[From: *Translating and the Computer 14*. Papers presented at a conference… 10-11 November 1992 (London: Aslib 1992)]

ONLINE CORRECTION AND TRANSLATION OF INDUSTRIAL TEXTS

Gert van der Steen (part I) and Bert-Jan Dijenborgh (part II)

Volmac Lingware Services, Postbox 2575,3500 GN Utrecht, The Netherlands

ABSTRACT

Why is the automatic translation of simple texts still such a difficult process? Industrial manufacturers are anxiously waiting for this facility to arrive and, with the European market widening, this need will only increase. There is a growing demand for online automatic error-free translations into various European languages. In view of this development, Volmac Lingware Services (LS) has developed a method for handling textual problems.
The paper consists of 2 parts. Part I describes the services and tools which LS has developed for the automatic correction, standardization and translation of texts. Part II discusses in more detail the linguistic aspects of the services.

PART I: THE LANGUAGE EDITOR: MOTIVATION AND RESULTS

1. INTRODUCTION

For most companies, the production of texts is a necessary evil. The texts are a "sideline" to a main product which requires all the attention there is. The customers of these companies know this, too. For this reason, the United States government decided to allow new types of aircraft to fly only after the complete documentation is available to the airline company in question. New drugs are not allowed to be sold either without instructions containing correct information.
These are just two examples of so-called informative texts the only function of which is to convey information as clearly as possible, and preferably in the user's mother tongue.
For many companies, the production of these texts causes growing concern. The increasingly complex machinery used requires more documentation, which employs terminology known only to specialists. Moreover, the progressive linking of information systems invites a more intensive exchange of texts. Under these conditions it is easy to lose sight of who has written which text in which location. This calls for the condition that text must be correct from the source onwards. A second requirement is that it must be possible to translate the text.

2. TEXTS OF AN INFORMATIVE NATURE

Below, a number of examples are provided of the production and the use of informative texts. A car manufacturer, an airplane manufacturer, an insurance agent and a software producer are discussed.

A car manufacturer. The car plants are spread over North America and Europe. Parts produced by different factories may be used for the production of one car. The writers of technical manuals work in

special publication departments and use ordinary word processors. The texts are structured in accordance with specific instructions.
The production of a revised version of a manual is a time-consuming affair. Many separate files have to be collected and writing is done both in German and English.
The readers of the manuals are car mechanics in America and Europe. For the European mechanics the handbooks need to be translated into more than ten languages. Translation is done by hand in a central location.
In the near future, the car manufacturer intends to streamline the production of manuals by using a so-called "markup language". This places markings in the text such as "chapter", "title", "paragraph", but also "material number", "warning". In this way texts are prepared for filing in a database. This will make it possible to supply marked text sections from various sources and not just from the publication departments.
This means that instead of complete texts, only sections of these will be translated. It is therefore important to keep an eye on the standardization of terminology and spelling. The plants, situated in distant locations, have a fairly independent character, which allows for some databases to be decentralized. In the long term, however, car dealers will have to be able to consult these databases and to receive relevant texts in their own language.

An airplane manufacturer. The factories of the airplane manufacturer are situated near to one another. In this case, the writers of technical manuals are also united in special publication departments and then fill out forms by hand. An external agency enters the forms into a computer. Mutations are also processed by means of forms.

Translation is not necessary in the airline industry. There is an agreement to use so-called "Simplified English" in maintenance manuals. Maintenance mechanics of airline companies are expected to have a command of this simple type of English. The instructions for Simplified English consist of a list of permitted words and a number of do's and don'ts with regard to sentence structure. The use of Simplified English is beginning to spread. Technical authors can follow courses in Simplified English. However, there is no strict checking as to whether the manuals meet the requirements.

The structure of maintenance manuals for the airline industry has to conform to the standards it has formulated. In the near future these will prescribe the use of the markup language SGML. For this reason, the airplane manufacturer has decided to modernize the production and updating of the manuals by using special word processors. These word processors check on-line whether the entered text meets the standards of SGML concerning structure. As with the car manufacturer, the expectation is that in future the production of texts will be decentralized to a greater extent and that higher standards will have to be met by the language used and the correct entry of information. Technical texts contain few redundant elements. This means that essential information is included in one place only. This is why carelessness during the entry of the text may later prove fatal.

An insurance agent. Each day, salesmen leave the main office in order to sell insurance policies. To this end they use forms in which both numbers and text must be entered. Text is used because there are so many exceptions to the foreseeable situations that it is impossible to include separate fields on the form for this purpose. Therefore, the information in the text is of vital importance.
Back at the main office, the salesman enters the data into the computer. The text cannot be processed automatically, but is subjected to further interpretation by staff in other departments. During this process, questions arise concerning the terminology used and ambiguities in the text which require feedback with the salesman and, if necessary, the customer.

A software producer. Software developers may produce software tools which customers subsequently use to write their own application programs. Volmac writes such application programs itself, as it did when developing the MODIX information system for the fashion retail industry. This program

manages data about the textile industry and its products. The system has an extensive support function. The fact that the system can be used throughout Europe places demands on the language used. A Parisian must be provided with information in French; his colleague in Milan who pays him a visit must be able to read the same text in Italian.

These kinds of tools require purpose-made documentation. Each software producer is familiar with the problem of providing accurate documentation on time. This is why Volmac is experimenting with development techniques which allow for the simultaneous production of software and documentation, so that the software developer is also the person who produces the documentation. Therefore, completeness, consistency and language need to be checked.

3. FEATURES OF INDUSTRIAL LANGUAGE: DEFINITION OF THE PROBLEM

The above examples share a number of features. The texts are all of an informative nature. Although the sentence structure is relatively simple, it may vary with the writing styles and personal preferences of the writers. Therefore, standardization is called for. A distinctive feature is the fact that the words and terminology used differ strongly for each application area. The handling of texts is usually part of a larger process which may be in various stages of automation. People are accustomed to having standards for such procedures prescribed. Sometimes, however, there is no check on whether these are met.

Due to decentralization of the input, an increasing number of staff is involved in entering text. By dividing the text into sections by means of markup languages, updates can be entered more efficiently. The time gained in this manner is lost, however, if the checking and translation (if any) of the text are not performed automatically, but continue to be carried out by staff.

Translation is a characteristically European problem. This process is in various stages of automation as well. In many cases, the computer is used as an electronic dictionary which contains the terminology employed by the company. Sometimes a translation program is used which provides rough translations which always require refinement.
With the European market widening, translations will need to be made into an increasing number of languages. At some stage it may become impossible for a company to maintain a large translation department.

Summing up:
- The number of people entering text is increasing.
- Texts are increasingly integrated into larger information systems.
- The sentence structure of a text is simple, but may vary due to personal preference.
- The terms used may be highly specialized.
- An increasing number of texts needs to be translated.

Therefore, the following functions are required:
- Automatic correction, simplification and standardization of terminology, spelling and sentence structure whenever possible;
- Automatic translation into as many European languages as possible.

Unfortunately, however, the computer programs which are needed to perform these functions are still at an early stage of development.

4. WHY ARE THE LINGUISTIC PROBLEMS STILL UNSOLVED ?

Why is a computer unable to handle natural language as accurately as numbers?
The reason is that a correct handling of language is as difficult as translation, while the automatic translation of everyday speech poses insoluble problems [Van der Steen (1), Eikelenboom (2)].
Briefly stated, the reasons for this are the following:

- Natural language has an almost unlimited vocabulary.
- In everyday speech, words and sentences may have different meanings ("ambiguity").
- Translation requires knowledge of the meaning of words, sentences and even entire texts and knowledge of "the world".
- The text to be translated usually contains errors.

As a result, the current spelling, grammar and style checkers are inadequately equipped to perform an error-free correction. An additional problem is that the automatic translation of text by a computer program which is designed for normal language takes a long tune.

Therefore, authorities in the field of automatic translation have reached the conclusion that it will only be possible to achieve results if the linguistic resources (vocabulary and grammatical structures) are restricted. In a small number of constructed systems good results have been achieved in this manner. In Canada for instance, weather reports are automatically and correctly translated from English into French and vice versa. A number of translation system designers try to follow this approach. However, for the time being the problem remains that it is difficult to adapt these systems to local conditions.

5. A SOLUTION FOR THE AUTOMATION OF INDUSTRIAL LANGUAGE

LS has developed its own views on the automation of industrial linguistic resources and supports these views with specially developed software. This software allows the systematic definition of linguistic resources for each application, so that automatic correction, standardization and translation into various European languages become a reality.

LS's view on the automation of industrial language

Over the past few years LS has carried out a number of research projects the results of which were published in the Journal of Software Research. [Eikelenboom (2), Kusters (3), Van der Steen (4)].
Summarizing, the following conclusions may be drawn from these articles:

- If the linguistic resources are restricted, there are more opportunities for correct handling.
- Restriction of linguistic resources may concern:
  - choice of words;
  - sentence structure;
  - subject of the text.
- The restriction of linguistic resources must be performed with care, while a good author support is indispensable.
- The translation process can be controlled by applying the following measures:
  - restrict and standardize the vocabulary;
  - assign only one meaning to sentences;
  - support the author by means of a special editor.
- Rapid translation is possible.

Considerations. During discussions about the restriction of linguistic resources, the fear is sometimes expressed that a certain freedom will be lost. Psychological research, however, indicates that people are willing to modify the language they use if they perceive the advantages of doing so. In important situations people intuitively produce shorter sentences.
Informative texts are written for a particular purpose: the conveying of information in as clear a manner as possible. They are not written to convey emotions or ideals.

The objective of the discipline of Human Computer Interaction is to allow the user as much freedom as possible. However, it prevents the user from making errors by offering a number of correct options with explanation.

What counts is the attitude of people who actually produce informative texts. So far experiences have mainly been positive.

In the examples mentioned above, everyone was intent on writing as clearly and unambiguously as possible. Restriction and standardization of linguistic resources will certainly produce texts which are easier to read and understand. It is even in the public interest for texts in maintenance manuals to have only one possible interpretation.

Most other objections to the restriction of linguistic resources often concern its automation. Actually these are related to the social aspect of automation and therefore do not contribute to the reflections on the pros and cons of restrictions of linguistic resources.
LS has reached the conclusion that the restriction of linguistic resources is the only way to achieve good automatic translation. This has been the point of departure for the automation of the handling of industrial texts and the development of software for the support of writers.

6. THE RELATION BETWEEN DOCUMENT AND LANGUAGE HANDLING IN INDUSTRY

From the examples of industrial language usage we draw the conclusion that the automation of language handling may follow in the footsteps of the automation of document handling.

At present, the markup languages mentioned above such as SGML and ODA are being used for this purpose. They extend the possibilities for text manipulation for the storage and the retrieval of logical parts of documents in databases.
In addition, they make it possible to postpone the assembly of combinations of document sections until the last moment.
This makes "on-demand publishing" possible. Individual words in the text may be marked for the purpose of HyperText applications, thereby increasing the accessibility of the documents.

The definitions of the document structures permitted are recorded in so-called DTDs (Document Type Definitions). With the help of "General Entities", character sets and abbreviations may be stored for the benefit of terminology. The management of the DTDs and the General Entities can be handled by means of repositories.

The controlled entry of SGML-structured documents is preferably performed by means of an editor. SGML and ODA take care of document structure, including chapter, title, paragraph and list. The elementary level is the text. Now the language handling can begin.
The restriction of linguistic resources is performed by grammars, lexicons (dictionaries containing information which is geared to the grammars) and thesauruses (a list of words and their synonyms). These may be added to the repository for document handling.

The linguistic check and standardization of the text also has to be performed by an editor. When document and language handling are combined, the obvious choice is to combine all checking

functions in one editor. Now "on-demand publishing" will also make it possible to include "on-demand translation".

Perspectives. The linguistic structure of each sentence in the text is absolutely fixed. This makes possible a precise handling of texts, from top level to character level. In addition to on-line translation, for the first time the retrieval of information on meaning level also becomes possible. The precision with which this can be done will approach that of numerical databases.

7. TASKS OF LS

LS has assumed the following tasks:

- the development of working methods for the analysis of linguistic resources for different application fields and for the construction of grammars, lexicons and thesauruses (the so-called "lingware") for these application fields;
- the development of tools to support these working methods;
- the development of a working method supporting the writers who use restricted language;
- the development of a correcting editor, including an effective human/computer interface;
- the development of a rapid translation program.

This development has been linked up as far as possible with existing methods and techniques in computer science and computational linguistics. It also links up with the functionality of tools from the so-called "language industry". [Obermeier (5)].

The work which LS has done in the field of automation of industrial language usage belongs to the language industry. Its accepted activities are:

- designing interfaces containing natural language programs;
- automatic translation;
- text handling and text representation;
- text generation;
- speech recognition and generation;
- language editors for authors.

Each of these activities has its own products which, however, have little in common with one another.

The activities of LS mainly concern text handling, automatic translation and language editors. LS clusters these activities in order to produce a certain synergy. The developed tools are versatile and can be integrated into other tools and systems.

8. LS'S LINGWARE AND SOFTWARE DEVELOPMENT CYCLE

LS follows a phased approach for the development of systems and subsystems for document and language handling in industry. These phases parallel the phases of a normal software development cycle. They are listed here, together with a short annotation of the activities which will be discussed in more detail later on in this paper.

Information Analysis. In this phase the type of language usage is investigated: the required languages and translations, the domain of the texts, the type of the required corrections and standardizations.

Functional System Design. The functioning of the linguistic (sub)system(s) within a larger system is determined in this phase. LS uses a blueprint for a subsystem for author support and translation, which is shown in figure 1.

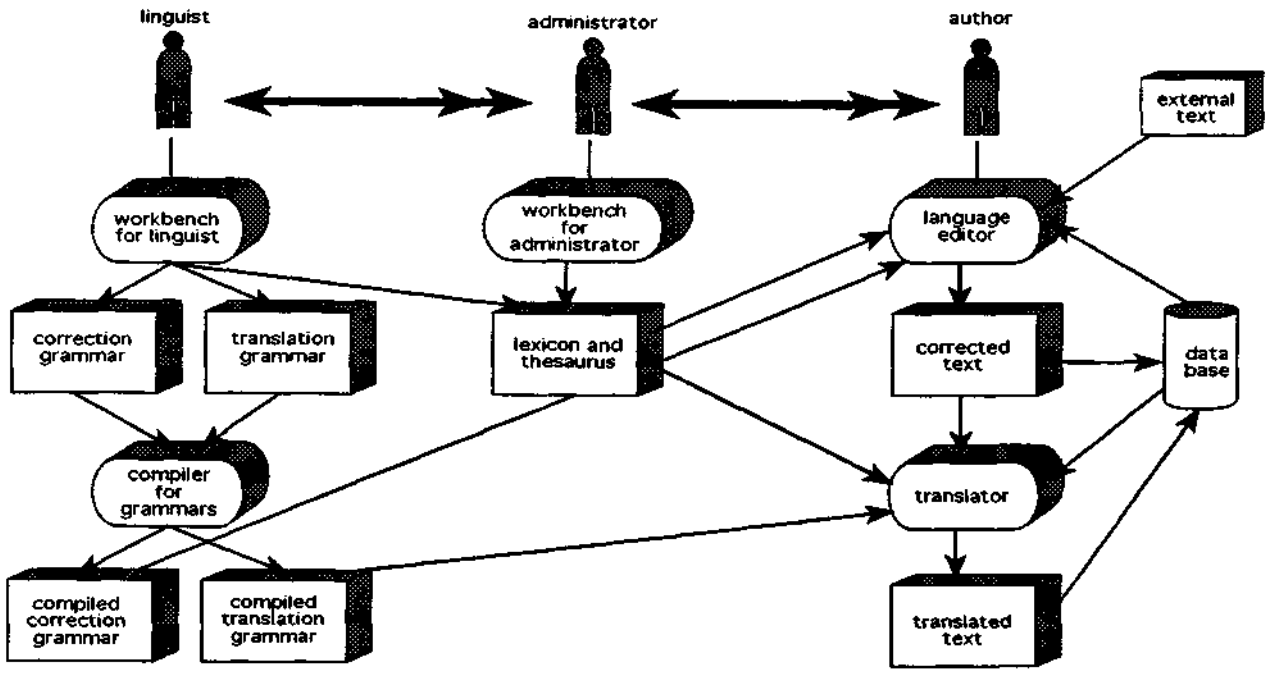


Figure 1 Blueprint for a subsystem
for author support and translation

This blueprint presupposes on-line processing. However, LS also develops systems for batch processing.

Technical System Design. In the design phase LS uses already developed software tools which can be used as complete subsystems or which can be customized to the desired functionality. These tools can be found in the blueprint of figure 1 as: Language Editor, Translator and Workbench for Administrator.
Also shown are the Workbench for the Linguist and the Compiler for Grammars. They are used for the analysis, design and realisation of the lingware. The Workbench for the Administrator is meant for the administrator within the organisation of the client who will have to update the lexicon and the thesaurus.

Realisation. In the realisation phase the software and lingware are coded, compiled and tested. The formalism of the lingware is strict. In fact, it acts as a kind of sophisticated programming language. The lingware is compiled in the same manner as a computer program and can be tested and debugged like any normal computer program.

Implementation. For the lingware component, the introduction in the organisation requires special activities. A writing guide for authors has to be developed. Courses have to be given for authors and administrators. A good user acceptance requires a fast feedback to initial problems of authors.

Features of the Language Editor and Translator

The objective of LS is to first correct and standardize texts at the source with the Language Editor and subsequently to translate them with the Translator.
The tasks of the Language Editor include checking and support by means of:

- correction of incorrect spelling and punctuation;
- changing the word choice by means of the thesaurus;
- improving the sentence structure;
- communication with the writer in case of doubt.

The task of the Translator is to automatically translate the total amount of texts produced by the Language Editor without any human intervention (in batch).
The previously constructed grammars, lexicons and thesaurus feed the Language Editor and the Translator.

Below we provide a concise functional and analytic subdivision of the features of the Language Editor and the Translator.

Functional:
- Correction is performed on-line, while the correction is provided as quickly as possible.
- Possibly after help from the writer all corrected sentences are guaranteed to be error-free.
- The corrected sentences are translated without any errors.
- At present, the translation speed on a MAC II or a PC386 is approximately 15 words per second.

Analytic:
- The basic formalism of the grammar is comparable to that of attribute and affix grammars.
- The formalism has been extended:
  - from parsing grammar to correction grammar;
  - transfer rules have been integrated into the grammar.
- If corrections cannot be solved in an exact manner there is a switch back to general error correction techniques after which the attention of the writer is requested.
- The Language Editor and the Translator share a software "engine". The engine integrates the information from the lexicon, the grammar and the thesaurus.
- The basic parsing algorithms are based upon parallel LR parsing; extension of a grammar will not quickly lead to a reduction in speed.
- The developed software is independent of the lingware.

A more detailed description of the Language Editor goes beyond the scope of this paper. We invite interested parties to attend a live demonstration.

Further remarks on the technical design of lingware

In the Technical Design phase the software tools do not need much customization. However, the lingware has to be developed carefully in order to precisely adapt the subsystem to the language use of the customer. The development can start from scratch or from already developed lingware for the specific language and domain.

The restriction of linguistic resources cannot be realized by means of prohibitions. After all, the number of prohibitions is always finite, while the language is capable of producing an infinite variety of sentences. Restriction can only be realized by means of a positive prescription of linguistic resources. That is to say, by means of do's instead of don'ts. For this we can use a reliable, manageable and verifiable method: a basic set of instructions may be expanded step by step in the direction of the desired linguistic resources. The only doubt which might exist is whether this will create sufficient expressive potential. For writing a novel this will probably never be the case. Up to now, the experience with industrial texts is positive, however. Stylistic excesses in industrial texts are usually due to the author's inexperience.

The procedure of linguistic analysis is the exact opposite of the development of translation systems for general use: it begins with the simplest use of language. Part II of this paper gives a detailed explanation of the procedure. A rough outline of the working method is depicted in figure 2.

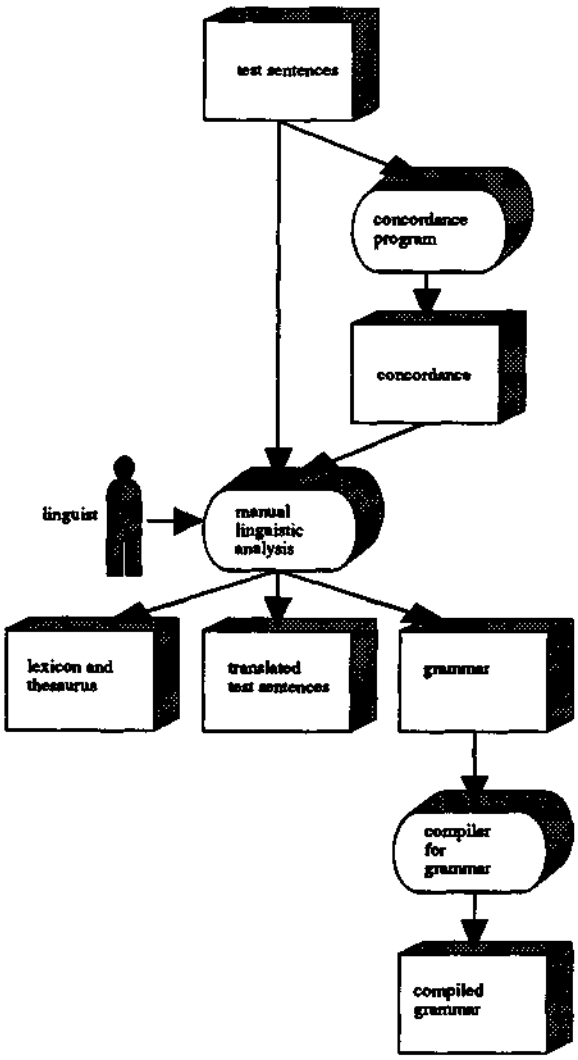


Figure 2 Linguistic analysis and the construction of lingware

9. STATUS OF DEVELOPED LINGWARE

This section describes the current status of the lingware.

Grammars:
The following grammars were developed:
- Dutch correction and translation Dutch -> English, aimed at:
    - help texts for textile companies;
    - help texts for an insurance company;
    - software manuals.
- Simplified English correction, aimed at:
    - texts for aircraft maintenance manuals.
- Dutch -> Spanish translation, aimed at:
    - help texts for an insurance company.

Currently under development are:
- Extension of the Simplified English correction, aimed at:
    - a feasibility study as to what extent natural English can be standardized to Simplified English.
- Simplified English -> French translation.

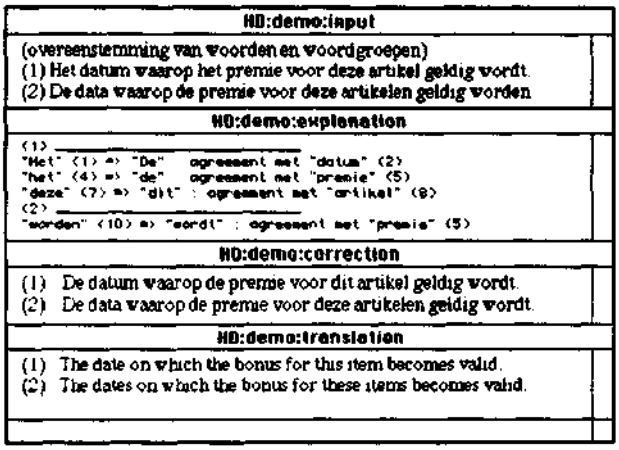
Lexica:
For all the grammars mentioned the corresponding lexica have been developed for the specific domains.

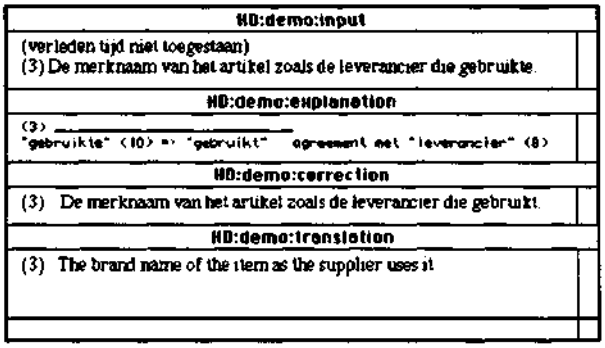
10. DEMONSTRATION OF CORRECTION AND TRANSLATION

The results of the system development of the "engine" are shown to the outside world by means of a demonstration. In order to give an impression of the possibilities we provide a number of examples in the form of snapshots of the windows which are used for this demonstration.
These are the windows for:

- input sentence(s), called HD:demo:input
- explanation(s), called HD:demo:explanation
- corrected sentence(s), called HD:demo:correction
- translated sentence(s), called HD:demo:translation.

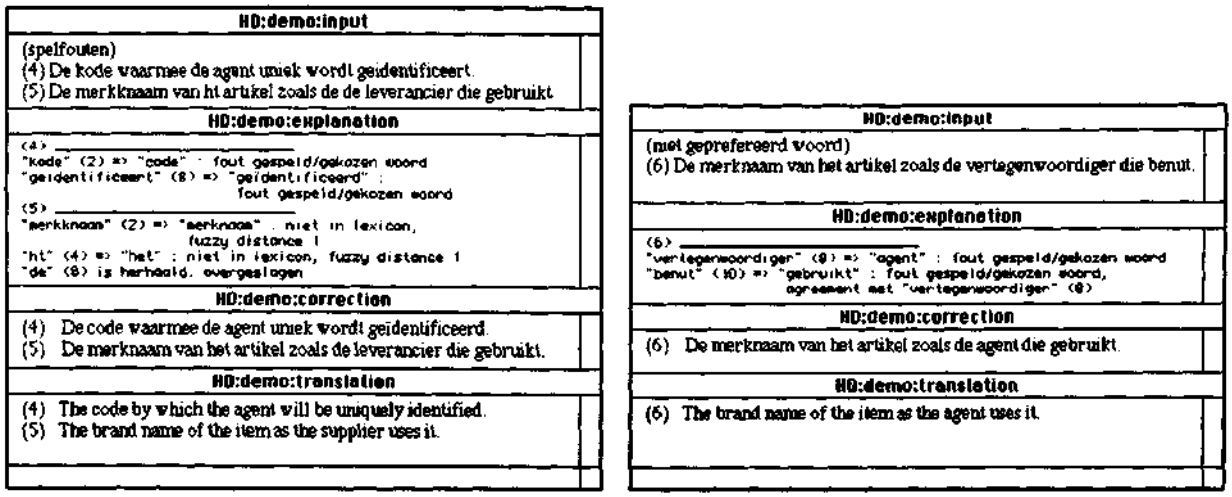
Each snapshot displays a particular linguistic phenomenon which is shown between brackets in the input sentences window. The caption provides an explanation. In the explanation window, numbers appear between brackets after the words. These are the serial numbers of the words in the sentence.





In sentence (1) there is agreement between separate words. In sentence (2) between word groups: "de premie" and "wordt geldig"

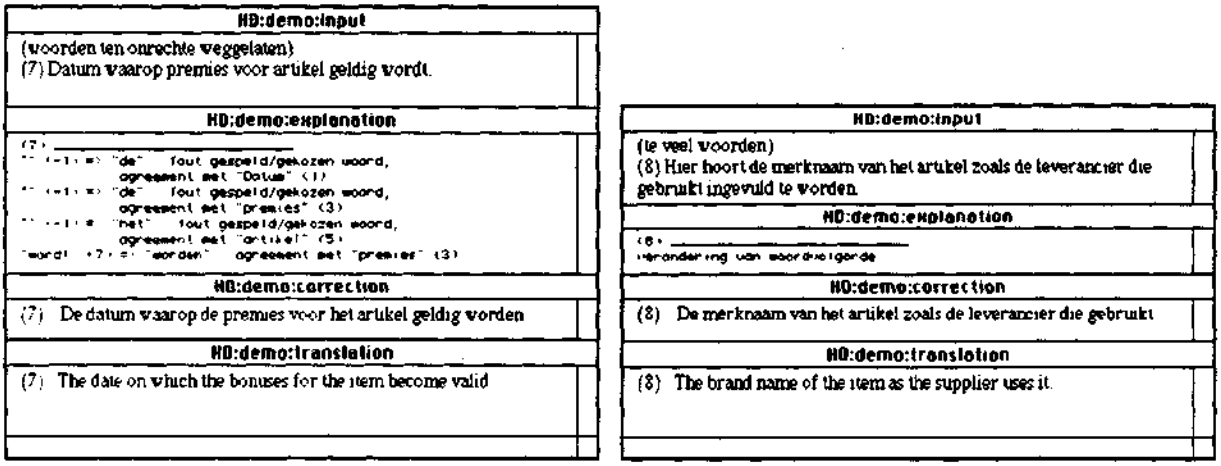
Sentence (3) contains a past tense. In this help text, however, past tenses are not allowed. This is why it is changed into a present tense.



Sentence (4) contains two recurring spelling errors: the preferred spelling is not used and a past participle spelling error is made. These errors are provided for by the grammar and the lexicon and are corrected.

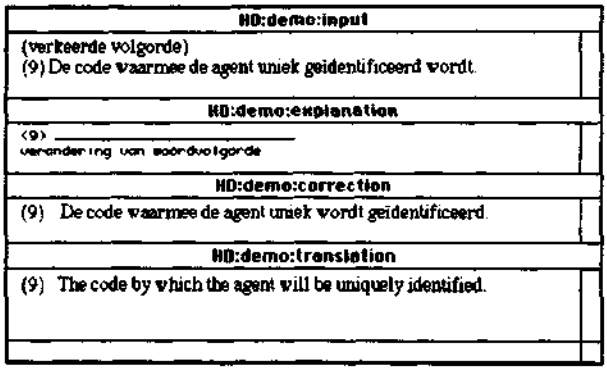
Sentence (5) contains typing errors. These are handled by a general correction mechanism which produces a number of alternatives, graded according to a distance criterium for typing errors which is here called "fuzzy distance".
This only produces alternatives which are grammatically correct, unlike most spelling checkers which cannot make use of a prescribed grammar. In this case the correctness of the correction cannot be guaranteed. The writer's attention will be drawn to this correction.

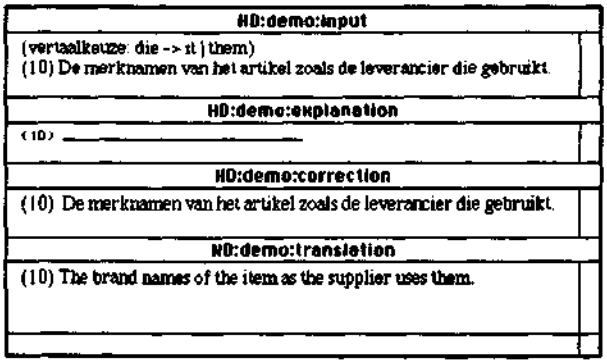
Sentence (6) contains two words which are in themselves correct but for which the organization uses another term. The replacement word "gebruikt" is automatically given the correct conjugation.



In sentence (7) words have been omitted which the grammar says ought to be there. These are added. The correct word form is selected on the basis of grammatical and lexical knowledge. "Wordt" is also corrected.

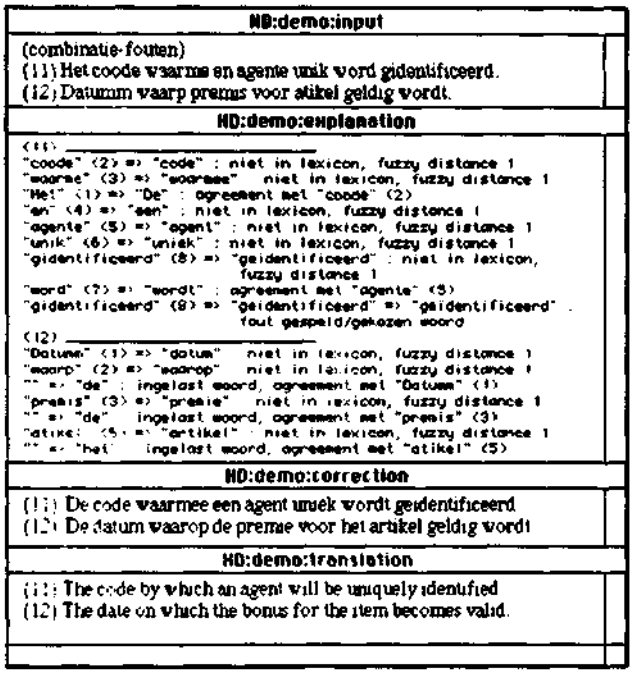
Sentence (8) contains too many words. Some writers have the tendency to express themselves in sentences like this one. This characteristic pattern has been incorporated into the grammar so that it may be corrected.





The word order in sentence (9) is not wrong as such. The organisation in question, however, prescribes the use of a different word order, so the sentence is corrected.

Sentence (10) is correct. The word "die" can be used for both "merknaam" and "merknamen". The English translation, however, can be either "it" or "them". The translation program takes this into account.



Sentences (11) and (12) contain combinations of the preceding errors.

PART II: LINGWARE DEVELOPMENT FOR HELP TEXTS AND TECHNICAL MANUALS

In the last few years we have seen numerous articles being published, reporting on the rapid development of products that can handle natural language. According to Keijzer, globally, 2,5 billion guilders will be spent in this market towards the year 2000. [Keijzer (6)]. A rather impressive figure! However, much remains to be done before we will reach that point. Field testing of existing systems has shown that the problem is not the development of lingware for a small, restricted language domain but the adaptation of the basic product to a specific new situation. First of all the process of adaptation is usually not transparent enough to the potential user. Secondly, the adaptation process is not marked by clear boundaries, thus no one can estimate the costs involved.

One of the principles of LS is that the lingware should be developed in a manageable way in order to prevent the above-mentioned problems. We adopt the same line of approach to lingware development as we do to commercial software development (c.f. section 8). Within LS we have developed working methods for a structured approach to analyzing restricted language domains and to developing the lingware for these language domains.
This part describes the working methods we developed for building the lingware for help texts and technical manuals respectively.
For a better understanding of our working methods we will first give a moment's thought to some frequently recurring terms.

11. RESTRICTED LANGUAGE. CORRECTION. STANDARDIZATION AND TRANSLATION

LS's ultimate goal is the creation of systems that, entirely automatic, yield correct translations of texts made up of restricted language. To achieve this goal we have opted for a system that consists of two main modules: a language editor and a translation module. The language editor has a linguistically supporting and controlling task. Text entered on-line has to be corrected and standardized where necessary, possibly in cooperation with the author. The following sections will discuss the terms "restricted language", "correction", "standardization" and "translation" in more detail.

Restricted language

Due to the restriction of the natural language domain, the problems related to automatic translation are restricted and manageable. The restricted language SL' is a subset of the natural language SL (see figure 3) and contains only a part of the lexical, syntactic and semantic variations that occur in SL. The abbreviations SL' and SL refer to "source language". In a translation context the source language is used as opposed to the "target language" (TL).

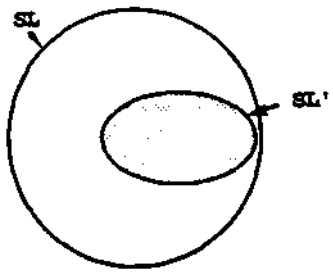


Figure 3 The relation between the restricted language SL' and the natural language SL

The development of a system for the automatic translation of restricted language therefore seems an obvious choice for those who aim to yield a small working product at a relatively short time. On the

other hand this approach causes typical problems related to restricted language. This can be shown by several short sentences from the MODIX corpus which we investigated. This corpus (written in Dutch) contains help texts which are derived from an information system developed by LS for the fashion retail industry. [Van der Tol (7)]. [N.B. we will number from sentence 1 onwards again].

(1) De datum waarop de leverancier de klacht heeft afgewerkt.
The date at which the supplier dealt with the complaint.

(2) Formaat EEJJMMDD.
Format CCYYMMDD.

(3) Alfanumeriek, 6 posities.
Alphanumeric, 6 positions.

(4) De commentaarregel biedt de mogelijkheid om het contract nader te beschrijven.
The comment line offers the possibility to describe the contract in more detail.

(5) Wijzigen bonusregeling leverancier.
Modify bonus scheme supplier.

Every automation expert will quickly grasp what these sentences are about. However this will not be true for every native speaker. The sentences (1) and (4) are correct sentences. A native speaker will also understand that sentence (5) is an order to change the bonus scheme of a supplier. He would only word it differently since it does not contain articles and the preposition "van" ("of"). Sentences (2) and (3) however will sound mysterious to him.
Sentences (1) to (5) and other similar sentences show that the restricted language SL' is a subset of the natural language SL, but not automatically a subset of the standard language SLstd, the language as described in the grammar books. Lehrberger states that a restricted or sublanguage SL' of the natural language SL can be regarded as the result from restrictions on and deviations from the grammar of the standard language SLstd [Lehrberger (8)]. The relation between SL, SL' and SLstd is then as shown in figure 4.

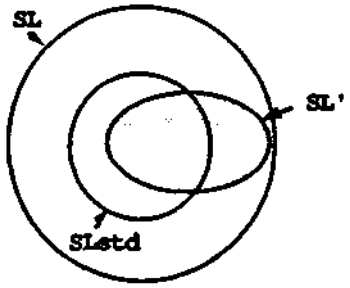


Figure 4 **The relation between the restricted language SL', the natural language SL and the standard language SLstd**

In this figure the intersection of SL' and SLstd contains those sentences of SL' that can be described in terms of restrictions on the grammar of SLstd (see sentences (1) and (4)). The remainder of SL' contains sentences that deviate from the grammar of SLstd in some way, although they are considered SL'-grammatical (see sentences (2), (3) and (5)).

A project focusing on restricted language or sublanguage should therefore take into account that we are not only faced with a reduction of problems. Some language phenomena that occur in the language domain to be implemented do not satisfy the grammar of the standard language. Since research on restricted languages has started fairly recently, grammars for these sublanguages are hardly available. It is therefore necessary to specify special rules for the linguistic description of those sentences in the subset that are not satisfied by any grammar of the standard language.

Specifying the grammar rules should preferably be carried out in close cooperation with the user(s) of the restricted language.

Correction and standardization

Entering source text is usually done on-line. This may result in texts not being examined thoroughly enough to detect all errors. And usually, little attention is paid to the way company standards are used with regard to lexicon, terminology and grammar.

We have developed a language editor to prevent the translation module from being confronted with incorrect language and/or language that does not meet the current standards. The language editor acts as a filter which, if necessary, corrects and standardizes the input, possibly in cooperation with the author. Applying the terminology used, the input of the language editor, I, can be as shown in figure 5.

In this figure the intersection of I and SLstd contains the sentences of I which are described by the grammar of the standard language. The section of I outside SL contains ungrammatical sentences. The remainder of I contains sentences which do not satisfy the rules of the grammar of the standard language SLstd, but which are nonetheless SL-grammatical.

The intersection of I and SL (the shaded area) contains the sentences belonging to the restricted language SL'. This is the set of sentences which has to be described by the lingware. In addition the lingware should also contain suggestions for correcting incorrect language.

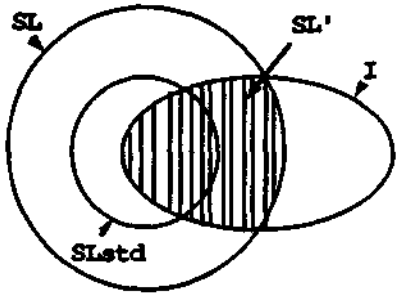


Figure 5 **The relation between the restricted language SL', the natural language SL, the standard language SLstd and I, the input of the language editor**

Correction. Which language errors should a language editor take into account? Kusters and Van der Steen give a survey of language errors in the MODIX corpus. [Kusters (3)]. The errors described are user errors, in a dialog between a human being and a computer, the computer is also likely to make mistakes. Véronis makes a distinction between errors made by a human being, the so-called user errors, and errors made by a computer, the so-called system errors. [Véronis (9)]. In addition he distinguishes two other types of errors: competence errors (errors stemming from a lack of knowledge, competence) and performance errors (errors slipping in while using language, for instance typing errors or errors due to the incorrect representation of a character by a scanner). We assume that no performance errors are made by the system for automatic translation. So performance errors are user errors. Competence errors on the other hand can be divided into user and system errors.
From a linguistic point of view, errors can be made at the lexical, syntactic and semantic level. The appendix on the typology of errors deals with possible errors at the first two levels.

The translation system developed is able to recognize and correct a large amount of the lexical and syntactic errors which are given in the appendix. For errors made at the lexical level the requirement holds that the correct word form is included in the lexicon. An incorrect word form will be replaced by the word form most resembling the incorrect one in the same syntactic category. Due to

the absence of semantics, this will not always be the correct word since the semantic competence of a system would then have to be very large in order to yield satisfying results. For error correction at the syntactic level it is necessary that the correct syntactic structure is incorporated in the correction grammar. Real errors are thus handled in a syntactically correct way.

At the syntactic level however, language errors are often referred to as language variation. Language errors must be corrected, language variation has to be standardized according to the wishes of the user.

Standardization. A human being can express himself in many ways. An example from the MODIX corpus: an automation expert can express his order to view the data of the sales department in the following ways:

(6) Raadplegen gegevens vertegenwoordiging.
(7) Raadpleeg gegevens van vertegenwoordiging.
(8) Gegevens van de vertegenwoordiging raadplegen.
(9) Raadplegen van de gegevens van de vertegenwoordiging.
(10) Raadpleeg de gegevens van de vertegenwoordiging.

The grammars of the standard language will usually only describe (10). According to the terminology used, sentence (10) is included in the intersection of I and SLstd. The other sentences are not ungrammatical, that is they are not part of SLstd but part of the remainder of the intersection of SL and SL'.
At the lexical level it may be noted that some words can be spelt in different ways. The variance in spelling also occurs in the MODIX corpus. We want to mention two examples:

(11) Code.
(12) Kode.

A dictionary of the standard language will usually include (11) as the correct spelling form and (12) as a variant, a word form not part of SLstd but part of the remainder of the intersection of SL and SL'.

When speaking of standardization we refer to the shaded part in figure 6.

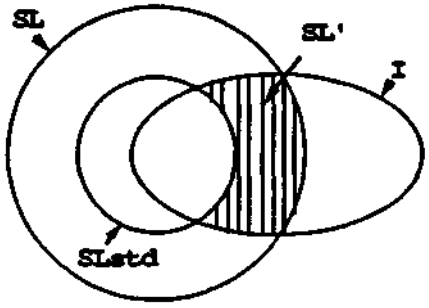


Figure 6 Standardization

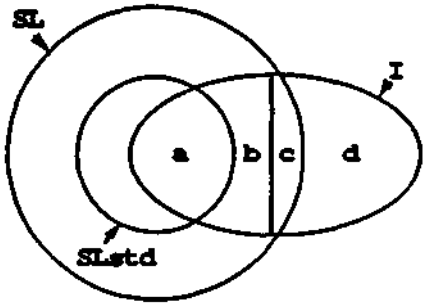


Figure 7 The subdivision of the language editor's input

Our language editor provides for standardization at the lexical level by means of a lexicon and a built-in thesaurus. Standardization at the syntactic level is provided for by way of the description of possible structural variants in the correction grammar. It is however impossible to include all variants. The competence of the grammar has its limits. Therefore the user should be consulted in order to make decisions about error prevention for which restricting the lexical and syntactical variants is of great importance (see the paragraph on Error prevention in section 12).

Does standardization refer to transforming variants into words and sentences that are part of the intersection of SLstd and SL'? Though sometimes very complicated, it is possible according to [Lehrberger (8)]. This is what the language editor we developed therefore actually does, though not always, for it is possible to think of situations in which this type of standardization is not desirable. An organization can use its own standard, which in some cases deviates from the standard language. In such a case the translation system will have to take this company standard into account. Figure 7 shows the subdivision of the language editor's input, as viewed by us.

Subset **a** contains all language phenomena of the restricted language which belong to the standard language, for example:

(13) De datum waarop de divisie niet meer geldig is.
The date at which the division is not valid anymore.

Subsets **b** and **c** contain those language phenomena of the restricted language which do not belong to the standard language, **b** contains those phenomena which the user regards as belonging to the standard of the restricted language, like:

(14) Landcode is alfanumeriek, 3 posities.
Country code is alphanumeric, 3 positions.

These variants of or deviations from the standard language need not or must not be transformed into a language phenomenon which belongs to the standard language. Subset **c** contains deviating restricted-language phenomena, for instance sentence (15), in which the order of elements after the auxiliary does not meet the company standard.

(15) Maatstelselcode is 2 posities lang, alfanumeriek.
Measurement code is 2 positions, alphanumeric.

These phenomena have to be transformed into phenomena that are part of the subsets **a** or **b**. This is the required and necessary standardization. Finally, subset **d** contains those phenomena in the input which do not belong to the language SL, for instance (16) which contains an incorrect verbal form.

(16) De code waarmee de leverancier het artikel identificeerd.
The code which the supplier uses to identify the item.

Correction means that phenomena from this set have to be transformed into phenomena which are described by the grammars and the lexicons of **a** or **b**.
Correction and standardization performed by the language editor yield output as shown in figure 8.

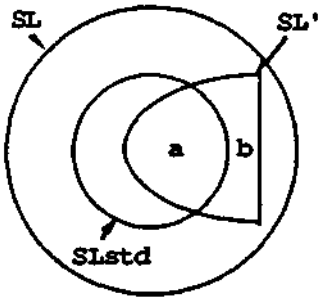


Figure 8 The output of the language editor, c.q. the input for the translation module

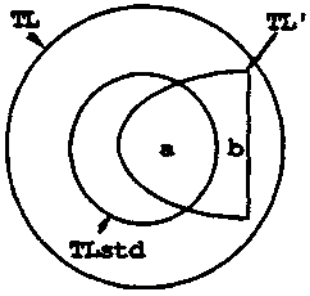


Figure 9 The output of the translation module

Translation

The output of the language editor, the corrected and standardized I, constitutes the input for the translation module. The translation module has to comply with a few requirements: the input must be flawless and has to be translated rapidly.
The output of the translation module is shown in figure 9. The language phenomena that are part of the intersection **a** of the standard source language SLstd and the restricted source language SL' are translated into the intersection **a** of the standard target language TLstd and the restricted target language TL', whereas the language phenomena that are part of the intersection **b** of the source language SL and the restricted source language SL' are translated into the intersection **b** of the target language TL and the restricted target language TL'.

12. WORKING METHODS

Having described the relevant terminology and typology, we will now proceed with the working methods which we developed for the analysis of specific language domains and for the implementation of the lingware belonging to those language domains. First, we describe the working method we developed for help texts. Next, the working method developed for technical manuals will be described.

Help texts

For the development of lingware for help texts we adopt the same stages as in the development of commercial software. The working method developed is continuously refined and can be listed as follows:

I
Collect examples of texts from the user's language domain, the so-called corpus C.

II
Submit the corpus to linguistic analysis and determine, in cooperation with the user, which parts can or must be described. (The situation may arise that a part of the corpus is very different from the standards used by a company. For another part of the corpus it may be true that it is still too complex to be dealt with adequately in view of the current knowledge in computational linguistics.)

III
Divide the part of the corpus that is to be described by the lingware, the input I, into a number of subsets of sentences that will be analyzed and described iteratively.

IV
Classify the sentences of a subset in a classification table on the basis of their grammatical structure.

V
Build both grammars, the bilingual lexicon with built-in thesaurus and the test suite for the subset.

VI
Insert suggestions for error corrections in the correction grammar and the lexicon. (These suggestions standardize the sentences of **c** and correct the sentences of **d**).

VII
Check after every iteration step whether the developed lingware L is still performing adequately.

VIII
Stop as soon as a residual set remains that is sufficiently small and so complicated that otherwise the grammars would have to be expanded excessively.

IX
Standardize the sentences in the remaining set until they are acceptable.

X
Account for a characteristic standardization pattern in the correction grammar.

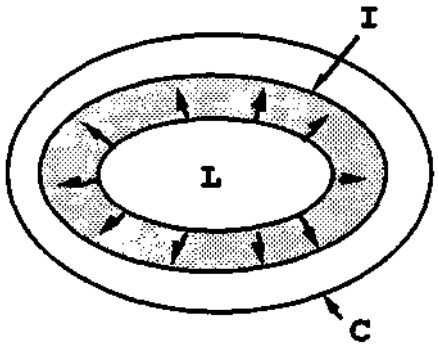


Figure 10 **The relation between the corpus C, the subset I to be described and the lingware L which is expanding iteratively**

The relation between the corpus C, the subset I to be described and the lingware L which is expanding iteratively is shown in figure 10.

The working method described above intends to provide insight into the development cycle to the lingware programmer as well as the prospective user. Programmer and user define as clearly as possible the domain to be described by the lingware. The expansion of the lingware L can be followed closely on the basis of the classification table. To get a clear picture of the costs involved in a project, we closely keep track of statistical data related to the development process of the lingware. In the following subsections some of these steps will be examined in more detail.

Classification of sentences. The classification of sentences constitutes an important step in the linguistic analysis of the corpus. We prefer to classify sentences according to a certain, already existing, division, resulting from our wish to be able to indicate in a simple way what the relation is between the restricted language described, and the other restricted languages and the standard language.

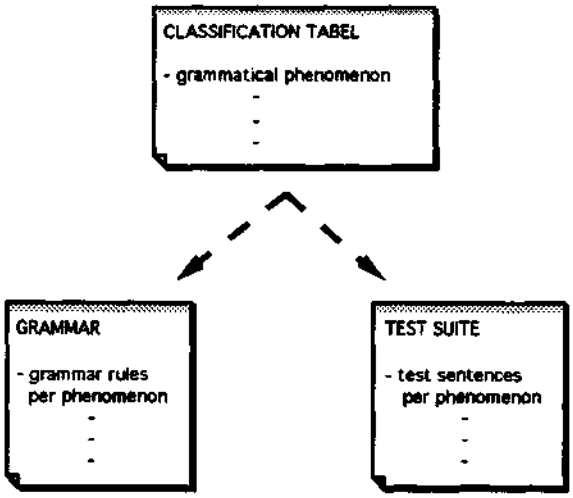


Figure 11 **The relation between the classification table, the grammars and the test suite**

Classifications are offered by grammars like the ANS, [Geerts (10)] and Quirk, [Quirk, (11)], and test suites like the Flickinger test suite and the Way test suite, [Flickinger (12), Way (13)]. Still the intersection of the grammatical phenomena that occur in for instance the MODIX corpus and the individual grammars and test suites, turned out to be too small. Thus the need arose to create an entirely new classification, geared to the restricted language. This classification is based on the grammatical structure of the sentences and constitutes the basis for building both the grammars and the test suite in a structured manner (see figure 11).

Classification of the grammar rules. An important reason for classifying sentences according to their grammatical structure is to provide the linguist with structured support when writing expanding grammars. As grammars grow, it becomes more and more difficult to maintain and manage them. This is the reason why we created a 1-to-l relation between the grammatical phenomena described in the classification table and their corresponding grammatical rules.
The creation of a 1-to-l relation between the grammatical phenomena and their corresponding test sentences in the test suite allows for the possibility to check rapidly whether the correction and translation modules still function adequately after every modification in the lingware.

Error prevention. The set of sentences that is not described by the lingware, the outer ring in figure 10, is divisible into two subsets. The first subset contains sentences which an organization considers too far removed from the standard language within the company. An author would absolutely not be allowed to use such sentences. In addition, the lingware of a product for natural language processing does not have to be able to handle it. The second subset contains sentences of which computational linguistics currently still lacks the knowledge for describing them adequately.

The question how to prevent the user from entering sentences that belong to one of these subsets is answered by, among others, [Gotlieb and d'Haenens (14)]. In this article on systems currently used for automatic translation by the industry and administration in Canada, they state that of all the systems they have examined, there is only one system that recovers the costs: Xerox' Systran. The main reasons behind its success are:

- The domain of the texts that have to be translated is severely restricted (operating and training manuals). Furthermore, Xerox spares no trouble or expense on editing the source text such that it can be recognized by the translation system.
- A great deal of time and money have been invested in building a translation system that meets the company requirements.

The same reasons can be given for explaining the success of another remunerative system for automatic translation (but not discussed by Gotlieb and d'Haenens), namely the TAUM-METEO system.

At Xerox technical writers use a special kind of English, the so-called Multinational Customized English (MCE). The guidelines to this language are owned by Xerox. In a brief course the authors are taught how to write unambiguously, brief and clear. They may only use the words of the lexicon in the translation system. They are also given a short survey of the grammar of MCE and examples are given of sentences which the translation system finds hard to deal with.

From this you may conclude that it does not suffice to analyze and classify only the sentences that have to be described by the lingware (the input I). If a system ever wants to be accepted by an author, it will also be necessary to classify the sentences that have not been described (yet). This whole classification constitutes the basis for the writing guidelines; guidelines that aim at making the system transparent. The writers have to learn which grammatical structures and which words to use, that is, which grammatical structures and which words are part of the set I and which are not. Thus, error prevention means reducing the outer ring of figure 10, which is visualized in figure 12.

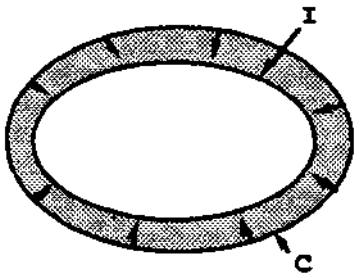


Figure 12 Error prevention

Results achieved. The description of the working method was followed by the actual implementation of the MODIX corpus. The language pair chosen was Dutch/English. The working method used for this corpus and language pair is discussed step by step below.

I
This step was not necessary. The text was already available, the so-called corpus C.

II
The MODIX corpus consists of 2900 lines. Most lines begin with a code. In most cases the code is followed by a sentence of at least one line. The corpus C has 1916 sentences. All sentences were submitted to a linguistic analysis. On the basis of this, we decided that 1470 sentences ought to be described by the lingware. So the set I contains more than 76% of the sentences of set C.

III
The classification of sentences and the implementation of the lingware for these sentences took place iteratively. For every iteration step 100 sentences were analyzed, classified and accounted for grammatically.

IV
The iterative analysis of the sentences of I has been compiled in a classification table and has not been included in this English version of the paper since the table contains only Dutch examples. For those interested, the table is available on request. Please contact the authors.
Since the language phenomena deviated sharply from any standard, we have chosen for our own standard. This classification is purely practical. Theoretical considerations concerning the drawing up of classification tables which are not restricted to the standard language seem to offer interesting grounds for research in the near future.

V
Before we started the development of the lingware, we aimed at covering at least 80% of I. Presently, 1291 out of 1470 sentences from I are covered by the developed lingware L. This amounts to 88% of I and has been achieved with a correction grammar of 436 production rules, a translation grammar of 322 production rules and a bilingual lexicon of 2595 entries.

VI
After every iteration step the test suite was used to test whether the lingware not only described the newly implemented phenomena but also whether all phenomena previously covered are still covered.

VII
During the development of the lingware we also implemented suggestions for error correction. The suggestions implemented are engrafted on errors detected during the linguistic analysis of the corpus.

VIII
Before the implementation started we set the acceptable size of the residual set to 20% of I. This goal was attained without the need for standardization. This is also the reason why the steps IX and X have not been carried out.

By means of the sentences of C which were excluded from I on the basis of linguistic analysis, the author of help texts can be given several linguistic suggestions concerning the language to be used. In general it can be stated that sentences should be short, clear and unambiguous. More specifically however, the author can be given tips based on concrete examples from the corpus. For those interested, the table drawn up for the MODIX corpus is available on request.

Statistics. During the development of the lingware for the MODIX corpus we collected statistics to gain better insight into the speed of development and the way in which the lingware develops. The graphs of figure 13 through 16 show the results. The target we set with respect to grammatical coverage (covering at least 80% of I) already exceeded our expectations after handling 700 sentences. This explains the horizontal lines after 700 sentences in the graphs that show the growth of the two grammars (figures 13 and 14). The graphs show the following relations:

- the production rules of the correction grammar and the number of sentences described (see figure 13);
- the production rules of the translation grammar and the number of sentences described (see figure 14);
- the lexical entries and the number of sentences described (see figure 15);
- the number of days and the number of sentences described by a linguist (see figure 16).

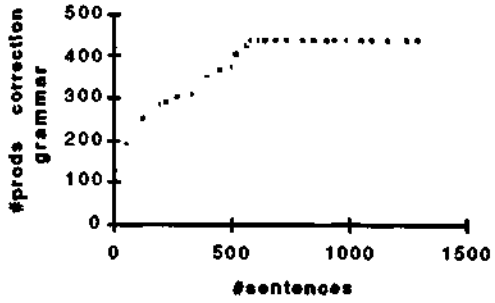


Figure 13 The relation between the number of production rules of the correction grammar and the number of sentences described

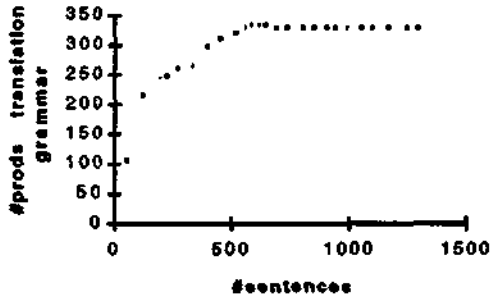


Figure 14 The relation between the number of production rules of the translation grammar and the number of sentences described

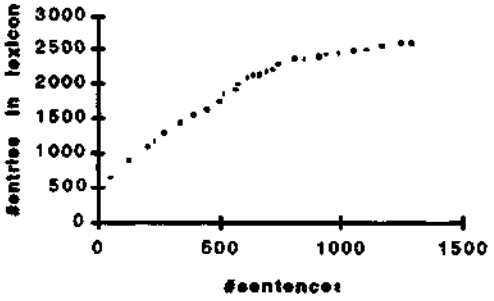


Figure 15 The relation between the number of lexical entries and the number of sentences described

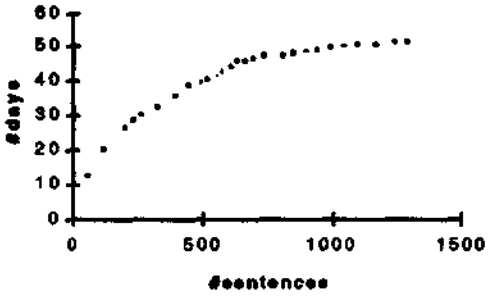


Figure 16 The relation between the number of days and the number of sentences described

The relation between the corpus C, the subset I to be described and the lingware L which expands iteratively, as shown in figure 10, is again given in figure 17. The figures show the number of sentences from the MODIX corpus that are part of the various sets.

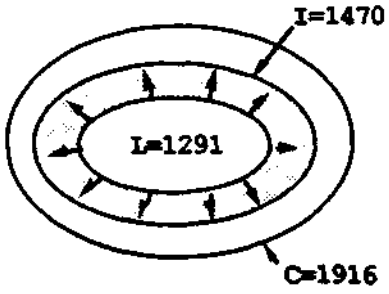


Figure 17 The relation (in the MODIX corpus) between the corpus C, the subset I to be described and the lingware L which expands iteratively

After the completion of the lingware for the MODIX corpus we realized a commercial project for a client in which we treated Dutch help texts with subsequent translation into English and Spanish. Other translation modules are planned for the near future.

Technical manuals

In this section we describe a working method for the linguistic analysis of the more complicated language domain of technical manuals, and for the implementation of the lingware for this domain.

Technical documentation: a notorious problem. In industry the development of adequate technical documentation is known to be a notorious problem. The president of Uniface, a successful software producer, recognizes the production of software documentation to be one of his main problems. [Wammes (15)]. Some parts of the industry already developed standards and guidelines describing the language usage allowed and not allowed in (parts of) their manuals. We think of for instance AECMA's Simplified English within the civilian aircraft industry, Xerox' Multinational Customized English, Perkins Approved Clear English, Caterpillar English, Ericsson English and Swedish and Rationalized French. These restricted languages are specified by lexicons in which lexical semantic ambiguity is not allowed and by small sets of more or less informal rules of grammar, style and orthography.

Software developers within our company also have problems with the development of adequate technical documentation. Having noticed our good results with help texts, they asked Lingware Services to examine whether (parts of) their documentation process could be formalized and automated in the near future. The input for this investigation consisted of a draft version of what is called the Draw_Master manual.

Technical manuals vs. help texts. We started with a linguistic analysis of the manual, following the same strategy as for the analysis of help texts. Sentence by sentence all 30 pages were studied. After the examination of the first few pages we already noticed an essential difference between the help texts of the MODIX corpus and the Draw_Master manual.

In the investigated help texts the set of words as well as the set of possible sentence structures is clearly restricted. These restricted sets can be formally described. The author using our correction and translation system for the creation of new help texts will feel little restriction, because the grammar and lexicon of the system cover the required language usage.

In the Draw_Master manual the set of words used is also restricted. The domain described is very limited. Yet, the set of possible sentence structures seems to be unrestricted. Unlike the authors of the help texts, the authors who wrote the documentation for the Draw_Master application did not appear to have used a restricted set of sentence structures.

There will be no problem developing a lexicon for this technical manual. However, problems arise when a grammar has to be developed. Whatever formal grammar is created, it will always restrict the author. While a modern linguist would feel uneasy knowing he was writing a grammar that would clearly restrict the linguistic performance of the authors, the software developers who faced the problem of inadequate documentation, were actually looking for specific guidelines, not only on text level, but also on sentence and word level. They were not at all unhappy with the idea of having to work with a grammar that restricts the syntactic diversity in their manuals.

Developing a basic grammar for technical manuals. Observing the general documentation problems and having learned that the restriction of the author's language usage for the sake of readability is allowed - even desired - we decided to build a basic grammar for the creation of simple technical manuals. Two questions arose: What rules should this grammar contain? And what is the best way to build a prescriptive formal grammar for technical manuals in a structured way? Not knowing which one would give the best results, a bottom up approach or a top down approach, we decided to follow both approaches to find an answer to these questions.

The bottom up approach is essentially based on the language used in a number of concrete texts (=language performance). The starting point was formed by the investigation of the syntactic structures in different technical manuals. The purpose was to see whether it would be possible to deduce a large enough intersection of syntactic structures that would be representative as a basic grammar for the language domain.

The intuition that people have about their own language (=language competence) is the basis of the top down approach. The starting point was formed by the investigation of source language grammars and writing guides, all of which describe the standard language usage. The goal was to translate a representative subset of the topics addressed in these grammars into formal grammar rules. Additional rules for error correction and standardization were taken from the writing guides.

For both approaches it was of major interest to see whether the grammars developed could form a good basis for the prescription of the language usage in a particular technical manual such as the Draw_Master manual.

The results achieved with the top down approach are now described in more detail.

The top down approach. The linguistic examination of the Draw_Master manual showed that the language usage in this manual closely resembles the standard language usage. Only a few minor deviations from the standard language usage could be found, such as for example the complete absence of the simple past tense and the sparse usage of pronouns and negation. The first goal of the top down approach was therefore the creation of a formal grammar that would describe a basic subset of Dutch, the language in which the draft version of the manual is written. Two grammars, [Geerts (10)] and [Van Bart (16)], and a writing guide, [Renkema (17)] formed the starting point for the creation of a basic formal grammar that would eventually contain almost 700 production rules, including rules for both the correction of syntactic errors and the standardization of syntactic and stylistic variation.

The grammar having reached this size, the working method developed for the implementation of help texts was picked up again to customize the basic grammar and the lexicon to the language usage in the software manual.

Page by page
- a number of sentences was analyzed;
- whenever they deviated too much from the basic language described or did not clearly belong to standards of the language domain, the sentences were rewritten;
- the lexical content of the sentences was, if not yet present, added to the lexicon;
- the sentence structures not yet present in the basic grammar but seeming to be typical sentence structures of the language domain, were added to the grammar;
- suggestions for error correction and standardization were, whenever necessary, added to the grammar and the lexicon;
- the performance of the updated lingware was tested.

After the complete analysis of the draft version of the Draw_Master manual, the lexicon with built-in thesaurus function contains a little over 3000 entries. The grammar contains 870 production rules, which means that the basic grammar has increased by 170 rules.

In the next paragraphs some important topics concerning the basic grammar and the working method followed are addressed.

Reusability and extendability. During the analysis of the corpus of MODIX help texts, the sentences were classified in a classification table. The classification table was based on the grammatical structures present in the sentences. One of the main reasons for classifying the sentences was to investigate the reusability of (parts of) the grammars and the test suite for the development of lingware for another corpus or language domain. It was assumed that the intersection of the classification tables of two corpora would immediately yield the grammar rules already present as well as the test set of that particular intersection.

However, the difference in grammatical complexity of help texts and technical manuals is enormous. While the number of different grammatical structures in the help texts was clearly restricted,

this can not be said about the grammatical structures in the sentences of the Draw_Master manual. It would be a very difficult - if not impossible - task to make a neat syntactic classification of the sentences in the Draw_Master manual. Therefore a classification table for the Draw_Master manual is lacking.

Nonetheless we were anxious to know the coverage of the actual grammar compared to the grammar of the help texts. A simple scan of the MODIX classification table shows that the Draw_Master grammar covers more than 90% of the grammatical structures present in the MODIX corpus. The same scan shows that the MODIX grammar is highly specialized, i.e. concentrated on the specific syntactic structures present in the corpus of help texts. It does not even cover 10% of the Draw_Master sentences.

From this we draw the conclusion that the grammar developed for the Draw_Master manual which is based on the core of some standard grammars and a writing guide may well be an ideal basis for the rapid development of grammars for comparable or smaller language domains.

Even larger language domains fall within our scope: because the grammar has been divided into a large set of modules describing different syntactic topics, future extensions are definitely feasible.

The (re)writing process. The extended basic grammar covers a subset of the Dutch language: the typical language domain of the Draw_Master manual. Sentences not belonging to that subset are not accepted, and have to be rewritten.

Although writing and rewriting manuals and other documents is a daily task for the people working in a documentation department, there are no formal guidelines for this process. The fact that the language usage has to be correct, clear and in accordance with certain standards is commonly accepted, but these topics are only informally described.

The formal implementation of the language usage in a technical manual gives us the possibility to accurately describe the grammatical and lexical coverage of the lingware. This description, when written down in a user manual, will give the author and the documentation staff exact and formal guidelines about the terminology, the orthography and the syntax allowed within the language domain.

Information about strongly deviating and therefore not permitted sentence structures is absent in the lingware. This information is however needed by the author and the documentation department. One way or another it should be present and it should be as accurate as possible. Therefore we developed a writing guide listing the not permitted sentence structures accompanied by alternatives that are covered by the grammar.

The problem with these not permitted sentence structures is that their number is in fact infinite. A writing guide will always give a minor subset of them. Here we present some structures sparsely encountered in the Draw_Master manual that are not described by the present version of the grammar. Each structure is followed by an example and an alternative.

• impersonal constructions, i.e. constructions with the impersonal pronouns "het" and "er".

example:
Er zijn operaties die de grafische objecten beïnvloeden.

alternative:
Sommige operaties beïnvloeden de grafische objecten.

- constructions in which heads and their complements or modifiers are separated by one or more elements, such as nouns separated from the relative clause or prepositional phrase they are modified by.

  example:
  Indien extra informatie van de gebruiker is benodigd, wordt *een dialoog* gestart *die de benodigde informatie verstrekt.*

  alternative:
  Indien extra informatie van de gebruiker is benodigd, wordt *een dialoog die de benodigde informatie verstrekt* gestart.

- complex verbal constructions such as "invloed hebben op" in the next example.

  example:
  Sommige operaties *hebben invloed op* de grafische objecten.

  alternative:
  Sommige operaties *beïnvloeden* de grafische objecten.

These not permitted structures can be replaced by alternative structures that are part of the grammar. They could even be described in a future version of the grammar. Both the grammar formalism used and the modular structure of the grammar offer this possibility.

Besides syntactic structures that could quite easily be inserted in a future version of the grammar, there are also structures that fall outside the scope of the grammar formalism in its present state. It concerns the correct and complete treatment of complex phenomena as e.g. discourse, pronoun resolution, collocation, gapping, scope and negation, all of which are hot items in today's computational linguistics.

Conclusions and remarks about user acceptance. The correction, standardization and translation of technical manuals such as the Draw_Master manual can be automated if the writing and rewriting process is founded on formal guidelines.

A user manual containing an exact description of the formal grammar and lexicon, and a writing guide containing the non-permitted sentence structures with - if possible - their alternatives supply the author with a set of formal guidelines. At present we are investigating how the author of a technical manual can be taught in an inspiring way to use only the restricted subset of sentences that are described by the lingware. Questions we ask ourselves are: How should the user manual and the writing guide be set up? And a tutorial? What online help messages should be generated? What other types of help facilities should be created?

Preliminary versions of a manual, a writing guide, online help messages and a tutorial have been developed. At present tests are on-going with different types of possible authors, such as people from our documentation department and automation experts. Will they accept a tool that restricts their language usage? Does the subset described by the lingware have to be enlarged/reduced? How quickly do the authors learn which grammatical structures are allowed? Is it easy for an author to reformulate a specific non-described grammatical construction? Is the Help offered of any help? In which manner does a reader appreciate or depreciate the restricted language usage? What will be the overall benefit for an organization to use our method? By which factors will this benefit be influenced? Questions that still wait for an answer.

These questions concern the last phase of our development cycle for lingware. Indeed, we are now entering this implementation phase with some of our clients. We hope to report about our experiences at the next conference.

2. Errors at the syntactic level

Errors at the syntactic level deal with the way words are structurally related in a sentence.

2.1. Performance errors
Performance errors occur rarely. We distinguish:

- Omission of a word:
  (28) *de identificeert de code.*

- Doubling of a word:
  (29) *de de agent identificeert de code.*

- Permutation:
  (30) *het agent soorten* instead of *het soort agenten.*

2.2. Competence errors
We distinguish:

- Agreement or concord errors: Errors in which the number and gender of two or more words do not correspond:
  (31) *de faxnummer* instead of *het faxnummer.*

  8% of all errors in the MODIX corpus are agreement errors.

- Punctuation errors:
  (32) *De agent, die de code identificeert.*

- Structure errors: errors due to sentence structures which the author was not allowed to use since they are unknown to the system.

LIST OF REFERENCES

1. Steen, G.J. van der, and Hasselt-van Rijssen, M.M. van, 1990, "Automatisch Vertalen", Journal of Software Research 2.3. 86-94.

2. Eikelenboom, A., and Kusters, E.D.M., 1991, "Automatisch vertalen: steun of ballast?", Journal of Software Research 3.2. 66-72.

3. Kusters, E.D.M., and Steen, G.J. van der, 1991,"Computerondersteuning bij restrictief taalgebruik", Journal of Software Research 3.2. 48-56.

4. Steen, G.J. van der, and Kuil, W.J.J. van der, 1991, "Lingware: de Vertaalhulpmiddelen van de Toekomst". Journal of Software Research 3.3. 3-13.

5. Obermeier, K.K., 1989, "Natural Language Processing Technologies in Artificial Intelligence. The Science and Industry Perspective", John Wiley & Sons, New York,.

6. Keijzer, R., 1991, "Markt voor Natuurlijke Taal Traag naar Miljardenniveau", Automatisering Gids 22.

7. Tol, P.A. van der, and Dijkstra, M., 1990, "MODIX, de eerste applicatie onder VTM", Journal of Software Research 1.2. 33-40.

8. Lehrberger, J., 1986, "Sublanguage analysis", in Grishman R. and Kittredge R. (eds.) "Analyzing Language in Restricted Domains : Sublanguage Description and Processing,", Lawrence Erlbaum Associates, London.

9. Véronis, J., 1991, "Error in Natural Language Dialogue between Man and Machine", International Journal of Man-Machine Studies 35.2. 187-217.

10. Geerts, G., Haeseryn, W., Rooij, J. de, and Toorn, M.C. van den, 1984,"Algemene Nederlandse Spraakkunst", Wolters-Noordhoff, Groningen.

11. Quirk R., Greenbaum, S., Leech, G., and Svartvik, J., 1974, "A Grammar of Contemporary English", London.

12. Flickinger, D., Nerbonne, J., Sag, I., and Wasow, T., 1987, "Toward Evaluation of NLP systems", 25th Annual Meeting of the Association for Computational Linguistics.

13. Way, A., 1991,"A Practical Developer-Oriented Evaluation of Two MT systems", Working Papers in Language Processing 26. University of Essex.

14. Gotlieb, C.C., and d'Haenens, L., 1991, "Machine Translation of Non-Literary Texts: Some Canadian Experiences", Machine Translation 6.1. 21-33.

15. Wammes, H., 1992, "Softwareproducent Uniface streeft een positie na als wereldspeler", NRC Handelsblad 25th of april.

16. Bart, P. van, and Sturm, A., 1987, "Zinsanalyse en de termen die daarbij gebruikt worden", Nijhoff, Leiden,.

17. Renkema, J., 1989, "Schrijfwijzer", SDU uitgeverij, 's-Gravenhage.

APPENDIX: TYPOLOGY OF ERRORS

1. Errors at the lexical level

Lexical errors are all errors within a word which do not affect the context.

1.1. Performance errors

We distinguish:

- Graphic errors in which 1 letter is misspelled:
  - substitution:
  (17) *incicatie* instead of *indicatie;*
  - insertion:
  (18) *aaantal* instead of *aantal;*
  - omission:
  (19) *lnd* instead of *land;*

Kusters and Van der Steen (3), identified 32% of all the errors in the MODIX corpus as this type of error.

- errors in which a group of letters is at issue:
  - transposition of two subsequent letters:
  (20) *anatal* instead of *aantal;*
  - errors concerning a larger group of letters:
  (21) *aanlat* instead of *aantal;*

- a syllable affected by an error:
  - omission:
  (22) *gebruikersgevens* instead of *gebruikersgegevens;*
  - doubling:
  (23) *gebruikersgegegevens* instead of *gebruikersgegevens;*
  - transposition:
  (24) *gebruikersvegegens* instead of *gebruikersgegevens;*

1.2. Competence errors

We distinguish:

- Phonographic errors: This concerns solely user mistakes. The user knows the sound of a particular word but not its correct spelling.
  (25) *geplende* instead of *geplande;*

- Morphological errors: In Dutch this merely concerns errors with deviating plural forms. Morphological errors may also be due to system errors if the lexicon lacks the correct word form.
  (26) *datums* instead of *data*

- Word errors: These are system errors: the dialog between man and computer is disturbed due to the fact that a word is missing from the lexicon.

- Word segmentation: This may concern a word being incorrectly split into two or more words or two or more words that are incorrectly joined together.
  (27) *rekening nummer* instead of *rekeningnummer*

  Segmentation errors occur in 14% of the errors made in the MODIX corpus.