Initial configuration of the icons symbolizing the steps in the translation process.

This IRMA feature illustrates the principles and techniques applied in the design of a CAT system such as CRITTER. The schematic breakdown of the translation process into three steps (see Figure 23) illustrates some general principles:

- The analysis and synthesis operations depend on a single parameter, the language for which they are designed; only transfer is defined for a pair of languages.
- Every step or operation is a program that outputs a result to serve as input for the following step, which allows for totally independent design and operation of the three steps.
- The analysis corresponds to reading and understanding the text for translation. This operation employs a unilingual dictionary and a grammar of the source language. The semantic structure resulting from this step is a formal representation of the meaning of the analysed sentence, which describes the objects in the sentence as well as their implicit or explicit relationships within that sentence.
- The transfer step uses the semantic graph of the analysed sentence to produce an equivalent semantic graph oriented toward the target language. This is the actual translation step, which basically uses a bilingual dictionary of correspondences between semantic structures.
- The synthesis step composes a sentence in the target language, working from the target-language semantic graph. For that purpose the system consults the unilingual dictionary and grammar of the target language so as to place words in the correct order, conjugate verbs, make agreements of gender and number, etc.

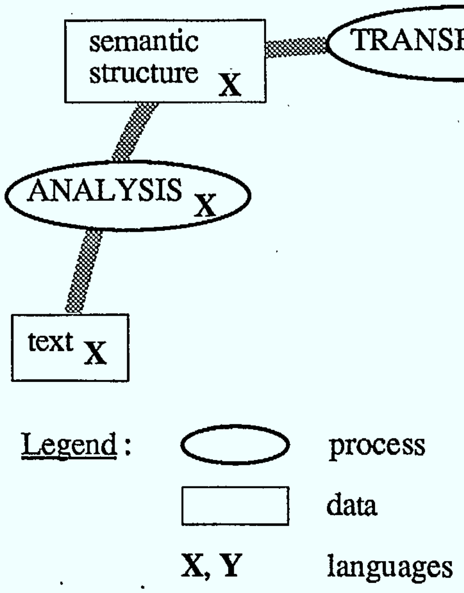


Figure 23 Schematic breakdown of the translation process.

When the translate command is given by selecting the cow icon, and intermediate structures are displayed, the system carries out only one of the three steps, depending on the point it has reached in the process. The appropriate icon is displayed in reverse video to indicate which step is being executed (see Figure 24). Thus it is necessary, in this situation, to give the "translate" command three times.

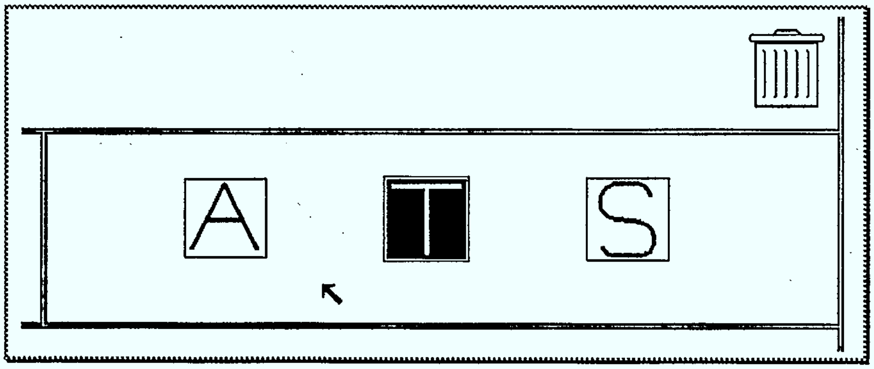
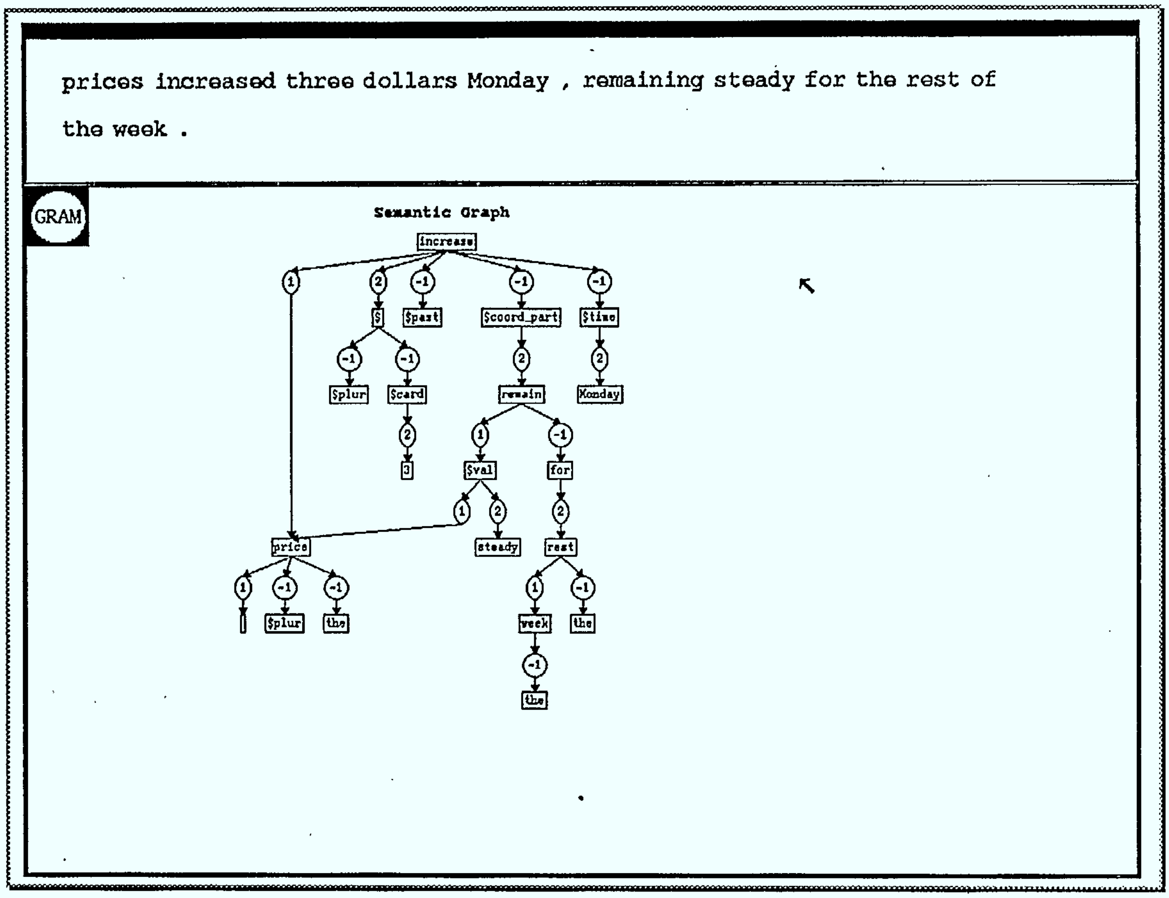


Figure 24 Appearance of the icons while the system is executing a transfer.

To see the structures actually being manipulated during the translation process, the user need only select the icon corresponding to the step desired. For each step, the interface will display a different screen, containing the data and results relating to that step. That operation is illustrated in Figures 25 to 29, as the sentence `les cours de bovins de court engraissement baissaient de cinq dollars` is translated.

When the information concerning each step is displayed by selecting one of the icons (see Figure 24), a new screen is superimposed on the main IRMA screen. There may even be up to three windows on the screen at once, as Figures 28 and 29 show. To close the most recent window and return to the previous one, the operator simply clicks the left-hand mouse button while the cursor is "nowhere" on the screen.

When the **A** icon symbolizing the analysis step is selected (see Figure 25), the text for translation is copied from the speech editor and displayed on the upper part of the screen. The result is a semantic graph (see [1], for explanations of semantic structures). Note that data items appearing on the analysis screen and containing homophones are not subject to the secondary effect mentioned in *Secondary effect of translation*, Section 2.3.



prices increased three dollars Monday , remaining steady for the rest of the week .

Figure 25 Screen illustrating the data and results of the analysis step.

The transfer phase, associated with the T icon (see Figure 26), uses the results of the analysis, in the upper part of the screen, for input data, as illustrated in the diagram in Figure 23. Only the spacing between nodes is different. The result of the transfer, in the lower part of the screen, is another semantic graph, this time in the target language.

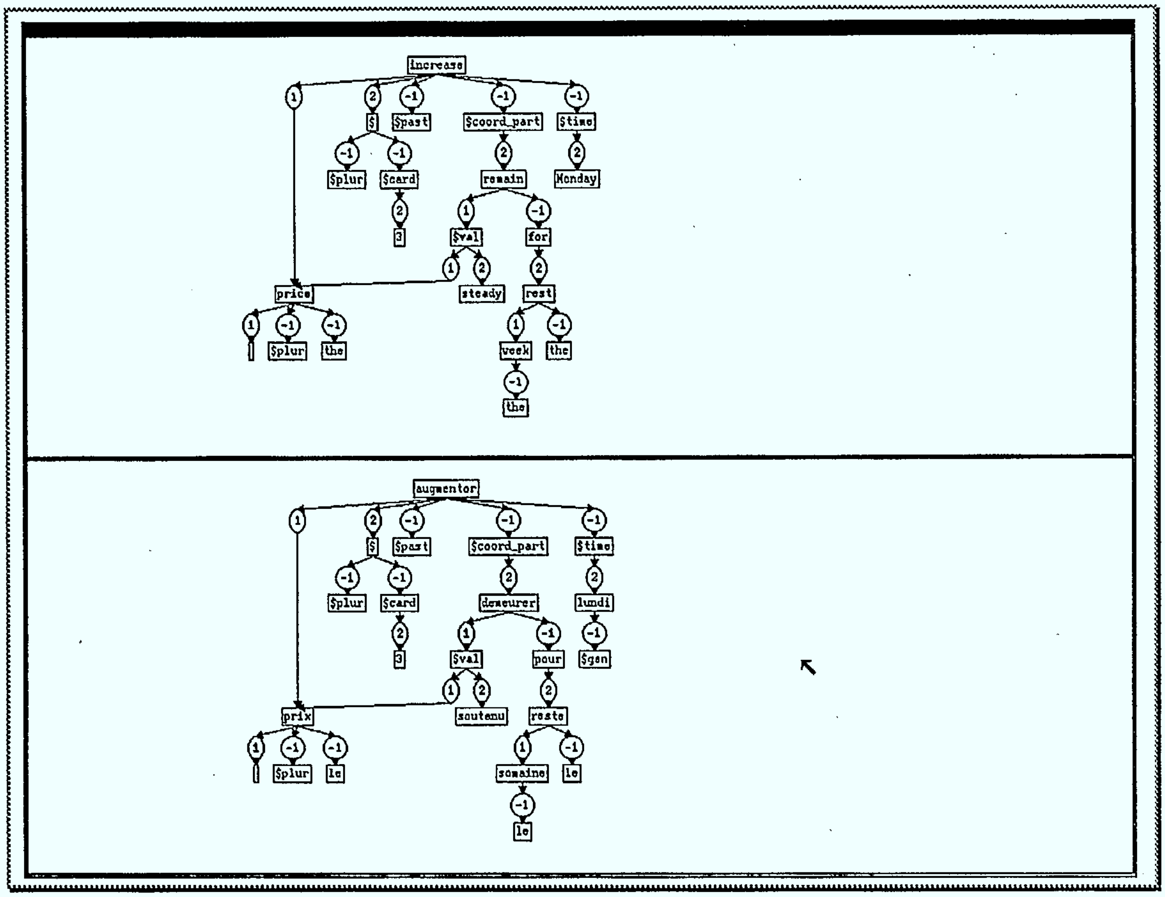
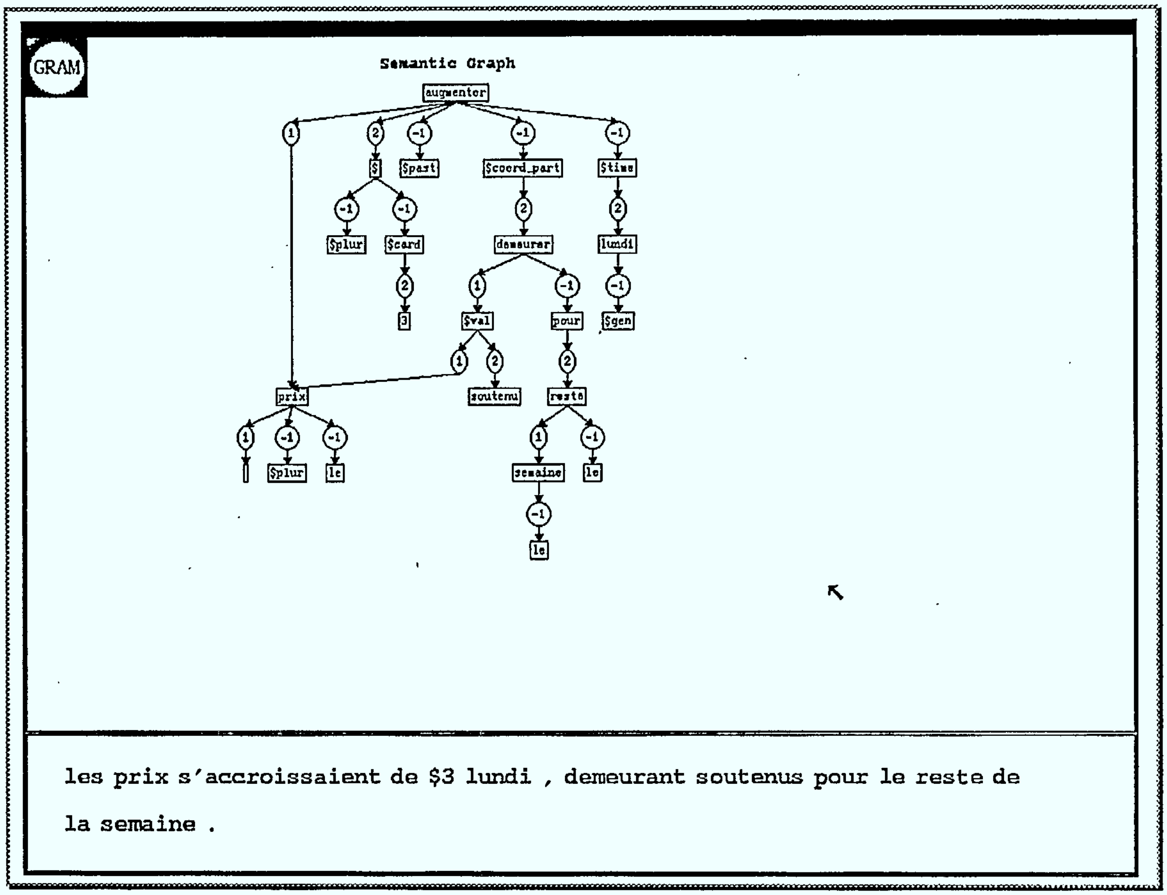


Figure 26 Screen combining the data and results of the transfer step.

The synthesis, associated with the s icon (see Figure 27) uses the results of the transfer, in the upper part of the screen, for input data, as illustrated in Figure 23. The result of the synthesis, in the lower part of the screen, is the translated sentence. This sentence is identical to the one displayed in the translation display window (see Section 2.4).

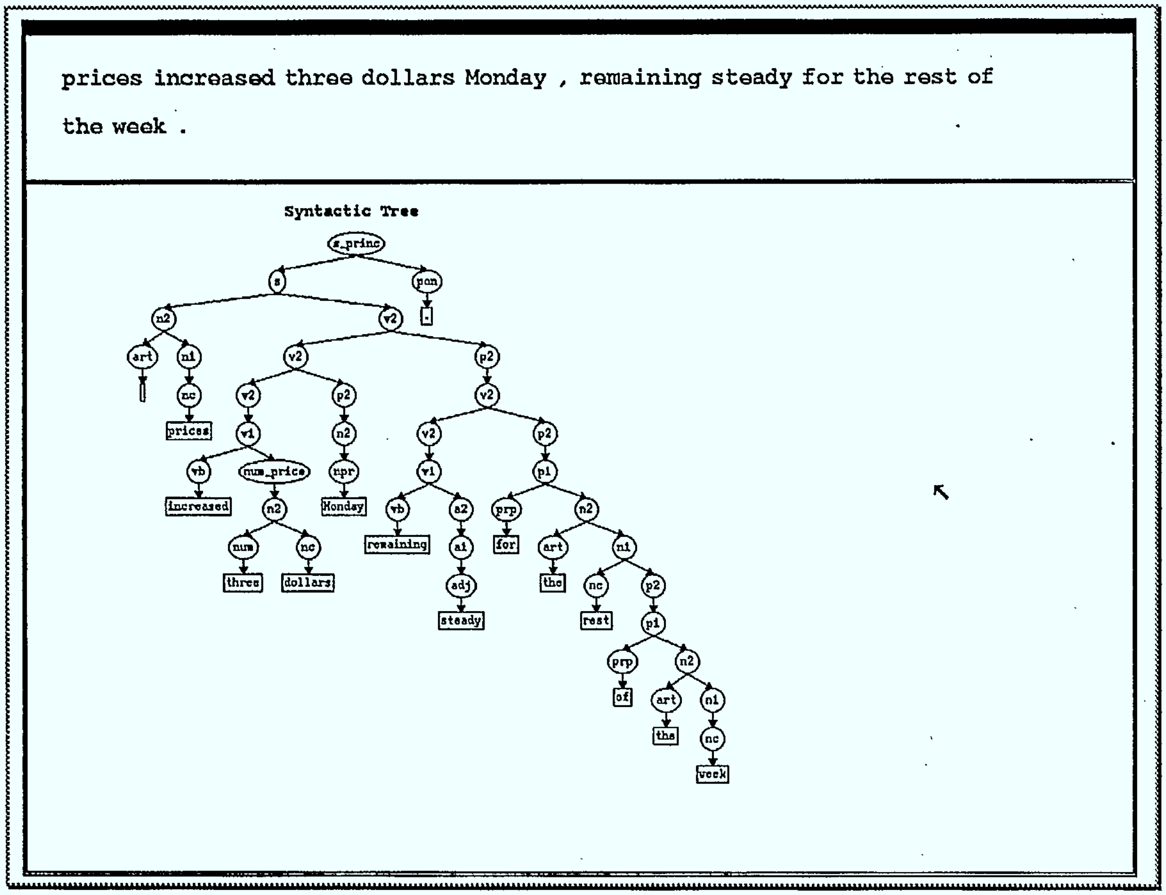


```
les prix s'accroissaient de $3 lundi , demeurant soutenus pour le reste de

la semaine .
```

Figure 27 Screen combining the data and results of the synthesis step.

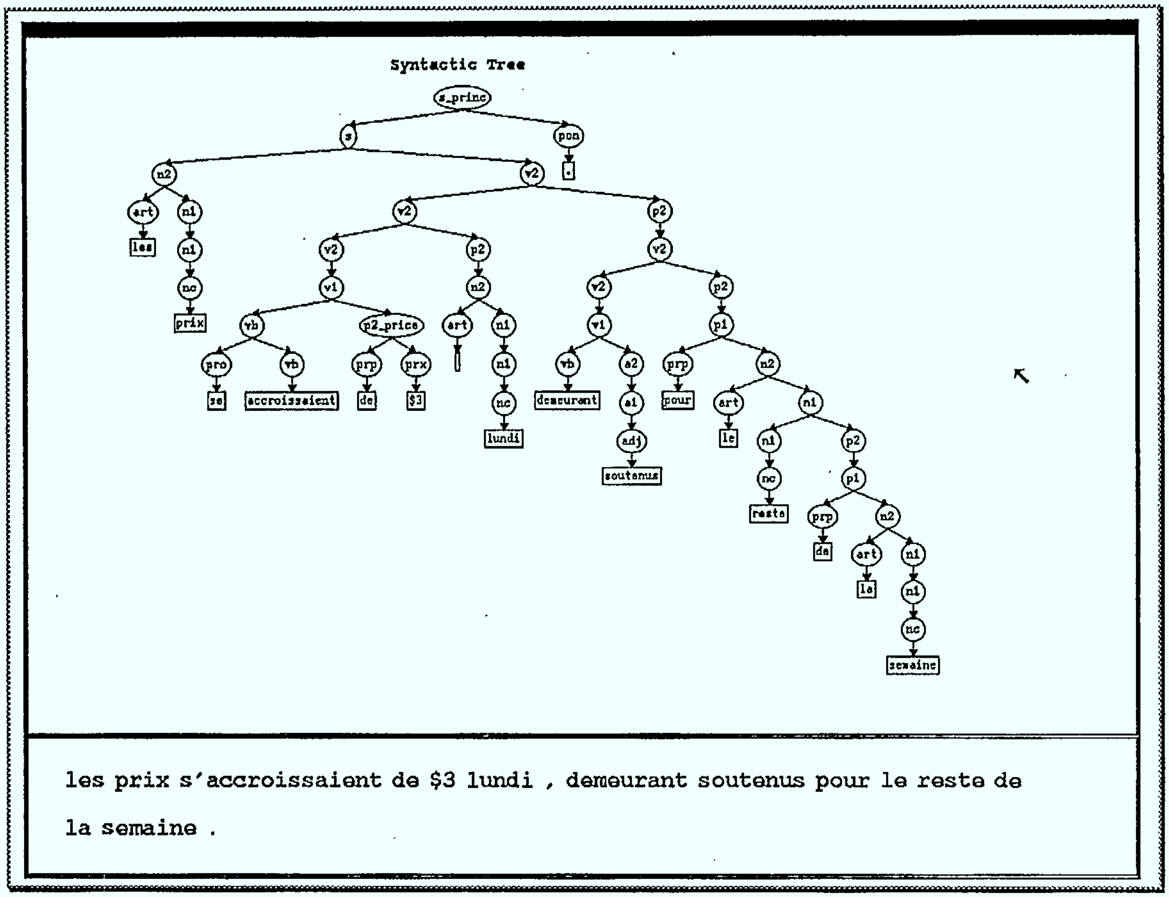
The analysis step screen contains an icon, in addition to the data and results (see Figure 25). When that icon is selected, a new screen appears, with exactly the same input data as on the previous screen but displaying as output an important auxiliary structure built during the analysis: the syntactic tree (see Figure 28).



prices increased three dollars Monday , remaining steady for the rest of the week .

Figure 28 Auxiliary screen containing the syntactic tree constructed during the analysis.

The synthesis step screen also contains an icon in addition to the data and results (see Figure 26). When that icon is selected, a new screen appears, containing the syntactic tree in the upper part of the screen and the synthesized sentence on the bottom (see Figure 29). The information on this screen is not arranged according to the rule that has been observed thus far, to the effect that the data from one step are to be shown on the top and the result on the bottom of the screen. This special arrangement was chosen to reflect the inversely symmetric relation between the analysis and synthesis.



les prix s'accroissaient de $3 lundi , demeurant soutenus pour le reste de la semaine .

Figure 29 Auxiliary screen containing the syntactic tree constructed during the synthesis step.

Design concepts

3

IRMA was developed on the basis of a few simple principles, which we wish to present here for a number of reasons. First of all, our general description of IRMA cannot be complete without some discussion of the original intentions of those who created its user interface. Secondly, this presentation obliges us to synthesize the ideas and principles we laid down to govern the construction of interfaces. And thirdly, we feel that this presentation is a first step on the path to establishing general principles for the design of *didactic* interfaces.

3.1. Demonstrator profile and target audience

Our first concerns when we approached the design stage were to develop a profile of demonstrators and identify our target audience.

We found that the demonstrators who would be using IRMA were mostly university students with no particular knowledge of computers. They were not computer science students or students familiar with the field. This meant that we could not hope that they would easily learn to work with complicated software and would quickly be able to handle complex operations. Moreover, they had to learn to use the system in a single day, during which they had to be instructed on CAT, the themes to be illustrated and the explanations to be given. That was an enormous amount of information – probably too much, if we wanted to ensure good demonstrations. Accordingly, for the learning process to be as efficient as possible, the IRMA interface had to be designed as a simple and sturdy tool, and the operations had to be easily understood and simple to remember.

The visitors to an exhibit such as Expotec, unlike the demonstrators, were not an easily identifiable group. Nevertheless, it could be assumed that such a group was certainly less homogenous than a group of people attending an international conference on computer linguistics, or those in a high school class. That meant that the members of the group had neither skills in a specialized field, nor age, in common. We therefore assumed that, if the public attending Expotec was that difficult to describe, it was because its motivations, interests and expectations were many and varied. Consequently, rather than using IRMA to illustrate only one or two simple, interesting and eye-catching subjects, we decided to aim the exhibit at these various audiences through an overview of a whole variety of themes, ranging from speech synthesis, which is probably one of the most impressive subjects, to the explanation of steps in translation, which is perhaps the most technical and least accessible topic.

We feel that this approach allows a demonstration like IRMA, encompassing so many different advanced technologies, to appeal to all the members of a varied audience. It was certain from the beginning that only a minority of those attending would be able to grasp all the details of the IRMA exhibit, but we categorically refused to design the exhibit to appeal to the lowest common denominator, i.e. the members of the audience with the least scientific knowledge. The fact remains that the explanation accompanying the demonstration was intentionally very simple, and generally tried to build on concepts that were already familiar or at least accessible to the majority of people (see Appendix).

3.2. Major themes

Difficulties associated with CAT

Computer-assisted translation (CAT) is a highly complex technological field, which thus far has probably experienced more resounding defeats than spectacular successes. One approach in which we have confidence is the design of specialized systems, with narrowly focussed skills, guaranteeing high-quality translations. CRITTER is one such system. Even though this reductionist approach increases the chance of success, there is nevertheless an unavoidable complexity inherent in the computer-assisted translation process.

We decided that a demonstration of CAT through IRMA should give spectators the opportunity to see or grasp the difficulties involved. The only proven method at our disposal to reach that goal, before the development of IRMA, was the display of the intermediate structures transmitted between the various steps in the translation process (see [3], for details on the first implementation of these displays). We therefore decided to apply that method, adapting it to IRMA's screen organization.

This solution had a twofold advantage. First of all, in most cases, it was highly probable that the spectators would be unable to grasp part or all of what is being presented. But since the seeming simplicity of the texts translated by CRITTER suggested that this was a trivial problem, we could at least hope that the impression left by the presentation on the operation of the system would be the correct one, i.e. of an intrinsically complex problem. Second, this illustration of the operation of CRITTER was a mean to increase the awareness of a few knowledgeable spectators concerning CAT, and to convince them, if necessary, that the demonstration involved no trickery.

Loss of information between spoken and written language

One aspect of which we all, as speakers, are rarely aware of is the loss of information between spoken and written language. Since IRMA uses a speech recognition peripheral, it was entirely appropriate to illustrate this phenomenon, particularly since it is simple to grasp. This aspect of spoken language is linked to the existence of heterographic homophones. French is probably a better example in that respect than English, since the plurals of regular nouns and many verb endings are not generally pronounced.

The decision on whether to illustrate this particularity of spoken language had to do with two constant concerns in our approach. Firstly, we wished to take a scientifically rigorous approach: IRMA was not a trumped-up demonstration whose only purpose was to amaze or mystify the audience. This application, if possible, had to reflect the current state of its component technologies, by showing their respective limits. Secondly, IRMA had to act as a magnifying glass for natural language phenomena, whether written or oral, recognized or synthesized, which are apparent only when one is exposed to them directly.

CRITTER'S ability to overcome ambiguity

CRITTER was not designed as part of the IRMA system. Rather, IRMA was simply a possible framework for the use of this translation system. That framework had the advantage of making it possible to test CRITTER in an environment for which it was not originally intended: the translation of texts entered in spoken form.

As compared with a written text, the equivalent dictated text suffers from a loss of information. That confusion is due, as we noted above, to the existence of heterographic homophones. CRITTER is able to resolve most of those uncertainties by means of its great expertise in the grammar and semantics of the subject area. This CRITTER feature made its integration in the IRMA system fully justifiable, with the intention of illustrating the potential of our translation system.

3.3. Organization of the interface

The implementation of the IRMA interface called for decisions on one-time problems such as the location of the windows on the screen, the choice of character sets and icons, etc. We had to find specific solutions to each problem, but throughout the process we tried to keep some governing concepts in mind, ideas underlying the design of IRMA. We will discuss those basic concepts in this Section, attempting to justify our choices somewhat, and will describe some applications of those principles, through examples or references to the previous chapter.

None of the following principles is based on an in-depth study of the literature in the field of cognitive ergonomics. We relied simply on our common sense, and experience in the design of such interfaces (see [3]).

Clarity and simplicity principles

One of our constant concerns during the development of the IRMA interface was our preference for *clear* and *simple*, as opposed to complex and flashy, solutions. These principles were applied separately to each case, since there is obviously no established method for arriving at a clear and simple solution. We could only choose between several possible options, selecting the one that we considered more clear and simple than the others. Rather than trying to define exactly what we mean by "clear" and "simple", we will illustrate our understanding of those concepts through concrete examples.

In the speech editor, we intentionally decided, after thorough reflection, to use the speech recognition software only for entering the words making up the sentence. A well-known example of 33 speech recognition software is the oral input of MS-DOS commands. The first solution considered for the speech editor was to allow the user to dictate both the words in the sentence and the editing commands, such as "beginning of sentence", "previous word" and "delete". This mixture is quite feasible, if a mini-glossary of commands is provided, and would probably have been simple for the demonstrators to learn. For the public, however, we felt that this mixture of commands and data might prove confusing. Hence we decided on a more primitive, but much more comprehensible, method.

In general, the simplicity of the IRMA user interface is fairly obvious when the IRMA screens are compared with those of the CRITTER system user interface [3]. The following are two other examples in which the simplicity criterion played a deciding role.

Firstly, we decided that IRMA would process only one sentence at a time, rather than having several sentences and translations on the screen at once. The latter approach would have resulted in a cluttered screen, making it much more difficult to read, and would necessarily have called for the use of a smaller, and hence less legible, character set. The speech editor that we designed, however, bears no similarity whatsoever to the kind of text editor or word processor visitors may be familiar with.

Secondly, and still in comparison with the CRITTER system demonstration interface, we preferred not to provide commands allowing the user to intervene in the processing by Prolog. Those commands allow the user to terminate an execution unconditionally, to order the search for another solution, and to re-synthesize a sentence following the analysis, thereby obtaining a paraphrase of the original sentence without translating it again. Those commands are useful for explaining in detail the various CRITTER features, but we thought it unadvisable and even detrimental to include them in IRMA, for there would be no point in providing an audience of non-specialists with such details.

Language of graphic symbols

The introduction of graphics screens and the concepts of icons, buttons and selectors have given birth to what may be called a *language of graphic symbols*. The power of this symbolic language lies in the explicit references it creates between the everyday world of objects suggested graphically and the computer world of procedures activated through the selection of those graphic symbols.

We decided to apply this principle to IRMA, hoping to make its interface more user-friendly and intuitive. Thus in the control panel, there are selectors allowing the user to pick one of a list of available values either by means of a circular list, or by direct access. In this case the reference to a television channel selector is sufficiently clear for users to understand the purpose of the selectors. The table of configuration parameters contains buttons as well as selectors; once again the similarity to a button to be pressed is clear, and the user can easily grasp how these controls are to be used.

The other commands in the interface are associated with icons. The symbolism of the icons lies solely in the images they present, and the references thereby created. While a garbage can, a keyboard and an auxiliary window are fairly simple to represent, a command such as "translate" is more difficult to render symbolically. Thus it was necessary to innovate. As in the demonstration interface, it is the CRITTER logo – the "cow" for the Agricultural Market Reports – that is used. While a translation is being carried out, the cow changes appearance, and that animation signals the step reached in the computer processing in progress. The parallelism between the two processes – translation and animation – creates a link between "translating" and "eating", since "eating" is the theme of the animation. To request a translation, the user asks the cow to eat, by selecting the appropriate icon.

The last point that created a problem for us, and to which the solution is also graphic, was the matter of how to represent heterographic homophones in the speech editor. We started from the principle that the text is *written* in the speech editor, and that what is written is durable and true. Accordingly, using that logic, homophones should appear on something resembling a piece of paper or a card, which can be deleted or moved around. What is symbolically hidden beneath the card is the solution to the basic problem: identifying the proper spelling from the available choices. This solution is calculated by CRITTER and appears when the interface simulates moving the card.

Flow of information and symmetry of displays

The IRMA console is used mainly to display the data manipulated by the system. One part of that information is redundant: the text entered and recognized orally, and the text translated and transmitted for speech synthesis. There are two reasons for providing the written copy of the dictated text: a necessary echo of the recognized words, and an illustration of the problem of homophones. The echo of the dictated text has a moderating function, in that the user waits for the word to be displayed before pronouncing the next word, and an "approval" function, since the display of the written data shows what the speech recognition system has "understood". There is no real reason for displaying the translated text. The only grounds for our decision are the need for symmetry in the display, which in practical terms means a symmetric organization of information on the screen. The echo of the words recognized (input) is presented in the upper part of the screen, and the translated text (output) below. The distribution illustrated by this first contact with IRMA very intuitively defines the flow of information on the screen, i.e. from top to bottom.

We maintained this same flow of information in the more technical IRMA screens that present the internal information manipulated by the system. The intrinsic symmetry of the translation process, in its breakdown into steps, also allowed us to keep a sort of symmetry in the information presented on the various screens.

To increase the idea of a flow of information, there is a distinction between the display of the input and output at every step. When the screen for the step in question appears, the input is shown in a block as static information. The output, on the other hand, is partially calculated at that specific time and is dynamically completed once it is displayed. This distinction applies only when the information is displayed for the first time. If, during the presentation, the user wishes to show the information in a screen that has already been displayed, the input and output appear in a block, unchanging. This feature increases spectators' awareness of the concept of a *machine* that begins with known input, and then calculates and displays the output.

Explicit actions-reactions

One aspect that is always confusing during a demonstration, from the spectator's point of view, is the seeming lack of an action provoking a visible reaction; the spectator sees only the result, and not the cause. If that situation occurs only once or twice it is not catastrophic; but if it recurs repeatedly during a sequence of operations, it is highly probable that the spectator will lose track of the flow of events in the demonstration.

We therefore attached a great deal of importance to respecting the rule to the effect that spectators must be able to identify easily the causes of a reaction by the system. Obviously, the least expensive solution is to require that the demonstrator keeps up a running flow of commentary on every step carried out. That is probably an excellent idea, but we wished to reinforce this aspect by obliging the demonstrator to perform explicit actions.

The best illustration of what we call an "explicit action" is the selection by means of the mouse of an icon whose meaning is evident. The spectator can watch the cursor move and stop on the icon, and understands those actions as the cause of the next step in the demonstration. This means of interaction between the demonstrator and the system is obviously slower and more restrictive than pressing a key on the keyboard and having it interpreted as a command. The following are some examples of explicit actions and reactions in IRMA.

Enabling and disabling the keyboard. The keyboard is activated by selecting an icon (see Figure 17). As immediate reactions, a small window opens and replaces the icon, and the cursor is automatically shifted into that window, activating it. The characters entered at the keyboard are then displayed in the window. To de-activate the keyboard, the user simply moves the cursor outside the window, also a visible action.

The "translate" command. The fact that the "translate" command is associated with an icon means that the user must explicitly move the cursor within the window to request a translation. It would have been possible to have the translation process start automatically when the final period in a sentence is encountered.

Nonetheless, IRMA is not devoid of exceptions to this rule, and the development of the interface involved removing steps that contained other steps. The following are a remaining exception to the rule, and a problem that was corrected.

Closing the screens associated with the steps in translation. Unlike the opening of screens associated with those steps, which occurs once an icon is selected, in keeping with our action-reaction principles, the window is closed by "selecting" any point away from all the objects in the window. Thus there is no specific location associated with this command. This operation may take spectators by surprise, since they may not necessarily notice the left-hand mouse button being clicked.

Opening the table of configuration parameters. It was originally by selecting the title `IRMA DEMO INTERFACE` that the demonstrator was to display the table of configuration parameters. At that point we wished to conceal those parameters and not mention them during the presentation. But when the concept of the *single-speaker* speech recognition system was introduced, the `Speaker` name became an entirely justified operating parameter, and its selection one of the actions in the demonstration script (see Appendix, Act I, Scene I). We therefore decided to associate a specific icon with the step of opening the table.

Simplicity of using the mouse

The mouse used with SUN stations has three buttons. That is probably two too many, considering the Macintosh model. Although we do not wish to advocate the use of a one-button mouse, the context in which IRMA is used is such that we need only one of the buttons on the SUN mouse. Our only reason for this simplification is the profile of the demonstrators, discussed earlier.

3.4. Design of tools

Centralization of peripheral commands

The design of IRMA called for the use of a fairly large selection of varied hardware (see Chapter 4). Thus it was essential that this profusion of hardware be transparent to IRMA users and spectators. We decided to conceal all that we wished to keep from the spectators inside the stand, and allow them to think that the IRMA console controlled everything: speech recognition, linguistic processing and speech synthesis. To simplify the user's task, and if we wished the illusion to be perfect, the management of the speech recognition and synthesis peripherals had to be centralized in the IRMA console, as well as the management of linguistic processing.

The main goal of this centralization is essentially of a practical nature. The combination of all the commands on a single screen obviously simplifies the use and management of the speech peripherals. Moreover, the illusion created through this seeming simplicity of management for the peripherals has a beneficial effect on the simplification of the demonstration. No technical details or strange operations, such as the use of a second keyboard for the recognition PC (see Section 4.2), disturbs the demonstration.

Limited command set for the speech editor

The speech editor was a major problem throughout the development of the IRMA interface. The design of the editor gave rise to many questions, particularly on the command set to be made available for the user. Finally, we settled on a minimal set, described in Section 2.3. It was not immediately apparent that this solution was the right one, or even the best one, partly because the examples of editors at our disposal did not meet the criteria we had set for ourselves, i.e. great simplicity of use and the possibility of easily correcting speech recognition errors.

Here is a simple problem that clearly illustrates the automatic tendency to follow the example of the technology we are exposed to: we wondered whether to introduce the concept of a *cursor* that would move around the written text. Clearly, a cursor is first of all a concept found in all editors. In our case, it was not a completely justified element. First of all, the editor was planned to be used only to echo the dictated sentence; an implicit cursor remaining at the end of the sentence being displayed would be sufficient, so that the next word would automatically be placed after the others. Secondly, in the event of an error in speech recognition, a word must be replaced. This means that a word must be identified, rather than a position in a word, as a real cursor would do. The only function that our *cursor-less* approach eliminates is the possibility of inserting a word between two others in the sentence. In our opinion, this function, which after all is of little utility in the use of the speech editor, did not in the least justify the investment necessary for the implementation of a real cursor.

Technical details

4

The IRMA kiosk on display at the Old Port of Montreal during the summer of 1989 in fact contained much more hardware than what was apparent to the eye. In addition to the SUN 4/110 workstation, its screen and keyboard, the micro-headset and the loudspeaker, a look inside the rear panels showed a portable SUN disk drive, two PC-compatible Zenith 386 computers and their keyboards, a PC screen, a DECtalk synthesizer, an audio amplifier, and two power bars (see Figure 1).

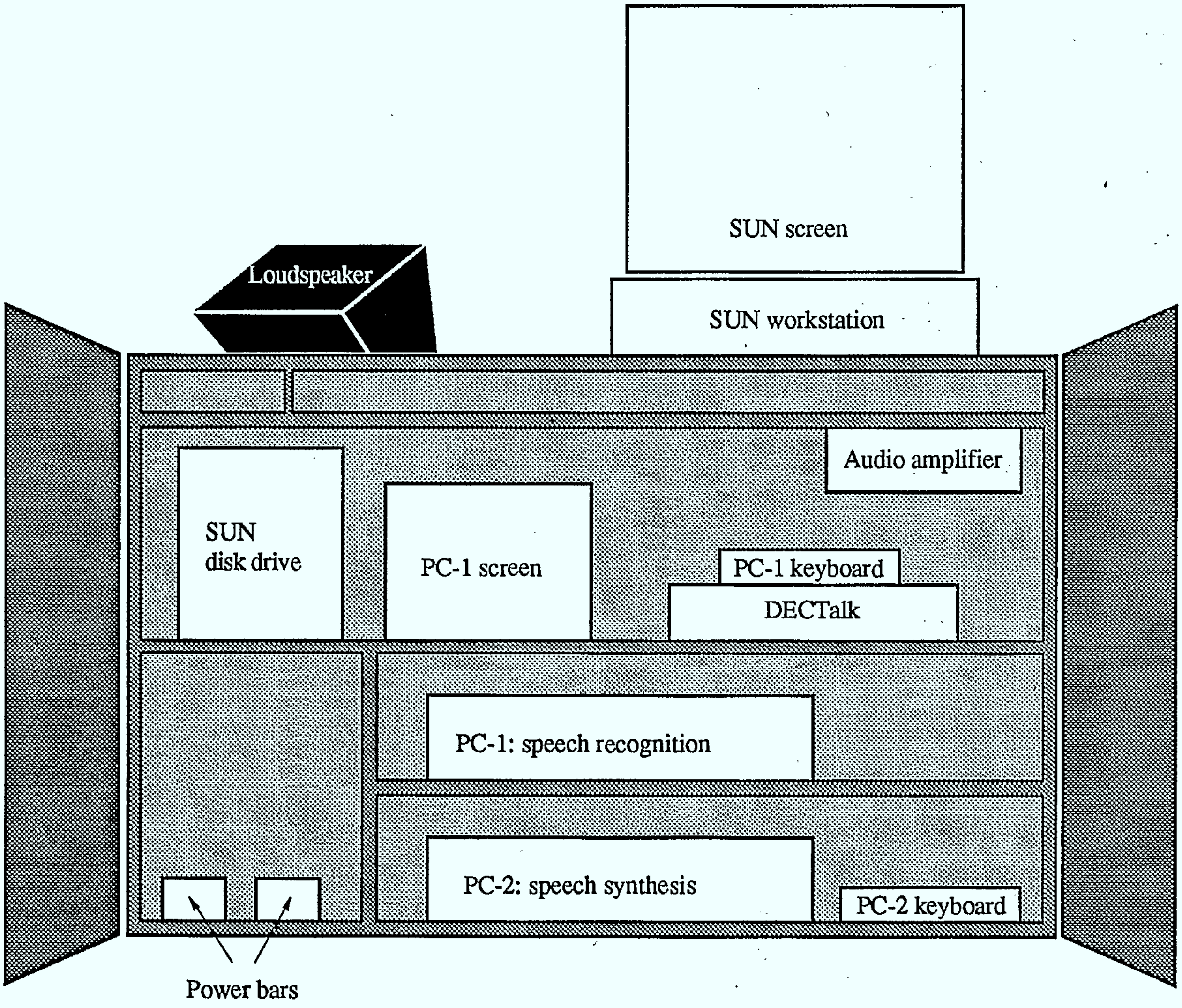


Figure 1 Layout of the hardware inside the IRMA kiosk.

Basically, all this equipment can be divided into three groups: *the Speech Recognition Group*, *the Linguistic Control and Processing Group*, and *the Speech Synthesis Group*. The first group comprises the micro-headset and a PC microcomputer (the "recognition PC"), with its keyboard and screen. The second group is made up of all the SUN equipment (workstation, keyboard, mouse, screen and portable disk drive). The third group consists of the second PC and its keyboard (the "synthesis PC"), the DECTalk device, the audio amplifier and the loudspeaker.

The fact that there are three independent computers in the IRMA configuration (the SUN 4/110 and the two PCs) is not an indication of any kind of parallelism in the operation of the system: the three machines generally are not all working at the same time and when they are, it is only by chance. If they are working simultaneously it is certainly to our advantage, but not of our planning. Were it not for the hardware requirements of some of the components at our disposal – the speech recognition and synthesis systems for French are designed for a PC-type machine, whereas the CRITTER system works only on a SUN workstation – we would certainly have been delighted to integrate all the operations. In fact, ease of use, one of our primary requirements in the design of IRMA, forced us to centralize all the control operations in the same user interface. The distribution of tasks among the three machines therefore obliged us to design relatively complex communications channels and protocols.

This chapter describes the internal configuration of each of the components of the three groups, as well as their interrelations.

4.1. Control and Processing Group

The Control and Processing Group has three main functions: managing the user interface, linguistic data processing (i.e. translating sentences) and synchronizing operations with the Speech Recognition and Synthesis Groups.

The SUN station does not need much in the way of special hardware (aside from its usual peripherals), but the software takes up a great deal of space in memory (12 or 16 Megabytes of RAM is generally not excessive). Moreover, a math co-processor ("Floating Point Unit") greatly accelerates the display of some graphics (the unit on display at Expotec 89 unfortunately did not have one).

For software, the CRITTER system is written entirely in Quintus Prolog [6], and the user interface uses the Quintus ProWINDOWS environment [7], an object-oriented programming package that can be applied to develop graphic interfaces for programs written in Quintus Prolog. The entire system runs in the SunView 1 environment [8], a multi-window interface of the SunOS operating system [9].

The activities of the other two groups are managed by the "recognition PC" and the "synthesis PC", respectively. The SUN station is physically linked to them by serial ports, and synchronizes operations by means of "messages". In short, the two PCs can be considered peripherals of the SUN station in the IRMA configuration. The nature of the communications protocols used is examined in greater detail in the following sections.

4.2. Recognition Group

When a sentence to be translated is dictated, all the input and word recognition operations are carried out by the recognition PC, which is at the centre of what we refer to as the Recognition Group. As they are identified, the recognized words are transmitted to the Control and Linguistic Processing Group, which manages the speech editor (see Section 2.3). In turn, the Control Group can send messages to the Recognition Group in order to change its operating mode.

The recognition PC thus has two tasks to be performed simultaneously: managing the speech input and recognition operations as such, and ensuring bi-directional communications with the Control Group. The first of those tasks is carried out by the VoiceScribe-1000 system, described in the following two sections. The second is performed by a group of programs that we will describe in the third section.

The VoiceScribe-1000 system and DragonKEY software

VoiceScribe-1000[3] is a single-speaker voice recognition system for isolated words. This is a commercial system, in the form of a card inserted in a PC, a micro-headset connected directly to the card, and a library of software programs used to take advantage of the card's features.

The VoiceScribe-1000 card transforms the electric signal from the microphone into a digital signal, and all the recognition operations are carried out by the software side of the system. We used the DragonKEY software [10] that came with the card, whose primary function is to allow PC applications to be controlled with commands given through a microphone.

DragonKEY generally handles speech recognition in two steps, corresponding to two separate table look-ups. First, the system tries to associate the digital signal received from the card with a tag representing a word known to the system. This is done by searching a table of digital-signal/tag pairs, called the application's "vocabulary". Second, a character string is associated with the tag identified, and used as an output value by DragonKEY. That second step is the result of searching a "lexicon", itself part of the application's "grammar". That (regular) grammar defines all the valid sequences of tags, and makes it possible to reduce the area to be searched during the speech recognition process.

The system manager is responsible for providing DragonKEY with the grammar (including the lexicon) for the application. The vocabulary is provided by the user, when he or she "trains" DragonKEY to "recognize his or her voice". This is done by repeating four or five times the "speech string" corresponding to each tag in the lexicon into the microphone. Since this is a single-speaker system, each user must prepare his or her own vocabulary.

DragonKEY works as a memory-resident program, i.e. it remains inactive in memory while other applications are executing, waiting for a pre-determined signal. That signal is the arrival of information from the VoiceScribe card. When it is received, the program sets the recognition mechanisms in motion, writing the character string associated with the digital signal recognized to the keyboard buffer. Thus when DragonKEY recognizes a

[3] VoiceScribe-1000 is a registered trademark of Dragon Systems, Inc.

word, the result is exactly the same as if the corresponding character string had been entered at the keyboard.

The VoiceScribe environment in IRMA

As mentioned above, IRMA uses the VoiceScribe-1000 system with DragonKEY operating software to recognize the dictated sentences to be translated by IRMA. When the user boots the PC, the "autoexec" batch file activates the Dragon card and loads the resident program, as well as those programs responsible for communications between the recognition PC and the rest of the IRMA system. We will discuss those programs in the next section. For the moment, we will look at what DragonKEY needs to correctly recognize the sentences dictated by the user.

Since IRMA translates from English to French and vice versa, the system must be able to recognize words in both languages. Thus the system must be provided with two separate grammars: one for English and one for French. The CRITTER system now knows over 2,000 word forms in English, and over 5,000 forms in French. Since the VoiceScribe-1000 system cannot manage a vocabulary of more than 1,000 words (whence its name), and IRMA exists only for demonstration purposes, we decided to limit the size of the lexicons to a fraction of CRITTER's capacity, recognizing just over 100 forms in each language.

Note that heterographic homophones (see Section 2.3) are identified at this point. From the point of view of grammatical description, a set of heterographic homophones is associated with a single tag, but the character string associated with it represents the list of all possible spellings. Thus the recognition system manager must identify the words with pronunciations sufficiently similar as to be confused. For the speech dictation aspect of IRMA, we decided on a grammar that could be described as "trivial": every sequence of words from the lexicon constitutes an acceptable sentence. In terms of regular expressions, `<sentence> = <word>*`. This implies that the grammatical validation of sentences (validation of the relative positions of words within a dictated sentence) is not done at the time of input. Rather the CRITTER system is responsible for doing so.

Of course, there is nothing particularly advantageous in this approach for the recognition operation, since all the words in the lexicon are "active" at any one time, i.e. they are all potential candidates. However, two factors justify this approach. First, the speech editor functions (see Section 2.3) assume that it is possible at any time to return to any point in the dictated sentence and re-dictate a word. For that "re-dictation" to be possible, it must be possible to resituate the speech analyzer in the proper grammatical context, a complicated operation in view of DragonKEY's capabilities (unless, of course, there is only one possible grammatical context, which is the case with our trivial grammar).

Secondly, the CRITTER computer-assisted translation system, to which the dictated sentences are sent, has extremely sophisticated French and English grammars at its disposal. The idea of having to duplicate this information for the vocal analyzer seemed particularly cumbersome and heavy, as any changes to the CRITTER grammars (and they are frequent) automatically call for equivalent changes in the speech grammars. What is more, the VoiceScribe syntactic description language is far from having the expressive power of CRITTER's grammatical formalisms. At best, we can hope to build VoiceScribe grammars that will define languages including those defined by the corresponding

CRITTER grammars. But that step, insofar as it attempts to extend beyond a trivial grammar, rapidly becomes one of daunting complexity.

The ideal solution clearly lies in the direct use of CRITTER grammars to limit the searches involved in speech recognition. Unfortunately, CRITTER's grammatical analysis is designed in a top-down fashion and operates through backtracking. This method is very poorly suited to real-time validation, as speech dictation normally requires.

In addition to the French and English grammars, the speech recognition system must have vocabularies of all potential users of the system available, for each of the two languages. Those vocabularies are created by the users themselves, from the DragonKEY "console". It is worth noting that users are not obliged to train the system to recognize their voices for all the words in the two lexicons. If they plan to use IRMA's speech recognition capacities only for a few selected sentences, it is necessary to train VoiceScribe only for the words in those sentences.

Communications between the Recognition and Control Groups

From the point of view of the recognition PC, communications with the Control Group consist, firstly, of the reception of command and synchronization messages from the latter through the serial port. Command messages are generally intended for the DragonKEY program: changing the grammar or vocabulary, turning the microphone on and off, etc. Synchronization messages are intended more for the communications device itself: requests for acknowledgment (see below) and other signals. Secondly, the recognition PC also sends messages to the Control Group, which may be words recognized by the VoiceScribe system or various signals such as acknowledgments, warnings that a command sent to DragonKEY has failed, etc.

The communications program as such comprises two processes: a "receiver" and a "transmitter". Those two processes operate in parallel in the MultiC environment [5], which is responsible for multitask management of C programs. The transmitter continuously reads the keyboard buffer, and sends its contents more or less verbatim to the serial port, for the Control Group. That is how the words recognized by DragonKEY are sent to the speech editor.

The receiver, for its part, reads the contents of the serial port continuously and sorts them. Synchronization messages are generally handled by the receiver itself. For example, consider a request for acknowledgment, mentioned above. This is a way for the Control Group to verify that the Recognition Group is listening and able to provide information. When the receiver receives such a request, it sends a message via the keyboard buffer, which is then read by the transmitter, which automatically re-transmits it to the Control Group. This verifies that the loop is working. The TEST button on the user interface (see *Configuration parameters*, in Section 2.2) initiates the request for such an acknowledgment. The test is successful if the acknowledgment arrives at the Control Group within a few seconds. Note, in passing, that whenever the Control Group sends a message to the Recognition Group, that message is followed by a request for acknowledgment. This is a simple way to check that everything is operating properly.

When the receiver receives a command message for DragonKEY, it issues an interrupt for the latter, and transmits the command by writing it to a specific memory location (that is DragonKEY's protocol for receiving its commands). It then waits for the result of the command (success or failure) and, if it fails, may send an error message to the Control Group by writing to the keyboard buffer (see Figure 2).

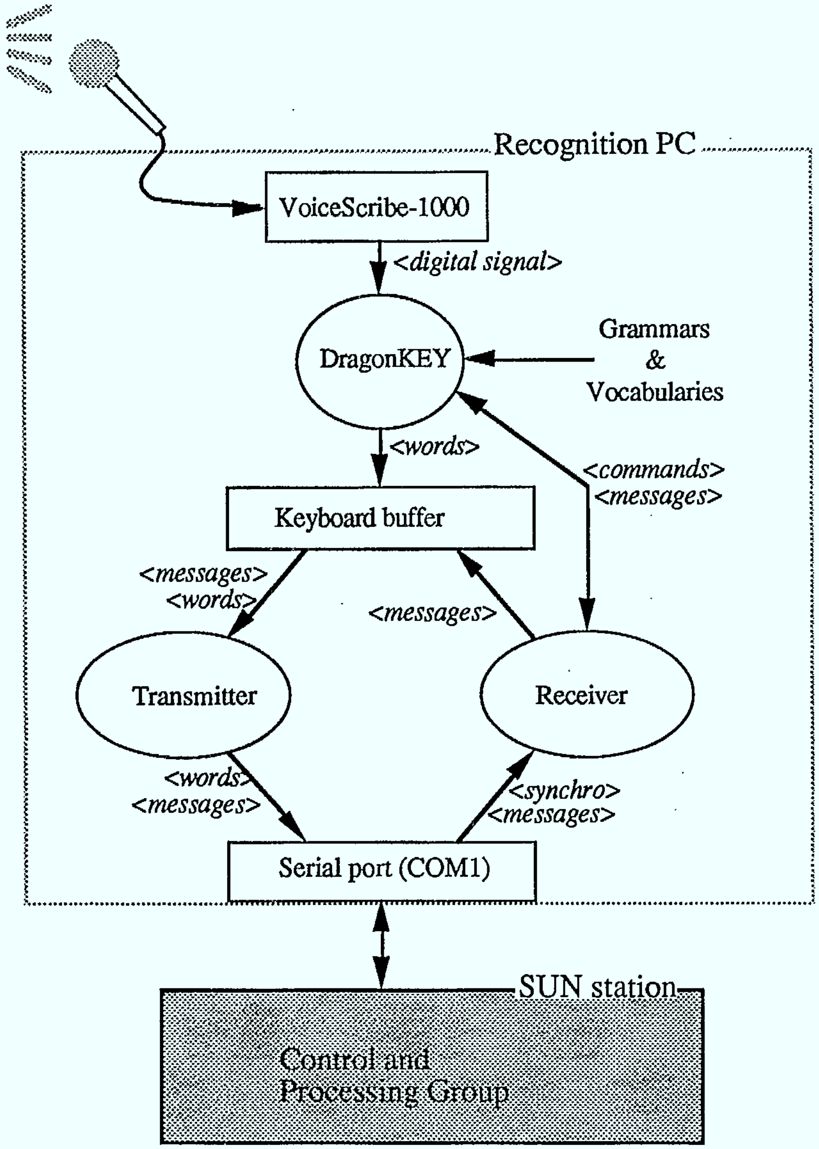


Figure 2 Data flow between the Recognition and Control Groups.

The list below gives all the types of messages transmitted over the serial cable between the two groups. The protocol is relatively simple: the first character in the message identifies its type, and a predefined character string marks the end. Everything that appears between those two markers ("..." in the list below) constitutes the body of the message. Two types of message need no end marker, namely the <^A> and <^C> issued by the Control Group. The end marker would be useless, since they consist of a single character with a special meaning for the receiver of the Recognition Group.

Recognition Group → Control Group:

`#...<LF>` character string associated with a word recognized by VoiceScribe

`<^B><LF>` acknowledgment

`<^B>...<LF>` general message (warning of command failure, etc.).

Control Group → Recognition Group:

`#...<NUL><LF>` command to DragonKEY with the possible values below for ...:

- `console` start the DragonKEY "console" program allowing vocabulary tables to be created by users, among other functions;
- `microphone on/off` turn microphone on or off;
- `overlay french/english` change source language (specifies VoiceScribe grammar to be used);
- `vocabulary ...` user identification (specifies which VoiceScribe vocabulary to use);

`<^A>` request for acknowledgment;

`<^C>` request for re-synchronization (the receiver of the Recognition Group cancels all characters received at this point and waits for a new message).

4.3. Synthesis Group

The sentences translated by CRITTER may be reproduced in speech form by the IRMA Synthesis Group. That group uses two separate speech synthesis systems for the purpose, namely DECtalk (for English synthesis) and TELEVOX (for French). This dichotomy is a clear illustration of the state of the art where speech synthesis is concerned; systems now available produce sometimes surprising results, but are often designed specifically for a given language and cannot be re-programmed.

DECtalk

The DECtalk[4] device is in the form of a peripheral (like a printer), which accepts English "sentences" as input and reproduces them in "spoken" form as output. The input is through a serial link, typically with a computer, and the output through a built-in loudspeaker, a standard audio output (the type to be plugged into a hi-fi system), or a telephone-type jack.

Although DECtalk can synthesize only English, it offers the user great flexibility in voice settings: feminine or masculine voices, flow, expressiveness, etc., and precision of pronunciation. IRMA uses the device as simply as possible, with the default voice ("Perfect Paul"), sending the sentences to be synthesized directly, with no attempt to improve pronunciation.

[4] DECtalk is a registered trademark of DIGITAL Equipment Corporation, Inc.

TELEVOX

The TELEVOX[5] system for speech synthesis in French resembles the VoiceScribe recognition system (see Section 4.2); it is an add-on PC-size card, accompanied by operating software. Like VoiceScribe, the entire system works in MS-DOS by means of a driver. When a sentence is synthesized, the communications program (described briefly in the following section) calls the driver and provides the sentence for synthesis. The TELEVOX software analyses the sentence and derives a series of "frames" from it, i.e. parameters that can be used by the synthesis card. The card then transforms those frames into an analog signal ready for amplification.

Sentences are sent to the system in the form of normal ASCII character strings. The user can insert certain "control characters" in the strings, aimed at modifying the pronunciation of the sentence. IRMA ignores those possibilities, and sends the sentence "as is" for synthesis.

Communications within the Synthesis Group and with the Control Group

The "synthesis PC" manages synthesis operations, controlling the distribution of work and communications with the two synthesis systems, as well as handling communications with the Control and Linguistic Processing Group.

Communications between the Control and Synthesis Groups are via a serial cable, linking the SUN station to the synthesis PC. They are unidirectional, and consist of only one kind of message, i.e. sentences to be synthesized by DECtalk or TELEVOX. Every sentence is preceded by a control character indicating whether it is in French (character f) or English (character a), and terminates with an end of sentence code (character ~). Inside those sentences, note that accented characters are encoded in accordance with the IBM standard extended ASCII character set. Communications are managed by a program in C, which continually reads the PC's serial port and transmits messages received to the appropriate synthesis system.

The DECtalk device is connected to the PC synthesis by means of a serial link. If the sentence to be synthesized is in English, it is sent to the second serial port on the PC (the first is used for the link with the Control Group). French sentences are sent to the TELEVOX card internally (see Figure 3).

As noted earlier, both speech synthesis systems allow the user to set some of their operating parameters. Those parameters are adjusted by the communications program of the Synthesis Group when the synthesis PC is turned on, and cannot be altered from the IRMA interface.

The audio outputs of the DECtalk and TELEVOX systems are connected to the audio amplifier, which includes a three-voice mixer (the third voice is used to amplify any "beeps" from the speech Recognition Group). Thus the sound output levels of the two synthesis systems can be set separately.

[5] TELEVOX is distributed by ELAN Informatique, under licence of CNET.

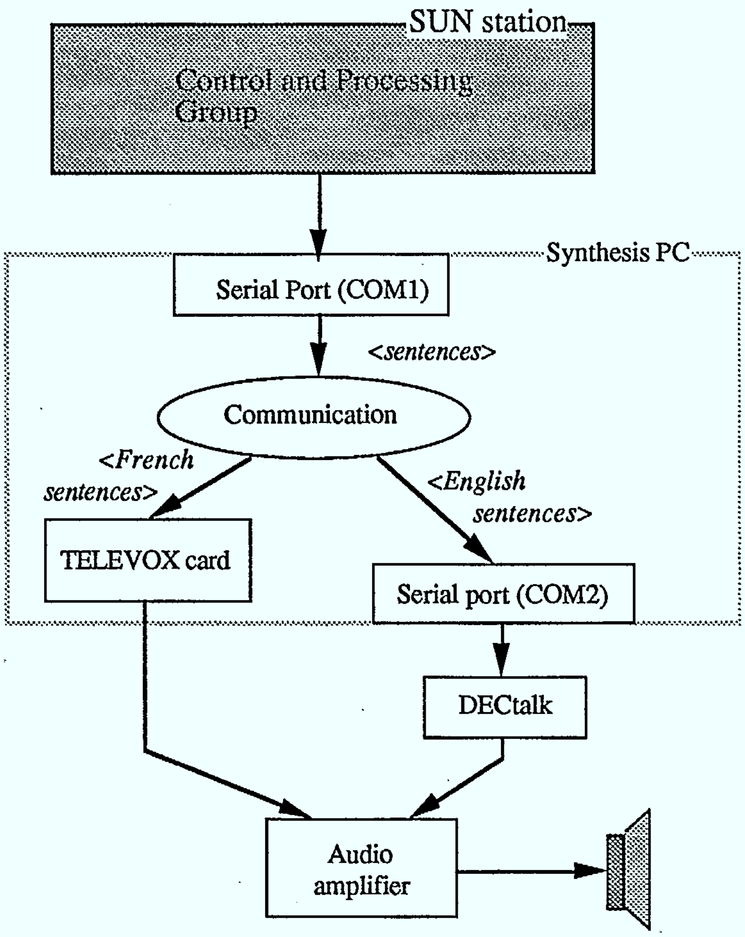


Figure 3 Data flow between the Speech Synthesis and Control Groups.

IRMA variants

5

Even after IRMA was installed at the Old Port of Montreal for Expotec, changes were still being made and experiments conducted. The two most interesting experiments are described below.

5.1. The Tubach experiment

As mentioned in Section 4.2, IRMA's speech recognition capabilities are based on the use of the VoiceScribe-1000 card. Dictation using that card is rather laborious, for two reasons: first, the pronunciation of words one by one is not particularly pleasant; and second, recognition errors are very frequent (it is not unheard of to have to repeat four or five words within a single sentence of about ten words). Those errors are attributable to the lack of syntactic constraints in the recognition operation – the use of a trivial grammar – (see *The VoiceScribe environment in IRMA*, in Section 4.2), and the very high level of ambient noise during the demonstrations, but they can also be attributed to the fact that the VoiceScribe system by no means represents the state of the art in speech recognition.

The visit by Jean-Pierre Tubach, a Professor at Télécom Paris (École Nationale des Télécommunications - ENST), to CWARC in summer 1989 led to an interesting experiment. Mr. Tubach took advantage of his sabbatical with us to attempt to design a more advanced voice input interface for IRMA. Two systems were tested for that purpose, namely Datavox VECSYS and Media50 X-COM, which are both billed as "add-on" systems, to be installed in a PC-type computer. The former is a single-speaker continuous speech recognition system, and the latter also works with continuous speech, but is designed for several speakers. The details of the experiment are discussed in [4].

The changes made to the IRMA user interface were kept to a minimum for the experiment, so that the new and old interfaces were identical in appearance. The only distinguishable difference in this IRMA variant was the dictation method, with sentences being dictated by "sentence members". This means that the following sentence could be dictated into the microphone (/ indicates an optional pause in speech, and // a mandatory pause):

```
Les prix de gros / de vaches de boucherie // ont augmenté / de
trois dollars le kilo // à Montréal // en raison d'arrivages très
faibles // point final
```

In fact, it would have been possible to dictate entire sentences with either of the speech recognition systems tested, but it was decided that the "sentence member" method was the most appropriate for this type of application.

The results of the recognition process were surprising to say the least, for both systems, and far superior to anything we could have hoped for with a system such as the VoiceScribe-1000. And for good reason, as the new systems represented an entirely different technology. However, it should be noted in VoiceScribe's defence that the success of the experiment with continuous speech recognition was also founded on the use of highly restrictive grammars, a solution that we dismissed when setting up the Speech Recognition Group. Hence, in this IRMA version, dictation was limited not only to a fraction of the vocabulary known to CRITTER (which was also the case for VoiceScribe), but also to a fraction of the syntactic constructions allowed by the translation system analyzer.

The experiment showed, nevertheless, that dictation for an environment such as IRMA's can be more than a toy. With a view to the design of a translator workstation based on the CRITTER technology, the results obtained point to the possibility of translators dictating translations or corrections as an alternative or complement to computer-assisted translation.

5.2. Multi-process IRMA

The Prolog program in charge of managing the user interface, linguistic data processing and synchronization of speech input and output operations is designed as a single process. When the user requests a translation, by selecting the cow icon, the program begins executing the predicates of the system's linguistic modules, and the management of the interface is interrupted until the operation requested is completed. If the user manipulates the data on the screen in other ways during this interval, those operations are memorized and executed in order when the program reassumes control of the interface.

One of the consequences of this architecture is that it prevents the user from interrupting a translation, other than by interrupting the execution of the whole program. This type of interruption is very useful, however, because the system occasionally "hangs" when working with ungrammatical or particularly complex sentences. Rather than the usual few seconds, a translation may require several minutes of processing in that case, and will produce an incorrect translation at best. Obviously, such problems can be very embarrassing during a demonstration!

One solution to this problem is to divide the main program into two or more programs that are executed as separate processes, as explained in [3].

The design of a "multi-process" version of IRMA was undertaken, in which the interface management part of the system is the "master" process, and the linguistic processing aspect executes as a "slave". For reasons we will explain in detail later (see Chapter 6), the implementation of this version was not completed. The separation of tasks into two programs and the design of inter-process communications protocols were carried out, but the work was interrupted at the time when the appearance of the user interface was to be modified to reflect the system's new features (estimated at one-half day's work). The advantages of such a feature are quite clear, nevertheless.

The first of those advantages is the uninterrupted management of the interface during the execution of translations. When the user asks IRMA to carry out a translation, the interface management process delegates the request to the linguistic processing process, and continues to manage the interface while waiting until the slave has completed the job. That means that the user is free to perform other operations during that time, such as dictating another sentence, asking for a change in the source language, etc. When the slave has completed the job requested, the master retrieves the information and displays it.

Secondly, if the machine on which the master process is running is one of a network of SUN stations, it is possible to start the execution of the slave on another machine in the network. This means that the machine dedicated to the master process is freed up, generally with an improvement in the response time for the interface, as well as in translation time, if the slave process is executed on a more powerful machine.

The third advantage, of course, is the possibility of interrupting a translation in progress (or any other job assigned to the slave for execution). In this version, the basic IRMA screen includes a supplementary interrupt button. When the user selects this button with the mouse, the master process sends a signal to the slave process, which interrupts the execution of the current job. The system is then ready to carry out another translation.

This method of breaking up the work between a master process and a slave process (interface management versus linguistic processing) was also adopted for the implementation of the standard CRITTER user interface [3], and has given excellent results. A further breakdown of the process, and wider use of the possibilities offered by the parallelism of the system in general, are research subjects that we find extremely promising.

Conclusion

6

Following its intensive use during Expotec, a period during which over 500 demonstrations were conducted, the IRMA kiosk was dismantled and brought back to CWARC. Over the same period, it was decided that the associated software would no longer be maintained or improved, for the following reasons:

Firstly, the quantity of hardware necessary for the operation of IRMA and the complexity of the configuration would have obliged us to devote a site and all that hardware just to IRMA demonstrations. That would obviously be a waste of resources, all the more difficult to justify in that we have other, less cumbersome, means of demonstrating the CRITTER system's capabilities.

Secondly, we feel that we have reached a limit to how much the speech peripherals can be integrated into the CRITTER system. Proceeding in that direction would call for a massive investment in the purchase of more sophisticated hardware, and consideration of major changes to the techniques developed for the translation system. The limited resources of the CAT group and the objectives set for CRITTER force us, for the moment, to concentrate only on that project.

Thirdly, IRMA was designed specially for Expotec 89: the demonstration was intended for the general public, and stressed mainly the themes of the exhibition. When removed from that context, the system loses much of its raison d'être. Furthermore, as a demonstration tool, it does not adequately represent the CAT group's research aims.

The decision to abandon IRMA does not mean to suggest, however, that the experiment was not a fruitful one. Quite the contrary. The experience was enriching in several ways.

IRMA is first and foremost a demonstration program. As such, its development caused us to reflect considerably on the impact of such demonstrations and the general principles for designing them (see Chapter 3). While this exercise had no or little direct scientific spinoffs, its impact as a promotional tool and a means of presenting the results of a basic research project to the general public is beyond doubt. In that way, the repercussions of IRMA on the CRITTER project far surpassed our expectations. We are now convinced that the development of demonstration software should be part of the planning for any major computer research project, just as important as preparing technical reports and scientific articles.

From another point of view, IRMA illustrated the potential of the CRITTER system to play an active role in the speech recognition process. Although that participation in IRMA was limited to overcoming the ambiguity inherent in heterographic homophones, it clearly appears that it would be possible to go much further. Thus the design of IRMA gave us a chance to measure the extent and determine the nature of the work that would be necessary to integrate a speech dictation peripheral in the CRITTER system, should the inclusion of such a device become desirable in the context of the actual use of a computer-assisted translation package. More specifically, it appears that the ideal solution lies in the inclusion of phonetic information within the CRITTER dictionaries, and the use of a speech recognition system capable of producing "phoneme lattices". Under those conditions, the use of input-driven analysis techniques for the translation system would improve control of the speech recognition and the visual echo for the speaker.

The development of a Micro-computerized Phonetic French Dictionary (DMFP) is now under consideration, and may lead to expanding CRITTER's dictionaries. Recognition systems such as the ones mentioned above exist, for example the "Phonetic Engine" distributed by Speech Systems, Inc. With regard to analysis methods, our group is currently exploring the possibility of replacing CRITTER's existing grammars with mechanisms operating as the input string is entered.

Thus it is not unthinkable or even unrealistic to imagine a high-performance speech recognition peripheral integrated in a computer-assisted translation environment, in the fairly near future.

Bibliography

[1] P. Isabelle, M. Dymetman, E. Macklovitch, *CRITTER: un système de traduction pour les rapports sur les marchés agricoles*, rapport technique du CCRIT, Ministère des Communications Canada, ISBN 0-662-95181-6, 1988, 24 pages.

[2] L. Bourbeau, F. Pinard, *Dictionnaire Micro-informatisé du Français : Description de ses propriétés morpho-syntaxiques*, Progiciels Bourbeau Pinard inc., ISBN 2-9800622-1-9, 1987, 143 pages.

[3] Jean-Luc Cochard, Pierre Plamondon, *L'interface d'utilisation du système CRITTER*, rapport technique du CCRIT, Ministère des Communications Canada, à paraître, 25 pages.

[4] J.-P. Tubach, *Reconnaissance de parole continue en entrée d'un système de traduction assisté en français et en anglais*, rapport technique du CCRIT, Ministère des Communications Canada, à paraître.

[5] P. Mabilleau, P. Hamel, *Multi_C, A Multitasking Kernel and its Application to Voice Response on a PC*, proceedings of Speech Tech'88, 1988, pp. 157-160.

[6] *Quintus Prolog Reference Manual*, Quintus Computer Systems, Inc., Mountain View, California, 1987.

[7] *Quintus ProWINDOWS User's Guide*, SunView Version, Quintus Computer Systems, Inc. and International Computers, Limited, Mountain View, California, 1988.

[8] *SunView 1 Programmer's Guide*, Sun Microsystems, Inc., 1988.

[9] *SunOS Reference Manual*, Sun Microsystems, Inc., 1988.

[10] *VoiceScribe™-1000: Speech Application Designer's Handbook*, release 2.10, Dragon Systems, MA, 1986.

Appendix

Demonstration script
EXPOTEC 89

Computer-Assisted Translation Group

Canadian Workplace Automation
Research Centre
Department of Communications

Laval

June 1989

Introduction

This document is a detailed description of a proposed script for demonstrators of the IRMA system at EXPOTEC 89. The script has been designed to highlight the most interesting features of the system.

It is important to understand that IRMA's features are in no way limited to the specific examples used in this script. In fact, the system is generally able to handle all Agricultural Market Reports. However, it does not contain the necessary information for translating any other kind of text. Moreover, since IRMA is a laboratory prototype that is still at a very preliminary stage, there is no guarantee that it will perform adequately on examples "improvised" by demonstrators. Consequently, it is strongly recommended that the demonstrators use only the examples given in the following script.

The demonstration is broken down into four sections:

- the three steps in the translation process, illustrated through the translation of anEnglish sentence;
- the reversibility of IRMA, illustrated by the re-translation into English of the sentence translated in the previous step;
- speech synthesis: translation of an English sentence with the loudspeaker on;
- speech recognition: translation of a French sentence dictated into the microphone.

Before you begin:

Ensure that the system has been correctly started up, and that the *initial interface* is displayed on the screen. If the "source text" and/or "target text" areas are not *blank*, select the garbage can icon to clear them.

Act I

STEPS IN A TRANSLATION

Act I, Scene I: Preliminaries

Remarks

IRMA is a *prototype* system capable of translating from English to French and vice versa, both in writing and orally. Given the great difficulty of the problems, IRMA is oriented toward the translation of a simple sub-language: Agricultural Market Reports. The technology is applicable to many other sub-fields, however.

➠ **Display the configuration parameters by selecting (with the left-hand mouse button) the appropriate icon on the control panel (the cow, above the IRMA icon).**

Remarks

We will be using speech recognition later on in the demonstration. The recognition process uses specific models for every speaker, and so we have to tell the system who will be dictating.

➠ **Check that the proper speaker has been selected, and close the configuration parameters screen by moving the cursor out of the window.**

➠ **Select the following parameters:**

Source language: English
Display structures: yes
Microphone: off
Loudspeaker: off

Remarks

We are now going to translate a sentence from English to French. The process has three steps: analysis, transfer and synthesis. At the end of each step, the system will display graphics representing the results.

Act I, Scene II: Sentence analysis

➠ **Display the mini-editor, by selecting the keyboard icon, and press the F1 key. The sentence *prices increased three dollars Monday, remaining steady for the rest of the week* appears as the source text.**

Remarks

The system can translate sentences such as this one, which is entirely typical of the sub-language of Agricultural Market Reports produced by Agriculture Canada.

➠ **Select the IRMA icon (the cow).**

Remarks

Note how the IRMA icon changes while the system is working. Note also that the A icon (analysis) appears in reverse video. IRMA is analysing the sentence. In this step, the system reads and understands the text to be translated. This is a highly complex process, and a crucial one: the system cannot produce a correct translation if it misunderstands the source language text. Once the analysis is completed, the system will build graphics describing its understanding of the sentence analysed.

➠ **As soon as the IRMA icon returns to its original appearance, select the A icon: the semantic graph will appear.**

Remarks

This semantic graph is a formal representation of the meaning of the sentence analysed. It explicitly identifies the objects and relations in the sentence, as they apply in the world of market reports. For example, the semantic concept of "ascending movement" is formalized as an "increase(X,Y)" relation, between a certain quantity X (in our example, the prices) and a certain increment Y (in our example, a number of dollars). In theory, all sentences with the same meaning are represented by the same semantic graph. An important characteristic of semantic graphs is that they *explicitly* identify the information presented *implicitly* in the sentences. In our sample sentence, it is fairly explicit that prices are increasing (since "prices" is the subject of the verb "increase"), but it is entirely implicit that "steady" also applies to prices. The analysis process determines this fact and reflects it in the semantic graph.

➠ **Select the GRAM icon.**

Remarks

To build the semantic graph of a given sentence, IRMA must first carry out a complete grammatical breakdown of the sentence. The result of that breakdown is a *syntactic tree*. The tree in our sample sentence shows that we are dealing with:

- a main sentence (*s_princ*), broken down into a sentence kernel (*s*) and punctuation (*pon*);
- the sentence kernel is broken down into a subject noun group (*n2*) and a verb group (*v2*);
- the noun group is broken down into an article (*art*) and a common noun (*nc*);
- the main verb (*vb*) of the verb group is *increased*;
- etc.

➠ **Click anywhere in the "syntactic tree" window to return to the "semantic graph" window.**

Remarks

Note that the semantic graph, although it represents the meaning of the sentence, remains linked to the source language. The concepts that appear on the nodes (such as *increase*) are oriented toward the language analysed.

➠ **Click anywhere in the "semantic graph" window to return to the main screen.**

Act I, Scene III: Sentence transfer

➠ **Click on the IRMA icon again.**

Remarks

The next step consists in using the semantic graph of the analysed sentence to produce an equivalent semantic graph, oriented toward the target language.

➠ **As soon as the IRMA and T icons return to normal, click on the T icon to display the source and target semantic graphs.**

Remarks

The nodes in the target graph are now labelled with concepts oriented toward the vocabulary of the target language. Thus it is during this transfer step that the actual translation operations take place. Those operations are greatly simplified by the fact that they are conducted at the very *abstract* level of the semantic representations. At that level, the grammatical details that vary greatly from one language to another are left aside. The source and target graphs in our example are perfectly *isomorphic*, even though the grammatical structures of the sentences involved are different. The source and target language semantic graphs may be different when the two languages express the same idea in very different ways (e.g., *John swam across the river* <-> *John a traversé la rivière à la nage*).

➠ **Click anywhere in the TRANSFER window to return to the main screen.**

Act I, Scene IV: Synthesis of the sentence into the target language

➠ **Click on the IRMA icon again.**

Remarks

The S icon (*synthesis*) is highlighted. IRMA is now synthesizing the French sentence from the target semantic graph, by consulting a French dictionary and applying all the French grammar rules, so as to place the words in the correct order, conjugate verbs, make agreements of gender and number, and so on.

➠ **Once the IRMA and S icons have returned to normal, the translation *les prix augmentaient de 3 dollars lundi, demeurant soutenus pour le reste de la semaine* is displayed under "target text".**

➠ **Click on the S icon to display the target semantic graph and the synthesized sentence.**

Remarks

IRMA has managed to produce a correct sentence in French from the target semantic graph. Note that some choices made in the target sentence clearly illustrate IRMA's ability to understand. For example, the adjective *soutenus* agrees correctly with *les prix*, since IRMA understands that it is the prices that are steady.

➠ **Click on the GRAM Icon.**

Remarks

While the analysis process starts from a sentence and produces a semantic graph, passing through the syntactic tree step, the synthesis process works the other way around: it produces a sentence from a semantic graph. Once again, there is a syntactic tree as an intermediate step.

➠ **Click anywhere in the "syntactic tree" window to return to the "semantic graph" window.**

➠ **Click anywhere in the "semantic graph" window to return to the main screen.**

Act II

THE REVERSIBILITY OF IRMA

Remarks

IRMA has a unique feature that distinguishes it from other computer-assisted translation systems: it is *reversible*, meaning that it can translate from French to English and English to French, using the same linguistic descriptions (dictionaries, grammars, etc.). To illustrate this feature, we will now re-translate the sentence that IRMA has just produced, back into English. This time we will not ask the system to display the graphics.

➠ **Select the following parameters:**

Source language: French
Display structures: no
Microphone: off
Loudspeaker: off

➠ **If you wish, clear the source language zone by selecting the garbage can icon.**

➠ **Display the mini-editor by selecting the keyboard icon, and press the F1 key (the sentence *les prix augmentaient de 3 dollars lundi, demeurant soutenus pour le reste de la semaine*, which was under "target text", will now appear as the "source text").**

➠ **Click on the IRMA icon.**

Remarks

IRMA now produces a re-translation back into English. (The translation is displayed under "target text".) Note that the translation obtained is correct. Note also that it is not identical to the original English sentence, but is entirely equivalent.

➠ **You may re-display the original English sentence under "source text" by changing the "source language" parameter to "English", displaying the mini-editor and pressing F1.**

Remarks

This variation is a natural outcome of the fact that IRMA generally knows several ways of expressing the same idea. This *paraphrasing* feature can be used to obtain a bit of variety in the translations produced.

Act III

SPEECH SYNTHESIS

Act III, Scene I

Remarks

IRMA includes a speech synthesis device allowing the system to read out loud the French and English translations it produces.

➠ **Select the following parameters:**
Source language: English
Display structures: no
Microphone: off
Loudspeaker: on

➠ **Display the mini-editor by selecting the keyboard icon, and press the F2 key (the sentence *fat cows were discounted, while the top heifers were up $3-$4* is displayed as the "source text").**

➠ **Click on the IRMA icon.**

Remarks

Draw the spectators' attention to the "target text" zone and the loudspeaker. Ask for silence during the speech synthesis.

➠ **The sentence *les vaches grasses se vendaient à rabais, alors que les meilleures génisses augmentaient de 3 à 4 dollars* is displayed as the "target text" and read out loud by the speech synthesis module.**

➠ **Simply select the loudspeaker icon to play the speech synthesis again.**

➠ **You may point out that IRMA is not limited to word-for-word translation, as shown by the translation of *discount* by *vendre à rabais.***

Act III, Scene II (optional)

➠ **At this point you may show that the synthesis also works with English as the target language. Select French as the source language, display the French source text (F2 in the mini-editor) and request the translation.**

Act IV

SPEECH RECOGNITION

Act IV, Scene I

Remarks

IRMA can also work with dictated sentences. Given the actual state of the technology, this capacity remains somewhat limited. The system recognizes individual words, and must be specifically trained for each reader.

➡ **Select the following parameters:**

Source language: French
Display structures: no
Microphone: on
Loudspeaker: on

➡ **Dictate the sentence *les cours des bovins de court engraissement baissaient de cinq pour cent.***

➡ **Turn the microphone off.**

Remarks

Note that the speech signal contains less information than a written signal. The noun *cours* and the adjective *court* (or *courts*) are pronounced in exactly the same way (they are *homophones*). In our sample sentence, we have two occurrences of the sound "*court*", one as a noun and the other as an adjective. The noun *bovin* is pronounced in the same way in the singular and plural forms. Finally, *baissait* and *baissaient* are pronounced in the same way. Moreover, the slight difference in the pronunciation of *baissé* and *baisser* is too small for the speech recognition device to distinguish them reliably.

All that means that there is a great deal of uncertainty about the output of the speech recognition module. To translate correctly, IRMA needs mechanisms to solve those ambiguities.

➡ **Click on the IRMA icon, and demonstrate the selection of homophones.**

Remarks

IRMA does not translate words in isolation. Translation, as we saw earlier, is based on a complete analysis of the grammar and semantics of sentences. Owing to the sophistication of its analysis, IRMA was able to correctly select the homophones in our sample sentence, and we could obtain a correct translation (*bovin de court engraissement* is translated by *feeder*). This ability is a result of IRMA's great expertise in the grammar and semantics of the texts concerned.

Act IV, Scene II (optional)

➠ At this point you may show that the speech recognition feature also works with English as the source language, by selecting English as the source language, turning the microphone on and dictating *lean yearlings sold at steady prices.*

➠ Click on the IRMA icon to request the translation and synthesis of the French.

Pour plus de détails,
veuillez communiquer avec :

Le Centre canadien de recherche
sur l'informatisation du travail
1575, boulevard Chomedey
Laval (Québec)
H7V 2X2
(514) 682-3400

For more information,
please contact:

Canadian Workplace
Automation Research Centre
1575 Chomedey Blvd.
Laval, Quebec
H7V 2X2
(514) 682-3400