

[*International Conference on the State of Machine
Translation in America, Asia and Europe. Proceedings of
IAI-MT86, 20-22 August 1986, Bürgerhaus, Dudweiler*]

Lisette Appelo

The Machine Translation System ROSETTA

Philips Research Laboratories
P.O. Box 80 000
5600 JA Eindhoven
The Netherlands

Concrete data about finished and planned Rosetta systems:

name of system	Rosetta2	Rosetta3*	Rosetta4
finished in	1985	± 1988	± 1991
status	research	research	research
type of system	interlingual (with isomorphic grammars)	interlingual (with isomorphic grammars)	interlingual (with isomorphic grammars)
speed:	1-3 seconds per word	unknown	unknown
type of analysis output:	semantic derivation tree (cf. section 2.2)	semantic derivation tree	semantic derivation tree
source languages	Dutch, English	Dutch, English,	Dutch, English,
target languages	Dutch, English,	Dutch, English (Spanish**)	Dutch, English, Spanish
size of dictionaries	5 000 entries	90 000 entries**	unknown
size of grammars after conversion into Pascal	14 000 lines of Pascal for one language pair	unknown	unknown

(all systems are implemented in Pascal on a VAX11/780 under VMS)

(*) The development of Rosetta3 is partially sponsored by NEHEM (Nederlandse Herstructureringsmaatschappij).

(**) For the development of the monolingual and bilingual Rosetta dictionaries for Dutch and English we will use the Van Dale dictionaries for Contemporary Dutch, Dutch-English and English-Dutch, which have approximately 90 000 entries. Obviously, the Rosetta dictionaries will have to contain more detailed and formalized information than dictionaries designed for a human user. Presumably we will only be able to make these extensions for a subset of the original dictionaries. The Spanish dictionary of Rosetta3 will be very small, it will have the minimal size needed for testing the Spanish grammar.

THE MACHINE TRANSLATION PROJECT ROSETTA

Lisette Appelo and Jan Landsbergen
Philips Research Laboratories
Eindhoven - The Netherlands

1. Introduction

In this paper we will give an outline of the approach to machine translation pursued in the Rosetta project, which takes place at Philips Research Laboratories.

After four years of small-scale preparatory research on machine translation, resulting in two experimental translation systems, Rosetta1 and Rosetta2, a fairly large project has started in 1985, in which both linguists (of the University of Utrecht) and computer scientists (Philips Research) participate. The project has a short-term goal, Rosetta3, and a long-term goal, Rosetta4. In the Rosetta3 phase of the project we will restrict ourselves mainly to the linguistic aspects of translation. Rosetta3 will have sophisticated grammars and large dictionaries, but it will translate sentences and phrases in isolation and will not make use of knowledge of the world. Because of these restrictions Rosetta3 will in many cases not be able to solve ambiguities and will therefore produce a set of "possible translations" for an input phrase ("possible" in a particular context). Rosetta4 will be focused on a particular application domain. It will have to give one translation; in order to be able to select the right one it must have knowledge of some specialized domain, furthermore it will translate texts instead of isolated phrases.

Instead of (or in addition to) providing the system with knowledge of the world, selection of the best translation can be achieved by "inter-editing", i.e. by involving the user of the system in the disambiguation process. This approach, in which there is an interactive dialogue with the user, appears to be useful for specific applications: applications in which the user is not a professional translator, but the author of the text to be translated. The interaction takes place during the analysis of the source text and therefore can be formulated completely in terms of the source language. From a pure research point of view it is very interesting whether a system can be constructed which is able to translate a source text adequately into a target text with this kind of interaction (with a possibly large - but finite - number of choices offered by the system to the user). Both Rosetta3 and Rosetta4 will offer the possibility of interactive disambiguation during analysis. The systems will be developed for Dutch and English as source languages and Dutch, English and Spanish as target languages.

It should be stressed that Rosetta is a research project and that the systems to be developed are not planned to be commercial products.

In section 2 of this paper we will discuss the Rosetta approach by formulating five principles on machine translation that we adopt and by showing how the global design of the Rosetta systems follows from these principles. In section 3 the approach is illustrated by an example from the domain of temporal expressions. In section 4 the results are summarized. In an appendix some concrete data about the Rosetta systems are given.

2. The principles of the Rosetta project

The approach to machine translation adopted in the Rosetta project, usually called the isomorphic grammar approach, can be characterized by five principles. These are "working principles", intended to be helpful for systematic research on translation and for the actual construction of translation systems. They should not be interpreted as falsifiable claims about language or about human translation. We will first enumerate the principles and then discuss each of them in more detail.

1. Principle of Explicit Grammars. Both the source language and the target language are defined by explicit grammars.
2. Compositionality Principle. The grammars are organized in such a way that the meaning of an expression is a function of the meaning of its parts.
3. One Grammar Principle. The analysis and the generation component for a particular language are based on the same grammar.
4. Isomorphy Principle. Two sentences are considered translations of each other if their meaning is derived in the same way from the same basic meanings.
5. Principle of Interlinguality. Analysis and generation components of various languages map into and from the same intermediate language.

2.1 The Principle of Explicit Grammars

The first principle states that the Rosetta systems are based on explicit grammars of both source language and target language. This presupposes that a sharp distinction is made in the systems between knowledge of language and knowledge of the world. It is due to this distinction that it is possible to start the development of a translation system with a phase in which only the linguistic aspects are taken into consideration, in the way sketched in the introduction.

It is fairly widely accepted now that translation systems must be based on an explicit grammar of the source language, but in general these systems do not contain a completely explicit grammar of the target language. In most cases the target language is - partially - defined indirectly, by means of contrastive transfer rules that specify the differences with the source language, in terms of surface structures, e.g. in METAL (Bennett and Slocum, 1985), or in terms of deep structures, e.g. in GETA (Vauquois and Boitet, 1985) and TAUM-AVIATION (Isabelle and Bourbeau, 1985). We think it important to have an independent criterion for correctness of the target text, not only for reasons of theoretical elegance, but also because of the intended interactive use of the system. The interaction will be entirely in terms of the source language. In applications where no additional post-editing of the target text is possible, the quality of the generated target text is very important. Note that this does not only concern the syntactic quality of the output, but primarily the semantic quality, i.e. the extent to which the output text means the same as the input text.

2.2. The Compositionality Principle

The Compositionality Principle, adopted in the field of Montague Grammar (cf. Thomason, 1974 and Dowty et al, 1981), can be expressed as follows (cf. Partee,

1984):

The meaning of an expression is a function of the meaning of its parts and of the way they are syntactically combined.

Obviously, following this principle will lead to an organisation of the syntax that is strongly influenced by semantic considerations. As it is an important criterion of a correct translation that it is meaning-preserving, this seems to be a useful guiding principle in machine translation. For an extensive discussion of the status of the Compositionality Principle in Montague Grammar we refer to Partee (1984). Here we will sketch the kind of compositional grammars used in the Rosetta systems, called M-grammars.

An M-grammar consists of three components: a syntactic component, a semantic component and a morphological component.

The syntactic component defines surface trees of sentences. The surface trees used in Rosetta, called S-trees, are ordered trees of which the nodes are labelled with syntactic categories and attribute-value pairs that bear other morpho-syntactic information. The branches are labelled with syntactic relations. S-trees are used as intermediate representations as well.

The syntactic component defines the set of correct surface trees by specifying (1) a set of basic expressions (i.e. basic S-trees), (2) a set of syntactic rules. These rules make it possible to derive new S-trees and ultimately surface trees of sentences from the basic expressions. The rules have "transformational power", they may perform various operations on S-trees.

The process of deriving a surface tree starting from basic expressions by applying syntactic rules recursively, in a "bottom-up" way, can be represented in a syntactic derivation tree.

In figure (1) a simple example of a syntactic derivation tree is given, for the sentence "he was working yesterday". The generated S-trees are not spelled out here, but paraphrased in English.

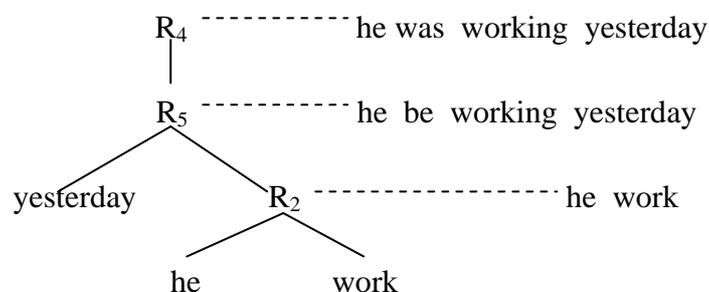


Figure 1. Syntactic derivation tree of (the surface tree of) the sentence "he was working yesterday".

The leaves of a complete surface tree correspond to the words of the sentence, but they have the form of categories and attribute-value pairs. The morphological component relates these leaves to actual, symbol strings.

On the basis of the definitions of the syntactic and the morphological component a function SYNTACTIC GENERATION can be defined which operates on an arbitrary syntactic derivation tree (a tree labelled with names of rules and basic expressions) and yields a set of sentences as follows. First the rules in the derivation tree are applied; the ultimate result is a set of S-trees. If the derivation tree is well-formed, i.e. if all rules are applicable, this set is non-empty. Subsequently the morphological component is applied to the leaves of the generated S-trees; the result is a set of sentences. In section 2.3. we will

show what role the function SYNTACTIC GENERATION may play in an MT system.

The semantic component of an M-grammar specifies the meanings of the basic expressions and of the rules of the grammar. In Montague Grammar these meanings are expressed in intensional logic. In the Rosetta system the meanings of rules and basic expressions are not elaborated in a logical language, but they are represented by means of unique names (of course, in the documentation of the system the definitions of the meaning in terms of logic are useful). The consequence is that a meaning of a sentence can be represented as a so-called semantic derivation tree: a tree with the same geometry as the syntactic derivation tree but labelled with names of meaning rules and basic meanings instead of syntactic rules and basic expressions. In figure 2 an example of a semantic derivation tree is given, corresponding to the syntactic derivation tree of figure 1. Here M_4 is the meaning rule corresponding to syntactic rule R_4 , YESTERDAY is a basic meaning corresponding to "yesterday", etc.

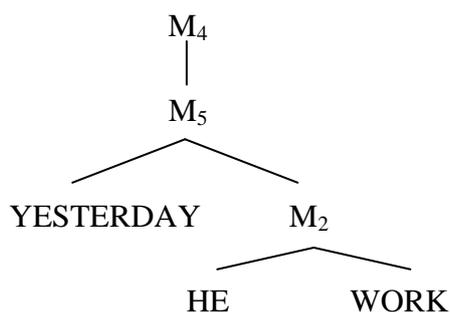


Figure 2. A semantic derivation tree.

As basic expressions may have various meanings, there is in general a set of semantic derivation trees corresponding to a syntactic derivation tree. As a basic meaning may correspond to various basic expressions and a meaning rule may correspond to various syntactic rules, there is in general a set of syntactic derivation trees corresponding to each semantic derivation tree. Clearly, this kind of semantic representation has been chosen for the purpose of machine translation, as we will see in section 2.5. Note that the representation is on the one hand less informative than an explicit logical expression, but on the other hand gives information about the way in which the meaning is derived. This information is lost after translation into logic.

2.3. The One Grammar Principle

The One Grammar Principle states that one and the same grammar is used for defining both the generation component and the analysis component of a language. In other terms, we require the compositional grammar defined above to be "reversible", because of this we sometimes refer to this principle as the Reversibility Principle.

Because of this principle M-grammars have to obey certain conditions. The most important condition is that for each compositional syntactic rule there must be a reverse analytical rule. For a more extensive discussion of these conditions we refer to Landsbergen (1984). Thanks to these conditions analysis algorithms can be defined which yield for any input sentence the set of syntactic derivation trees of that sentence.

In addition to theoretical motives (the idea that there is one notion of language, neutral with respect to generation and analysis), there are economic motives for adopting the One Grammar Principle. If we plan to make translation systems that translate both from and into a particular language, it is efficient

if these systems can be based on one grammar.

In the previous section we have seen that for an M-grammar based on the Compositionality Principle an effective function SYNTACTIC GENERATION can be defined, which maps syntactic derivation trees into sets of sentences. If the One Grammar Principle is obeyed, the reverse function SYNTACTIC ANALYSIS can be defined which yields for a sentence the set of syntactic derivation trees.

Before discussing the Isomorphy Principle it is good to notice that translation systems can be defined based on the basis of the three principles discussed up till now, as follows:

- Write M-grammars for two or more languages,
- Each M-grammar defines an analysis component and a generation component in the way described before, which relate sentences and syntactic derivation trees.
- For each language pair: define a transfer component which transforms derivation trees of the source language (SL) into derivation trees of the target language (TL).
- The composition of the analysis component for SL, the transfer component from SL to TL, and the generation component for TL yields a translation system from SL into TL, as outlined in figure 3.

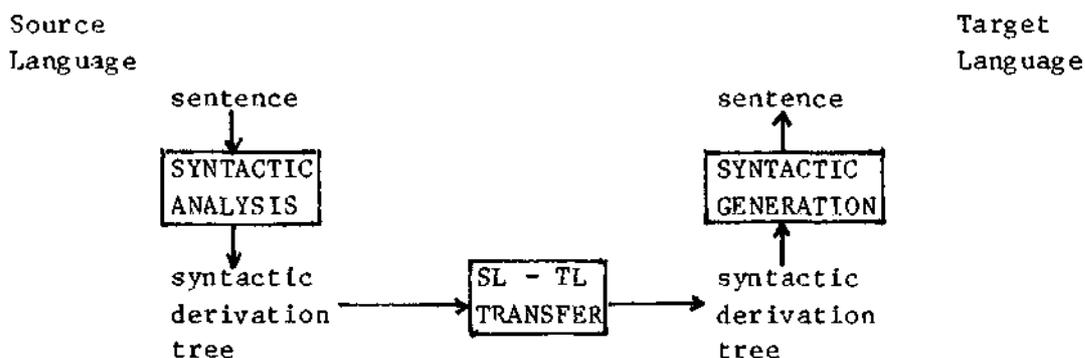


Figure 3. A transfer system based on compositional grammars of source language and target language.

This is a possible approach, but it has two disadvantages: (1) in general rather complex structural transfer will be necessary; it may be difficult to organize this in a transparent way, (2) it is hard to ensure that after transfer the resulting derivation tree (more exactly: one of the resulting derivation trees) is well-formed. The rules need not be applicable. So, the generated sentences are indeed correct sentences of the target language, but in many cases the generation component may act as a filter and no translation will be given. The Isomorphy Principle is a way to tackle this problem.

2.4. The Isomorphy Principle

This principle, which may also be called the Compositionality Principle of Translation, states: two sentences are considered translations of each other if their meanings are derived in the same way from the same basic meanings. Now that we have introduced the notions of syntactic and semantic derivation tree this principle can be expressed in a more technical way. Two sentences are considered translations of each other if they have the same semantic derivation trees, i.e. corresponding syntactic derivation trees. A sentence and its translation are derived from corresponding basic expressions by applying corresponding rules (Here "corresponding" should be interpreted as "with the

same meaning").

Following this principle comes down to attuning the grammars to each other in such a way that for each basic expression in one grammar there is at least one corresponding basic expression in the other grammar with the same meaning and similarly for each rule in one grammar there is at least one corresponding rule in the other grammar. So, in the two grammars there are corresponding sets of rules, related to the same meaning rule, and corresponding sets of basic expressions, related to the same basic meaning. Two grammars are called isomorphic if these corresponding sets of rules obey certain applicability conditions.

In a transfer system based on such "isomorphic grammars", the transfer component becomes simple: rules and basic expressions of the source language are translated locally into sets of corresponding rules and basic expressions. The translation is successful if at least one of the resulting derivation trees is well-formed. Guaranteeing that this is always the case is not trivial, but much easier than in a system with structural transfer, as described before. It may be interesting to make a comparison with EUROTRA (Johnson et al, 1985) at this point. Simple transfer has always been one of the objectives of that project, but it is hard to realize this in an organisation where the various language groups must be able to develop their grammars independently. In the Rosetta approach the requirement of simple transfer can be fulfilled because we are prepared to pay the necessary price: attuning the grammars of the various languages to each other.

The Isomorphy Principle is the most characteristic principle of the Rosetta approach: it leads to a translation project in which the main effort is: writing isomorphic grammars in parallel, for a particular set of languages. One of the attractive aspects of the approach is that translation can be studied from a purely compositional linguistic point of view. In section 3 this will be illustrated for the example of temporal expressions.

At first sight it may seem questionable whether a non-trivial translation relation can be defined on the basis of local relations between rules and basic expressions. The next points may illustrate why the approach is feasible.

- The rules and the basic expressions are chosen with translation in mind.
- The rules have transformational power, which implies that sentences may have corresponding derivational histories, but quite different surface structures. In addition, the rules may introduce - syncategorematically - words that are not basic expressions, e.g. determiners and auxiliaries.
- Basic expressions need not correspond to words, but may be syntactically complex. E.g., an idiomatic expression like "to lose one's temper" may be considered as having one primitive meaning, but syntactically it is to be considered as a compound S-tree, which displays its structural properties. For a more detailed discussion of idiomatic expressions in Rosetta we refer to Schenk (1986).
- Basic expressions may refer to "deeper", more abstract notions than those denoted by words as well. Examples are notions concerning tense and aspect, which are more amenable to local translation than auxiliaries are.

It should be kept in mind that the framework described here serves to define possible translations, as defined in section 2.1. If this system is enriched with knowledge of the world, the transfer rules may no longer be purely local, in the sense that their applicability may depend on the context. E.g., what the more plausible translation of a verb is may depend on the type of its arguments.

Notice that even after extensions like this there is no structural transfer, only a refinement of the local transfer.

2.5. The Principle of Interlinguality

The Interlinguality Principle states that an intermediate language IL is defined, into which analysis components of various languages translate and from which the generation components of these languages are able to translate. If we combine this principle with the Principle of Isomorphy, the main consequence is that the attuning of the grammars is now done not just for two, but possibly for a larger number of grammars and that semantic derivation trees constitute the intermediate language. Formally, the only difference with the system described in 2.4. is that the transfer component is split up into a component ANALYTICAL TRANSFER defining the translation from syntactic derivation trees into semantic derivation trees by means of local translation rules as presented by the grammar of the source language, as sketched in section 2.2, and a component GENERATIVE TRANSFER which performs the reverse function for the target language. This results in a system design as outlined in figure 4.

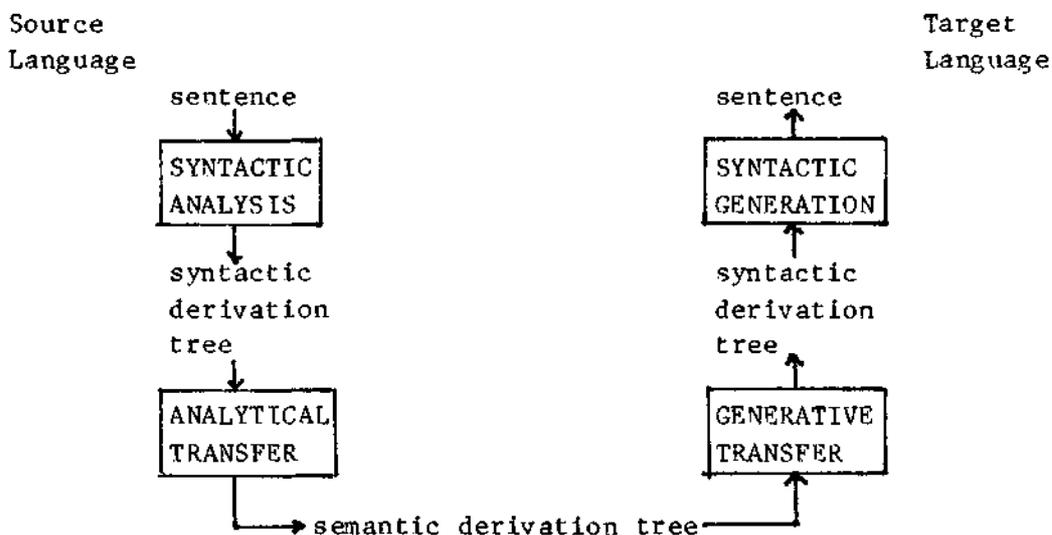


Figure 4. Global design of the Rosetta systems.

It will be obvious that in this framework there is not much difference from a formal point of view between local transfer systems and interlingual systems. The main difference in practice is whether the tuning of grammars is done for a pair of languages or for a larger set of languages. The advantage of the interlingual approach is that the collective tuning results in a translation system for each language pair. Of course a price has to be paid for this: tuning grammars to each other becomes more difficult when the set of grammars becomes larger. The main problem is that grammars of languages A and B and consequently the translation from A into B, may become more complex than necessary for A and B, because peculiarities of language C have to be taken into account. One of the goals of the Rosetta project is to find out how high this price is, for the languages Dutch, English and Spanish. It should be stressed that the isomorphy and not the interlinguality is the primary characteristic of our approach.

2.6. Deviations from the principles

The before-mentioned principles should guide us in designing translation systems in a systematic way, but they should not be interpreted as dogmas. In fact,

there are several reasons to deviate from the principles in building practical systems. We will mention three of them:

- It is inevitable that the Isomorphy Principle is violated for many basic expressions. Corresponding words of two languages often have slightly different meanings. Translation is not possible without some tolerance in this respect.
- The Principle of One Grammar will be violated in a practical system, because rules, or basic expressions, of which the only effect is the enlargement of the set of paraphrases in the target language, can better be eliminated from the generation component.
- Robustness measures are needed to deal with ill-formed input, unknown words etc. They tend to be in conflict with several of the principles.

2.7. Recent modifications

The framework described here is in fact the Rosetta2 framework. Rosetta3 will be based on the same principles and will have the same global design, but on a more detailed level there are several differences. One of them is worth mentioning here: in Rosetta3 a distinction will be made between "meaningful rules" and "syntactic transformations". Syntactic transformations are rules with one argument that do not change the meaning of the expression they operate on and that do not carry any other information relevant for translation. Syntactic transformations are not involved in the isomorphy relation and may therefore be introduced freely in each individual grammar.

Precautions are necessary for guaranteeing the effectiveness of both analysis and generation algorithms for grammars with syntactic transformations. These measures partially coincide with measures for a better organisation of large grammars, i.e. the introduction of subgrammars and the possibility of ordering rules explicitly. A discussion of these extensions is outside the scope of this rather informal paper. An example of a syntactic transformation is the "verb second" rule for an SOV language like Dutch, which places the finite verb at second position in the main sentence. This transformation rule is used in the example in section 3.

3. Examples concerning the translation of temporal expressions

In this section we will illustrate the approach sketched above by an example which concerns the temporal expressions in sentences.

Temporal expressions are:

- tense, a morphological form of a verb (simple tense), e.g. works, worked, or a combination of morphological forms of an auxiliary verb and a verb (periphrastic tense), e.g. has worked, is working
- time adverbials, ADVP's, PP's, NP's etc. expressing some notion of time, e.g. yesterday, for three hours.

To obtain isomorphic M-grammars with respect to temporal expressions in accordance with the principles explained in the previous sections, we have to define for each language:

- (i) the temporal expressions that are introduced categorically,
- (ii) the syntactic rules that combine temporal expressions and/or introduce temporal expressions syncategorically.

Due to the reversibility of the grammars (One Grammar Principle) a purely compositional definition of the translation relation will be sufficient.

The temporal expressions and the rules of each language have to be attuned to each other according to the Isomorphy Principle. This is not a trivial matter as can be illustrated by the following examples:

- a) the different use of tenses in languages

Dutch: (1) Hij woont hier
(he-lives-here)

English: (2) He is living here
(3) *He has been living here

Dutch: (4) Hij woont hier nu al 20 jaar
(He-lives-here-already-20-years)

English: (5) *He lives here for 20 years now
(6) He has been living here for 20 years now

The pairs (1)/(2) and (4)/(6) are considered to be translation equivalents. The Dutch simple present of (1) has to be translated into the English Present Progressive Tense in (2), but the same Dutch tense in (4) has to be translated into a Present Perfect Progressive Tense in (6). Sentence (3) is not a correct translation of (1), while (5) is incorrect English.

- b) an adverbial is not always translated into an adverbial

English: (7) He has just arrived
Spanish: (8) El acaba de llegar
(He-finishes-to-arrive)

The examples (7) and (8) are considered translation equivalents. In (7) the adverbial just is used in combination with a Present Perfect Tense, while in (8) the verb acabar de with an infinitive form expresses the same concept. This latter combination can be seen as a periphrastic tense.

The meanings of the rules and the time adverbials are defined with respect to a time model T which consists of a set of time points ordered by the relation " \leq ", meaning "earlier than". For this model an interval S, the moment of speech, and an interval E, an interval to which the event which is expressed by the clause is associated, are defined. In addition there are so-called reference intervals. Temporal expressions express properties of these intervals and relations between them. For a more detailed account of this model and the relation to other research in this area we refer to Appelo (1986).

The part of the M-grammar that deals with temporal expressions in clauses can be globally specified as follows:

The rules start to operate on a clause without any temporal expression, they add temporal expressions to the clause; the result is a clause with temporal expressions.

The rules are subdivided into five classes, which may be optional (the whole class may be skipped) or obligatory (one rule of the class has to be applied). The rules have either one argument, the clause, or two arguments: 1) the clause and 2) a temporal expression that is introduced categorically.

We divide the time adverbials into two types: durative (e.g. during three hours) and referential (e.g. yesterday).

The rule classes are (in "generative" order):

I. Aktionsart rules: optional rules that can change the Aktionsart of a clause by inserting some "auxiliary" verb. They imply certain temporal properties of the event.

II. Duration rules: optional rules that insert a certain type of durative temporal expression. Semantically, the length of the interval E is specified. Neither Aktionsart rules nor Duration rules will be discussed in the examples given here.

III. Aspect rules: obligatory rules that spell out aspectual (possibly periphrastic) tense forms of the verb if such forms exist in that language and insert a referential temporal expression (that may be abstract or anaphoric). We will indicate these rules as R-ASP_i. Semantically, they express properties of a reference interval (RE) and the relation between that interval and the interval E.

IV. Retrospective rules: optional rules that spell out (possibly periphrastic) retrospective tense forms and/or time adverbials if necessary in that language and insert a referential temporal expression (that may be abstract or anaphoric). We will indicate these rules as R-RETRO_i. Semantically, they express properties of a reference interval and a relation between that interval and R_E.

V. Deictic rules: obligatory rules that spell out deictic tense forms. We will indicate these rules as R-DEIXIS_i. Semantically, they express the relation between S and a reference interval.

The isomorphic derivation trees with respect to the temporal expressions for the translation equivalent pair (4)/(6) are shown in figure 5. Only the parts of the derivation trees that are relevant for this example are specified. For the other parts, the ADVPs, the PPs and the CLs we only specify the resulting S-trees, abbreviated as triangles.

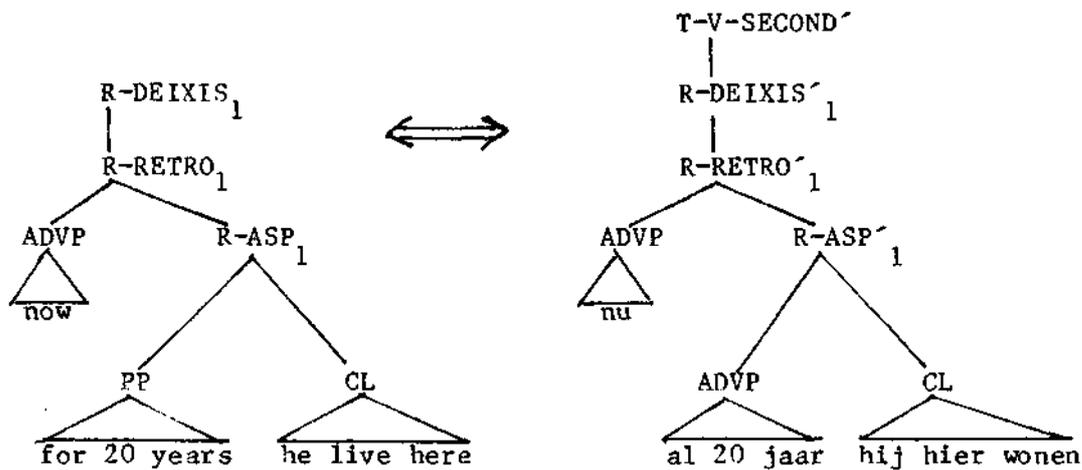


Figure 5. Isomorphic derivation trees of the English sentence "he has been living here for 20 years now" and the Dutch sentence "hij woont hier nu al 20 jaar".

For the English example the rules start to operate on the clause he live here.

In this example no Aktionsart rules and no Duration rules are applied.

R-ASP₁ is an imperfective aspect rule that inserts a temporal expression for 20 years and the auxiliary verb be and spells out the ing form of the verb: he be living here for 20 years. Semantically, R-ASP₁ specifies the length of the interval R_E and it indicates that R_E is a subset of E.

R-RETRO₁ is the retrospective rule that inserts the adverbial now and the auxiliary verb have and spells out the past participle form of the first verb: he have been living here for 20 years. Semantically, the rule specifies that R_E lasts until a reference interval R_S of which the properties are specified by the adverbial.

R-DEIXIS₁ is the present deictic rule for main clauses and spells out the present tense form of the first verb: he has been living here for 20 years, which is the output: a clause with temporal expressions. Semantically, the rule expresses that R_S is simultaneous with S.

The clause he live here corresponds to the Dutch clause hij hier wonen. The temporal expressions for 20 years and now correspond to the Dutch temporal expressions al 20 jaar and nu respectively. The Dutch rules corresponding to R-ASP₁, R-RETRO₁ and R-DEIXIS₁ in this derivation are R-ASP'₁, R-RETRO'₁ and R-DEIXIS'₁ respectively.

The Dutch rule R-ASP'₁ differs from the English rule R-ASP₁ in that it does not insert an auxiliary for the imperfective form as in English. It results in: hij hier al 20 jaar wonen.

R-RETRO'₁ differs from R-RETRO₁ also with respect to the introduction of the auxiliary verb. In Dutch it is not necessary to insert hebben/zijn if the clause has imperfective aspect. R-RETRO'₁ results in: hij hier nu al 20 jaar wonen.

R-DEIXIS'₁ is similar to R-DEIXIS₁.

In Dutch the finite verb of main clauses is placed in second position by a language specific transformation T-V-SECOND': hij woont hier nu al 20 jaar. This transformation does not have a counterpart in the English derivation tree as English is treated as an SVO language.

The example illustrates that corresponding rules can have quite different syntactic effects, cf. the imperfective aspect rules and the retrospective rules

in English and Dutch, and that the rules are syntactically rather powerful.

Note that in Dutch another retrospective rule exists (R-RETRO'₂), i.e. the retrospective rule for clauses with perfective aspect, which inserts an auxiliary verb (hebben/zijn) just like the English rule and spells out a participle form for the verb. Both R-RETRO'₁ and R-RETRO'₂ correspond semantically to the same retrospective rule (R-RETRO₁) of English. So, the local transfer from English to Dutch will translate R-RETRO₁ into two Dutch rules. Which of the Dutch rules is applicable depends on the input and therefore - indirectly - on the rules that have been applied before (in this case on the aspect rule).

As a final illustration of the consequences of the isomorphic approach we will consider the aspect rules. In Dutch these rules do not alter the verb form, because Dutch has no aspectual tense forms. This results in a systematic ambiguity of Dutch tense forms with respect to other languages as is illustrated by the past tense examples (9) - (11):

- Dutch (9) hij werkte gisteren
 English (10) a. he worked yesterday
 b. he was working yesterday
 Spanish (11) a. él trabajó ayer
 b. él trabajaba ayer

The isomorphic derivation trees for these examples with the names of the rules at the nodes and abbreviated resulting S-trees are shown in figure 6 (the perfective examples (9), (10a), (11a)) and figure 7 (the imperfective examples (9), (10b), (11b)). R-DEIXIS₂, R-DEIXIS'₂ and R-DEIXIS''₂ are the corresponding past deictic rules for main clauses and R-ASP₂, R-ASP'₂ and R-ASP''₂ the corresponding perfective aspect rules.

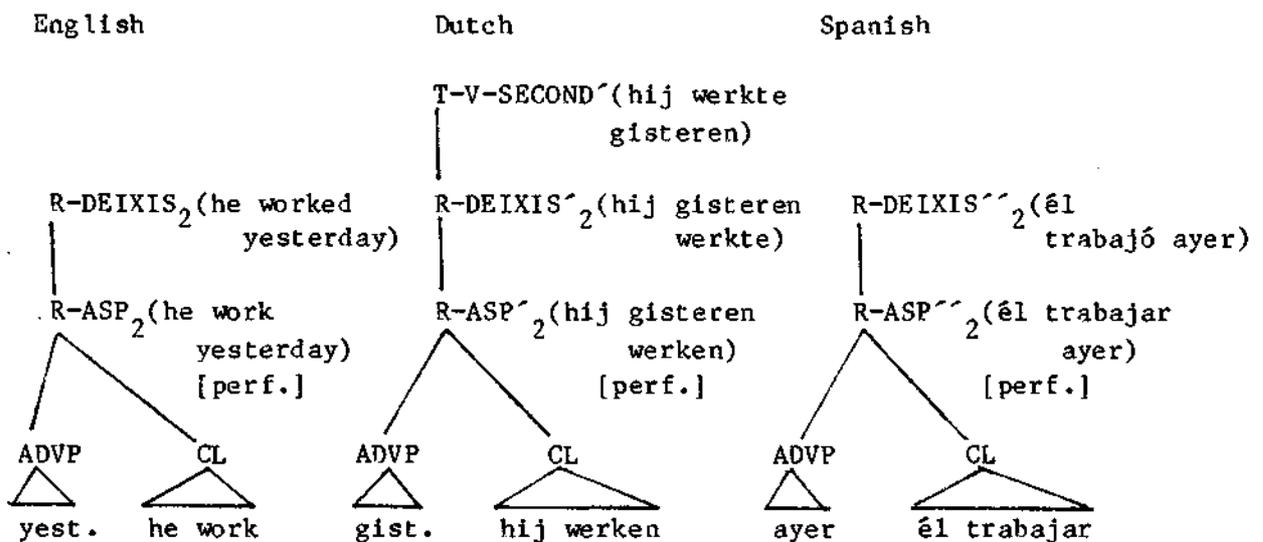


Figure 6. Syntactic derivation trees of (9), (10a) and (11a).

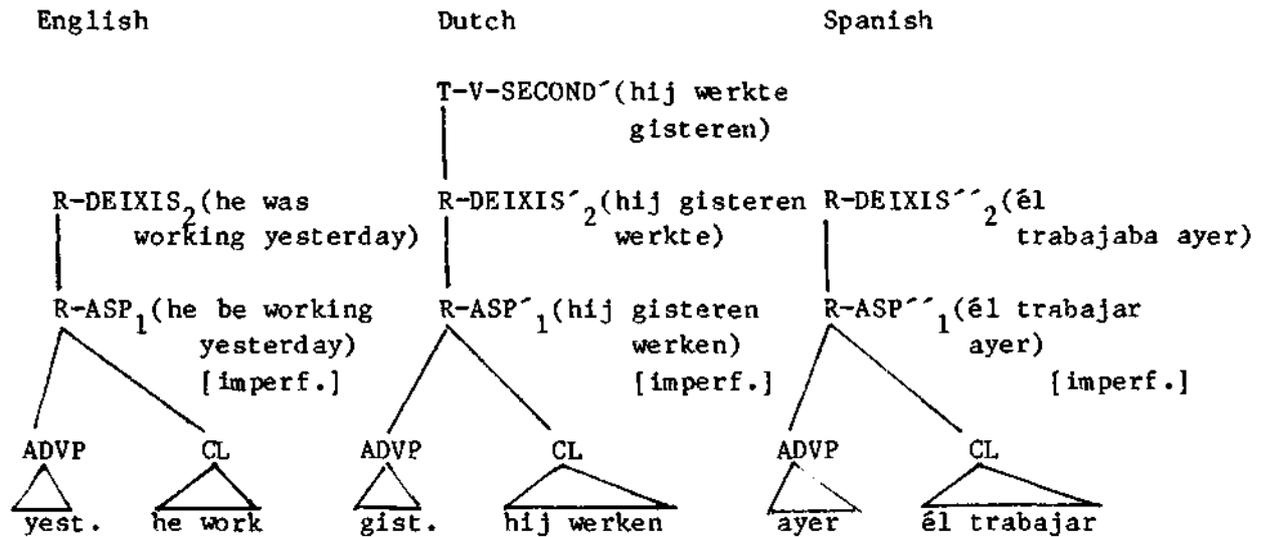


Figure 7. Syntactic derivation trees of (9), (10b) and (11b).

This example illustrates in what way a grammar of a particular language may be influenced by the grammars of other languages. If the only goal would have been to write a compositional grammar for Dutch, a different - non-ambiguous - solution might have been chosen.

4. Conclusion

In this paper we have given an outline of the linguistic framework for machine translation adopted in the Rosetta project. We have described the basic principles underlying this framework and indicated how the design of the Rosetta systems follows from these principles.

The most characteristic of these principles is the Isomorphy Principle, which enables us to define the translation relation between sentences of different languages in terms of a simple relation between their derivation processes. This principle can only be formulated in a framework where explicit grammars for both source and target language are formulated and where, moreover, these grammars are compositional, in the sense of Montague Grammar. The Isomorphy Principle is only useful for the development of translation systems, if these compositional grammars can also be used for analysis, which is expressed by the One Grammar Principle. The attuning of grammars as required by this approach can in principle be done for more than two languages. The Interlinguality Principle induces us to investigate this aspect as well.

Finally, we have tried to give an impression of the kind of linguistic research done in the Rosetta framework, by discussing a few examples concerning temporal expressions in Dutch, English and Spanish.

We have chosen to present a rather informal paper which concentrates on the general principles of our approach and to illustrate this by means of examples. More detailed information about the linguistic formalism and the design of Rosetta can be found in Landsbergen (1982, and especially 1984). For a presentation of the Rosetta approach from a somewhat different angle, as the result of a stepwise refinement of the transfer component, we refer to Leermakers and Rous (1986).

References

- Appelo, L. (1986), "A Compositional Approach to the Translation of Temporal Expressions in the Rosetta System", Philips Research M.S. 13.677, paper to be presented at the 11th International Conference on Computational Linguistics, August 25 - 29, 1986, Bonn.
- Bennett, W.S. and J. Slocum (1985), "The LRC Machine Translation System". In: Computational Linguistics 11(2-3): 111-122.
- Dowty, D.R., R.E. Wall and S. Peters (1981), Introduction to Montague semantics (Reidel, Dordrecht).
- Isabelle, P. and L. Bourbeau (1985), "TAUM-AVIATON: Its Technical Features and Some Experimental Results". In: Computational Linguistics 11(1): 18-28
- Johnson, R., M. King and L. des Tombe (1985), "EUROTRA: A Multilingual System under Development". In: Computational Linguistics 11(2-3): 155-170.
- Landsbergen, J. (1982), "Machine translation based on logically isomorphic Montague grammars", in: Horecky, J. (ed.), COLING 82, North-Holland, pp 175-182.
- Landsbergen, J. (1984), "Isomorphic grammars and their use in the Rosetta translation system", Philips Research M.S. 12.950, paper presented at the Tutorial on Machine Translation, Lugano. To appear in: King, M. (ed.), Machine Translation: the state of the art, Edinburgh University Press.
- Leermakers R. and J. Rous (1986), "The translation method of Rosetta", Philips Research M.S. 13.701, presented at the Int. Conf. on Machine and Machine-aided Translation, April 7-9, 1986, Aston University, Birmingham.
- Partee, B.H. (1984), "Compositionality". In Landman, F. and F. Veldman (eds.), Varieties of Formal Semantics, Foris Publications, Dordrecht.
- Schenk, A. (1986), "Idioms in the Rosetta Machine Translation System", Philips Research M.S. 13.508, paper to be presented at the 11th International Conference on Computational Linguistics, August 25 - 29, 1986, Bonn.
- Thomason, R.H. (ed.) (1974), Formal Philosophy, Selected papers of Richard Montague (Yale University Press, New Haven, 1974).
- Vauquois B. and C. Boitet (1985), "Automated Translation at Grenoble University" In: Computational Linguistics 11(1): 28-37.