American Journal of Computational Linguistics Microfiche 76: 62

# The TARGET Project's Interactive Computerized Multilingual Dictionary

**John Burge**
Departments of Modern Languages and Computer Science
Carnegie -Mellon University, Pittsburgh, Pa. 15213

Summary:

This document is a brief introduction to Carnegie-Mellon University's interactive computerized multilingual dictionary. It describes the use of this dictionary both by *translators* in the course of their work and by the *terminologists* responsible for updating and maintaining it. This discussion is placed in the context of the overall effort (known as the *Target Project*) to provide aids to translators. A final section presents the solution to the problem of representation of term equivalence adopted in Target.

## 1. The Target Project

Target is an interdisciplinary research project undertaken jointly by the Translation Center and the Department of Computer Science at Carnegie-Mellon University to investigate and develop computer aids for language translation.

Since high quality automatic translation does not seem to be immediately realizable, our efforts at introducing computerization into the translation task have been directed towards providing practical aids for translators. Working with the assumption that each translator can be provided with a standard video terminal connected by a dial-up line to a remote computing facility,[2] we are exploring primarily two aids. They are (1) an interactive multilingual dictionary, and (2) an environment consisting of (1) plus text manipulation facilities within a windowed page editing environment. The latter research will be described in a future AJCL paper; this document justifies and describes only the former, how it is accessed and how it is built up and maintained.

[1] It is an amended version of an informal description of the TARGET and TERMIN programs demonstrated at the Foreign Broadcast Information Services seminar on Aids to Translators, Washington, DC, May 1978

The configuration in daily use by the Translation Center at Carnegie-Mellon University involves a Lear Siegler ADM-3 terminal connected by a 300 baud dial up line to a PDP-10 run under the TOPS-10 operating system by the Computer Science Department and shared simultaneously by users working on many different projects

The primary motivation for the interactive dictionary is that a technical translator may spend up to 60% of his or her time simply looking up terms. This may include unsuccessful searches in several dictionaries, partly because these dictionaries are out of date by the time they are published. An interactive computerized dictionary would provide effectively "immediate" access to entries and moreover could be kept constantly updated at the central computing facility.

Currently the dictionary contains specialized terminology in English, French and German in a number of fields Specialized terminology was chosen because this is often most helpful in practice to the professional translator and also because this is where the benefits of standardization could be most immediately apparent The languages were chosen because they are the most immediately useful in the local environment, as were the fields (mainly finance, business and iron, steel and mining technology).

The next section shows in some detail how a translator would access the dictionary and determine a correct equivalent The section after that describes the facilities used to maintain and augment the dictionary The interface to the dictionary described in the next two sections represents the fruits of continual close cooperation over an extended period of time between researchers from the Computer Science Department and from the Department of Modern Languages Such cooperation, while it presents many problems initially, is a *sine qua non* of success in a venture such as Target.

While performing initial studies for the representation of equivalence between terms the most central relation in a multilingual dictionary -- we have departed from the common practice of using an alingual set of concepts realized differently in different languages Close examination showed that this could not accommodate some nuances of meaning in disparate languages and was not precise enough for making inferences when a particular equivalence was not already present in the dictionary. Moreover, it was found to be less efficient than another method which was investigated and ultimately adopted. Some arguments proposed for adopting this different method are set forth in the final section of this document

## 2. The TARGET Program

TARGET is also the name of the program used by translators to access the entries in the dictionary while doing their translation work This section describes how it is used. The illustrations are exact traces of the interaction between the program (in a roman font) and the translator (in an italic font).

We are first asked for the term names and the languages we wish to translate From (*i.e.* the source language) and To (*i.e.* the target language):

```
Term: bond
From Language: en
To Language: fr
```

Now, if there is only *one* equivalent for that term between those languages, we shall get that equivalent directly. In this case we have a choice to make:

```
    Term: bond
    From Language: en
    To Language: fr
  bond     Chemistry: Theoretical Chemistry;    (ch4)
           The Nuclear Industry: Nuclear Energy;    (at6)
           Financial Affairs - Taxation - Customs;    (fi)
    Select Code:
```

Let us say the article we are translating is in Chemistry. Then we just type the appropiate code. These are in parentheses in the example and are the same codes as used in the EEC's Eurodicautom system. Here we select a code:

```
    Term: bond
    From Language: en
    To Language: fr
  bond     Chemistry: Theoretical Chemistry;    (ch4)
           The Nuclear Industry: Nuclear Energy;    (at6)
           Financial Affairs - Taxation - Customs;    (fi)
    Select Code: ch4
```

and we shall get the appropriate fiche:

```
    Select Code: ch4

  bond     liaison (FR)

           Chemistry: Theoretical Chemistry;
           The Nuclear Industry: Nuclear Energy;

  Reference Terms:     bonding energy

    Term:
```

We have been told that the same equivalent is used for both chemical, and nuclear bonding  Had there been further information, such as a usage sample, a definition or a note, we would have been asked whether we wanted to see it with the question *More?* Answering *yes* would show the information to us.

After this first use, Target assumes that we are translating from English to French Notice that it does not ask us the From and To questions:

```
    Term: bond
  bond     Chemistry: Theoretical Chemistry;    (ch4)
           The Nuclear Industry: Hazards;    (at8)
           Financial Affairs - Taxation - Customs;    (fi)
    Select Code: fi

  bond     emprunt (FR)

           Financial Affairs - Taxation - Customs;

  Reference Terms:     government bond
    Term:
```

We can override the assumption by typing *all on one line* the term name, the source language and the target language. Here we check on the equivalent just obtained:

```
     Term: emprunt fr en

   emprunt   bond (EN)


             Financial Affairs - Taxation - Customs;

     Term:
```

English and French now become the new anticipated source and target languages, respectively.

## 3. The TERMIN Program

TERMIN is the name of the program used by *terminologists* to augment and maintain the dictionaries. Clearly the facilities in TERMIN must be more varied than the simple retrieval facility which is TARGET; the facilities in TERMIN are accessed through a set of *commands*:

| | |
|---|---|
| Help | (to get instruction) |
| Create Entry | (to enter a term) |
| Exit | (to leave TERMIN) |
| Retrieve Entry | (to get term and term information) |
| Edit Entry | (to revise or augment entry) |
| Delete Entry | (to delete an entry) |
| List Contents | (to list terms in one language) |
| Target | (to get target language equivalent) |
| Record Transaction | (to record session) |
| Term Hardcopy | (to make hardcopy of a term) |
| Dictionary Hardcopy | (to make hardcopy of a whole dictionary) |
| Regenerate | (to correct faulty dictionary) |
| Recover Space | (to recover space used by deleted and updated entries) |
| Do | (to execute a command file) |
| Start | (to start using an Option) |
| Stop | (to stop using an Option) |

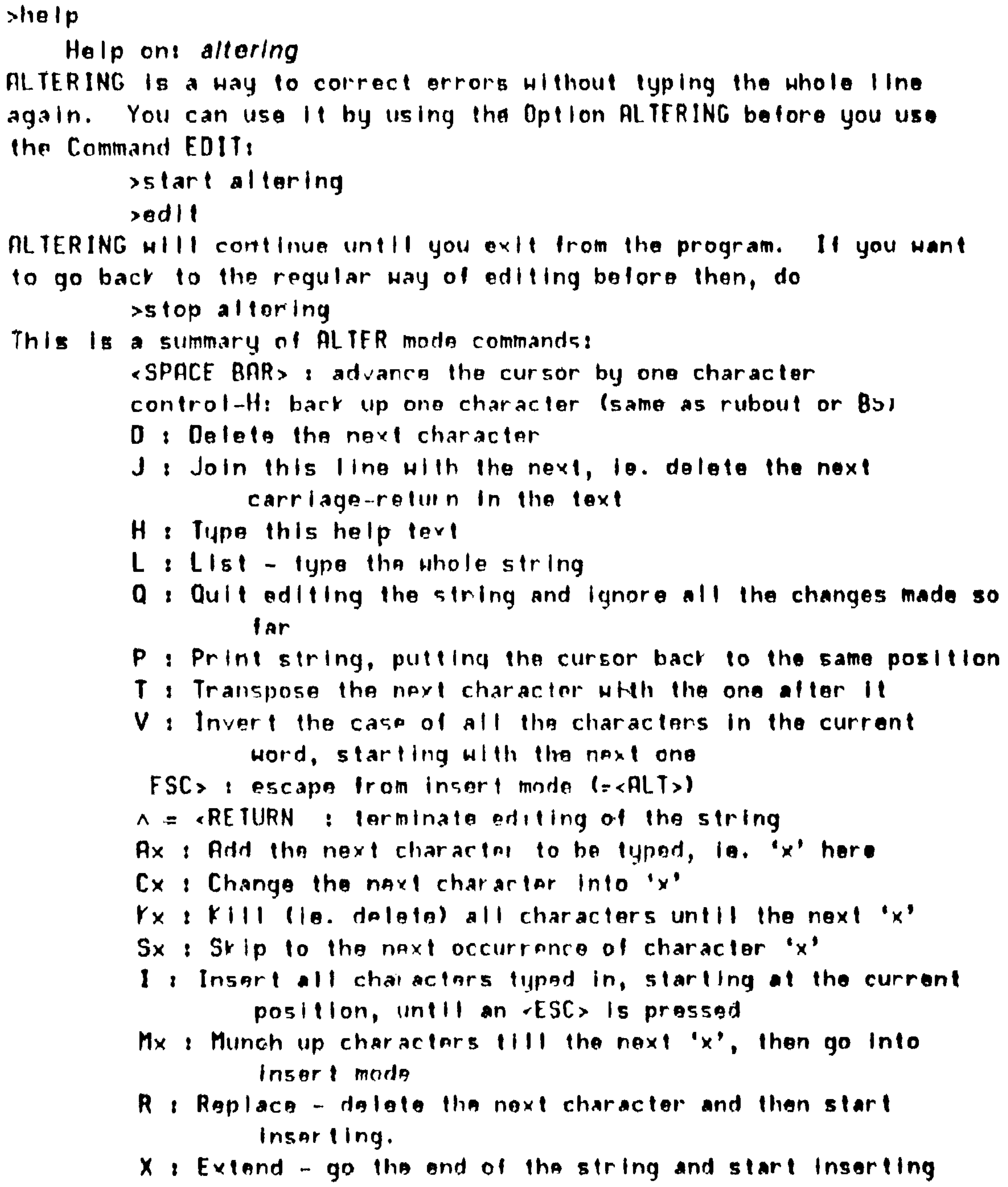
The TARGET program essentially just repeats the TERMIN command of the same name

The commands are the large functional units in terms of which we interact with the dictionary. We use them when entering new terms (*Create Entry*), editing existing terms (*Edit Entry*), printing dictionaries[3] (*Dictionary Hardcopy*), etc. -- and in fact for everything we do with the dictionary. A short description of the use of each command follows

[3] An example page is reproduced in an Appendix, exactly as it was printed by our Xerox Graphics Printer

### 3 I Help

This provides on-line access to written comments on various aspects of the use of the TERMIN program by terminologists. All of the twenty or so texts which may be accessed in this way were written by the terminologists (who also ordered the list of commands as above and provided the brief description beside each one).



```
>help
    Help on: altering
ALTERING is a way to correct errors without typing the whole line
again.  You can use it by using the Option ALTERING before you use
the Command EDIT:
        >start altering
        >edit
ALTERING will continue until you exit from the program.  If you want
to go back to the regular way of editing before then, do
        >stop altering
This is a summary of ALTER mode commands:
        <SPACE BAR> : advance the cursor by one character
        control-H: back up one character (same as rubout or BS)
        D : Delete the next character
        J : Join this line with the next, ie. delete the next
                carriage-return in the text
        H : Type this help text
        L : List - type the whole string
        Q : Quit editing the string and ignore all the changes made so
                far
        P : Print string, putting the cursor back to the same position
        T : Transpose the next character with the one after it
        V : Invert the case of all the characters in the current
                word, starting with the next one
         ESC> : escape from insert mode (=<ALT>)
        ^ = <RETURN  : terminate editing of the string
        Ax : Add the next character to be typed, ie. 'x' here
        Cx : Change the next character into 'x'
        Kx : Kill (ie. delete) all characters until the next 'x'
        Sx : Skip to the next occurrence of character 'x'
        I : Insert all characters typed in, starting at the current
                position, until an <ESC> is pressed
        Mx : Munch up characters till the next 'x', then go into
                insert mode
        R : Replace - delete the next character and then start
                inserting.
        X : Extend - go the end of the string and start inserting
```

Most of the texts provided in this way give hints and reminders in an informal manner, rather than a detailed sequence of instructions on how to use the command.

### 3 2 Create Entry

This command is used to create new entries in the dictionary for terms which have not yet been entered Let us suppose that we have prepared a fiche specifying the equivalent in French of the English term *bond* appropriate for the field of Chemistry. Space constraints prevent a detailed description of the interactions leading to the entry of this term into the dictionary, but the following is a trace of the process:

```
>create
  Term: bond
  Language: en
  Field Classification: ch4
  Equivalents: liaison
    Language: fr
  Grammatical Categories: nou
Select  Usage Sample
        Definition
        Reference Terms
        Note
        Synonyms? reference
  Reference Terms: "bonding energy"

bond      liaison (FR)

          CH4

Reference Terms:     bonding energy

nou

! New term "liaison" being entered in "FR"%
```

TERMIN does what it can to save us effort; here it has generated a small fiche to contain each equivalent which does not already exist. The minimal fiche it generated for *liaison* is this:

```
liaison   bond (EN)

          Chemistry: Theoretical Chemistry;
```

This will usually need to be augmented with other information, such as the grammatical category This is done by using the *Edit Entry* command (see below).

### 3 3 Exit

This is how we leave TERMIN:

```
>exit

EXIT
```

The user is now no longer using the TERMIN program, but is using the TOPS-10 monitor.

### 3.4 Retrieve Entry

This command is used to print whole entries at our terminal. Here is *bond*:

```
>retrieve
  Term: bond
  Language: en

bond      liaison (FR)

          Chemistry: Theoretical Chemistry;

Reference Terms:      bonding energy

nou
```

There is only one Equivalent, one Field Code and one Reference Term for *bond* at this point. More complex entries, with more of the optional term information, will display all that information as well *Retrieve Entry* shows *all* the information there is for *bond*; only portions may be displayed with the *Target* and *Edit Entry* commands.

### 3 5 Edit Entry

There is a number of reasons why it may be necessary to edit an entry. Perhaps the person who entered it mad a typo, perhaps it is necessary to extend an entry to include, say, a Usage Sample (or perhaps to replace the old one with a better one) or perhaps a new fiche must be entered for an existing term.

*Update the Field Code:* Suppose that further investigation of the term *bond* has revealed that the same French equivalent is also used for the "bond" holding the nucleus of atoms together as for the electronic bond which keeps different atoms together. Had this been known when the original fiche was created, it would have contained *two* field codes, *ch4* and *at8*. We need to update the original entry.

*Add a New Fiche:* The term *bond* is also used in financial and commercial circles (among others[4]), but here the equivalents in French are *not* liaison So we can enter a new fiche for this term.

*Deleting a Fiche:* It is occasionally necessary to delete a fiche within a term, but not the whole term.

*Edit Entry* is used to accomplish all these functions. It is the major tool used in maintaining the dictionaries.

## 3 6 Delete Entry

This is used to delete entire entries:

```
>delete
  Term: "blast off valve"
  Language: en
>
```

It will no longer be accessible

## 3 7 List Contents

This gives an alphabetical list of terms in a specified language. We can limit the list by specifiying the first one or two letters of the first and last term. It may be aborted by typing @ at any time, and will return us to the command level immediately.

Let us get a list of the English terms from *bo* through *c*:

```
list
 Language: en
 From Letter: bo
 To Letter: c
        'body centered cubic
        'bold'
        'bond'
        'bonding energy'
        'boom'
        'bore'
        'boring bar'
        'boundary position'
        'bracket'
```

[4] Note that there is nothing to stop different fiches for the same term sharing field codes A different fiche is needed whenever and only when, the equivalents are distinct

'breakdown'
'breaking'
'bridge connection'
'broker'
'brokerage'
'buffer'
'burden'
'business'
'business profit'
'butterfly valve'
'by means of'
'by-pass valve'
'by x-ray diffraction'
'capacity'
'capital'
'capital and reserves'
'capital gain'
'capital goods'
'capital-intensive'

### 3 8 Target

Target is the command (or program) we use to find equivalents for a term The TERMIN command *Target* works identically to the *TARGET* program (described above) with the exception that in TARGET if an abort character @ is typed in answer to the *Term:* prompt, the progam exits; in TERMIN, an @ at this point gets us back to the command prompt >

### 3 9 Record Transaction

This command is used to keep a record of the interaction between the program and the user It is used for studying how users interact with the system in order that it may become better tailored to their needs (All of the examples in this document were drawn directly from records made in exactly this manner). Each interaction between the system and the user may also be timed in the record by using the *Timing* option. This provides an extra tool for studying how the system is used in practice. It is also useful for some purposes to be able to annotate a record while it is being produced. TERMIN will ignore any line beginning with a semi-colon:

```
>; This shows that comments get into the record
```

### 3 10 Term Hardcopy

This command is used to print a specific term in the same format as that of the terms in the Appendix

The file just generated contains formatting information as well as the term itself. It must be *compiled* by the PUB document formatter

### 3 11 Dictionary Hardcopy

This is like *Term Hardcopy*, but the program will select the terms for us and put them in alphabetical order. We can choose the language, the initial two letters of the first and last terms, and can also restrict the fields for the terms, by specifying a set of field codes · This allows us to make selective microglossaries, choosing perhaps just those terms relevant to to the Petroleum Industry or Medicine. Here we illustrate obtaining an entire dictionary One page of it is reproduced as an Appendix.

```
dictionary
 Source Language: fr
 From Letter: a
 To Letter: z
 Restricted Fields? no
..............................................................................
..............................................................................
............................
Please see TEMP:FRENCH.PUB
```

This file must be formatted with the PUR document compiler. The title page of the hardcopy will describe any limits we may have imposed on its contents.

### 3 12 Regenerate

This command is used to re-establish the links between the various files which contain the dictionary. They can become incorrect when the computer crashes while certain operations are being performed, or when there are problems with the system.

### 3 13 Recover Space

When a term is stored after having been *Edited*, a new entry is made for the new version of the term The old entry is, however, still there, and hence takes up space. The same is true in the case of the *Delete* command. When a term is deleted, it actually only becomes inaccessible -- and so it is taking up space. Every so often this "wasted" space (which actually provides the potential for some backup)· is recovered using this command; it compacts the dictionary

### 3.14 Use of Commands

As can be seen from the foregoing, to use a command we type its name (e.g. ***help***). Upper case and lower case are equally acceptable. The program will then begin to prompt us for the further specifications neccessary to carry out the command. We may *Type Ahead* the responses to those questions, in which case the prompt is not given Help is usually obtainable by typing ? and any command can be aborted by typing @ in response to any prompt

### 3.15 Conveniences of Interactions with User

One convenience in TERMIN is that it is often not necessary to type the whole of the response to a prompt In fact, all we need to type is an unambiguous abbreviation, thus:

```
    retr
```

for *Retrieve Entry*, for instance. If we type an ambiguous abbrevation, the system can help us out:

```
>re

? re is ambiguous:
        Retrieve Entry          (to get term and term information)
        Record Transaction      (to record session)
        Regenerate              (to correct faulty dictionary)
        Recover Space           (to recover space from deleted terms)
>reco

? reco is ambiguous:
        Record Transaction      (to record session)
        Recover Space           (to recover space from deleted terms)
```

We can also be assisted when we make typing mistakes:

```
>lslt
  ... did you mean LIST CONTENTS              (TO LIST TERMS IN ONE LANGUAGE) ?yes
```

As it happens, users of the system often find this kind of help confusing initially and so there is an Option, called *Helpful*, to control it. Initially this option is turned off, but it can be turned on simply by typing:

```
    start helpful
```

TERMIN can often anticipate the answer to one of its questions. For instance, if we have just *Retrieved* a term and then we issue an *Edit* command, the chances are that the term we just retrieved is the term we want to edit. The *Defaulting* Option makes similar assumptions To use this option, we type:

```
>start defaulting
```

and then we can utilize it:

```
>retr liaison fr

liaison   bond (EN)

          The Nuclear Industry: Nuclear Energy;
          Chemistry: Theoretical Chemistry;

Reference Terms:      l'energie de liaison

nou mas

>edit
  Term: [liaison]
  Language: [FR]
```

(Note that we have accepted the default by simply striking the <RETURN> key.)

When accessing terms, we need not specify accents (unless they distinguish between two terms) This is a convenience to be used when *accessing fiches* and their contents: when we type the *text* in a fiche we must use the correct cases and accents.

## 4. Some Theoretical Issues

While designing the representation of terms and their interconnections within the dictionaries, researchers at the Target Project discovered some difficulties with some of the methods adopted by other terminology banks. This section is a brief presentation of some of them.

When a sense in one language is translated by a sense in another, they are said to be *equivalents* of each other This is what is crucial to the translation task, and what is under discussion in this section is *the representation of equivalence between senses.*

Two methods for doing this will be compared. In one, called the *Intermediate Concept Space Representation* (ICSR), there is held to be a language-independent set of concepts which are realized in differing languages each with the appropriate term. Figure 1 shows some equivalents between Schaufel (German), Aube (French) and Vane (English), which are appropriate in the field of Astronautics. It must be noted that what we have called "senses" above are represented by their term-names only both in the figures and in the text; this device is used merely for clarity of exposition.

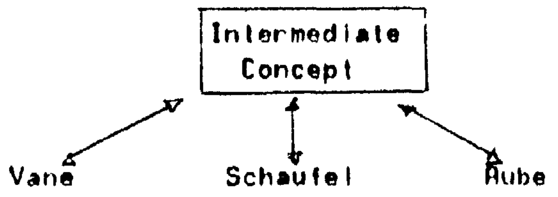


Figure 1: ICS Representation of Vane≡Schaufel, Schaufel≡Rube & Rube=Vane

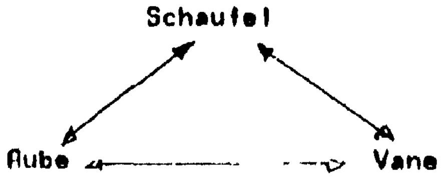


Figure 2: DE Representation of Vane≡Schaufel, Schaufel≡Rube & Rube≡Vane

In an alternative method, the *Direct Equivalent Representation* (DER), there is no need for such an intermediate concept space. Each sense accesses its equivalents directly, as shown in Figure 2.

ICSR is attractive because it offers a conceptual elegance absent from DER -- there is a universe of objects, each of which has a different linguistic representation in each language. This is a hypothesis about the nature of language which is known to be a misleading oversimplification for everyday usage, but its proponents presumably hope that it could turn out to be sufficiently true for the more restricted domains of specialized terminology

The two major ways of comparing these two alternatives are (1) in terms of the computer space taken in holding them and time taken in retrieving them, and (2) in terms of their adequacy when the dictionary must be modified The former indicates that under some circumstances ICSR can be cheaper in terms of space, but the latter shows that DER is resoundingly more adequate for the task of representing equivalence, and thus was chosen for Target.

## 4.1 Space and Time Analysis

Each of the lines in Figures 1 and 2, whether between an Intermediate Concept and a sense (Figure 1) or between two senses (Figure 2), represents what is called in computer parlance a *pointer*. Pointers need space in the computer and -- perhaps more importantly -- take processing time when used. Thus to get from "*vane*" in English to its German equivalent (*Schaufel*) requires the use of two pointers in Figure 1 (ICSR), but only one in Figure 2 (DER) This greater efficiency of DER is true for all pairs of equivalents.

Differences between DER and ICSR so far as space is concerned depend upon the number of languages in the multilingual dictionary. If there are *N* languages attached to an Intermediate Concept, there will be *N* pointers, one to each In the worst case for DER, each of the *N* will have a pointer to all of the *(N-1)* others, requiring $(N*(N-1))/2$ pointers Since there are three languages in Figures 1 and 2 (English, French and German), *N* is 3 and hence there is no advantage for either DER or ICSR The larger *N* becomes above 3, the greater is the advantage for ICSR; for instance, if *N* is 5; then in the worst case DER requires twice as many pointers as ICSR and if *N* is 7, DER requires 3 times as many in the worst case This worst case occurs when equivalents are present in the dictionary for *every* language, which may not be so in practice, especially while a dictionary is being compiled In the most favorable case, *N=2* and

DER has the advantage by a factor of 2. Dictionaries prepared for American use may often be English↔X, so that *N-2* and DER has a space advantage as well as the time advantage demonstrated above.

## 4.2 Modifiability

Irrespective of these considerations, a dictionary must remain functional while it is incomplete. To be realistic, it is probably uncommon for a dictionary to be "finished", and all automated dictionaries must be built incrementally, equivalent by equivalent. There are important differences between ICSR and DER, both in processing when equivalences are entered and when using an incomplete dictionary.

We need only consider as simple a case as the structures of Figures 1 and 2. Let us suppose that none of the equivalences *vane=Schaufel, vane=aube* and *Schaufel=aube* have yet been inserted in the data-base and they must be inserted. After entering the equivalence *vane=Schaufel*, ICSR will look like Figure 3a and DER like Figure 3b:

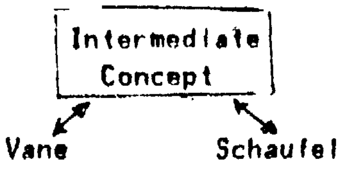


Figure 3a: Vane=Schaufel (ISCR)

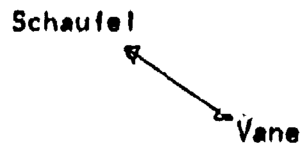


Figure 3b: Vane≡Schaufel (DER)

(Note that this is a case where *N-2* in the space analysis above, and so -- at this point -- ICSF has two pointers and DER only one.)

Now the equivalence *vane=aube* is to be inserted. With ICSR, the terminologist has no choice but to determine whether *aube* is equivalent to *Schaufel*. If they are, then Figure 1 is obtained. But suppose that they are not; then Figure 4a would be obtained:

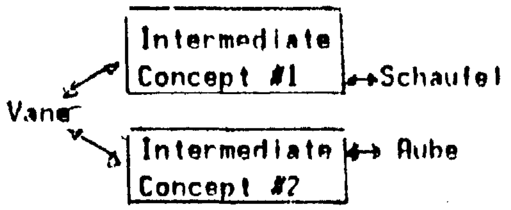


Figure 4a: Vane=Schaufel, Vane=Aube, but Schaufel≠Vane

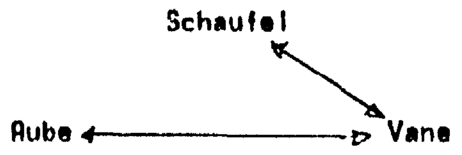


Figure 4b: Vane≡Schaufel, Vane≡Aube, but Schaufel≢Vane

Note that for ICSR, the terminologist is forced to check *every existing equivalent* of a term when adding another, a procedure whose complexity increases exponentially with the number of languages. The competence of the terminologist must extend to *all* the languages in the database. With DER, on the contrary, no more need be done than simply adding the new equivalence as in Figure 4b. *Only if there is a known equivalence* between *Schaufel* and *aube* will the situation shown in Figure 2 be obtained.

A tempting, but incorrect, solution to this problem for ICSR is to ***assume*** that *aube=Schaufel*, producing Figure 2 whether or not it is actually appropriate. This is a kind of risky and uncontrolled inference which ICSR can naturally force upon the user. There may be subtle differences in meaning between languages, yet ICSR forces transitivity of the relation of equivalence between all langauges. There is an alternative approach within ICSR, in which this is not the assumption, but this will lead to precisely the proliferation of senses which ICSR was designed to avoid. Furthermore, the simplification of intermediate concepts which are found to be redundant will be a complicated procedure.

In summary, the point so far is that the addition of an equivalence is a drastically more complex procedure in ICSR than in DER, and secondly that ICSR requires the terminologist to be as multilingual as the database, while DER does not. A further point may be made which concerns *inferencing*.

"Inferencing" means finding a near equivalent when an actual equivalent is not immediately obtainable. Of course, it is to be hoped that an automated dictionary will usually have an *immediate* answer to a user-request for an equivalent, in the sense that the requested equivalent has previously been entered explicitly. However, situations will occur where an immediate answer is not available. In that case, some form of *inferencing* may help. With ICSR, that inferencing has already been done in setting up the intermediate sense by means of the assumption above, and thus the information that it is an inference is lost at retrieval time. With DER, the pointers must be followed through explicitly and thus the system can report to the user the extent of the tentativeness of the derived near equivalent.

The disadvantage for the Intermediate Concept Space Representation, then, is that on the one hand finding an equivalent always takes *two* pointers, while Direct Equivalent Representation needs only *one*, and on the other -- more importantly -- DER is more able to represent nuances of meaning across languages and incomplete states of the dictionary. Hence Target uses the Direct Equivalence Representation for term equivalence.

***fraise a fileter*** thread mill (EN)
Iron Steel Industries: Machines and Apparatus;
nop

***fraise conique*** countersink (EN)
Iron Steel Industries: Machines and Apparatus;
nop

***fraise-mere*** hob (EN)
Iron Steel Industries: Machines and Apparatus;
nou fem

***fraises*** milling cutters (EN)
Mechanical Engineering: Machines for Moving and Processing Materials;
Iron Steel Industries: Machines and Apparatus;
nou fem plu

***frais fixes*** fixed costs (EN)
Economics;
Reference Terms: frais
nop plu
Usage Sample: ...une nouvelle augmentation des frais fixes... [Kre4376]

***frais generaux*** overhead charges (EN)
Technology and Industry In General;
Financial Affairs - Taxation - Customs;
nop
Usage Sample: ... les frais generaux (frais administratifs, de personnel et de gestion des polices d assurance). [SG86/77]

***frittage*** fritting (EN)
Iron Steel Industries: Pig Iron Production;
Mining: Preparation and Refining of Raw Materials From Mines;
nou mas
Definition: roasting process in steelmaking [fr78]

***frottement*** friction (EN)
Iron Steel Industries: Stress-relieving Deformation;
General Terminology;
nou mas