THE TRANSLATION METHOD OF ROSETTA

R. Leermakers and J. Rous
Philips Research Laboratories
Eindhoven - The Netherlands

(presented at the International Conference on MACHINE and MACHINE-AIDED TRANSLATION, April 7 - 9, 1986, Aston University, Birmingham)

CONTENTS

1 Introduction

Machine translation of natural language is an effort which touches upon various fields and subjects. Linguistics comes in to shed light upon morphology and syntax, and to offer some insights on semantics. However, certain aspects of semantics, like knowledge representation and automated reasoning, belong to the realm of artificial intelligence. Necessarily, machine translation involves the design and implementation of large systems, and the principles of software engineering should not be neglected.

Not very surprisingly, contrasting schools of thought have evolved, all emphasizing different aspects of machine translation. One judges achievements by looking at the performance of actual systems, the other by looking at the amount of theoretical understanding which has been gained. Some hold the A.I. approach to semantics to be more promising than linguistic methods, or vice versa. Also people differ in opinion about the relative importance of dictionaries on the one hand and grammars on the other. One school tries to integrate syntactic and semantic information, while others separate the two as much as possible.

In this paper we will explain and motivate the translation method of the Rosetta project. We will do so by presenting a stepwise way of unravelling the various aspects of machine translation. The general strategy is to view a translation system as composed of an analysis part and a generation part connected by a transfer module, and to systematically break down the latter. We do this by repeatedly identifying tasks inside the transfer module, which can be moved as new modules, with well-defined interfaces, into the (initially empty) analysis and generation parts. Thus, by each such move analysis and generation are augmented with a deeper level in a clear way, and the transfer task is reduced accordingly.

Subsequently, it is shown how the use of compositional and isomorphic grammars can help to simplify the translation problem. Compositional grammars establish a close relationship between syntactic and semantic structures of sentences. The way in which the form of a sentence is built up from the smallest units runs parallel to the way in which the meaning of that sentence is constructed from the meaning of these smallest units. Two such grammars for two different languages are called isomorphic when they are attuned to each other in such a way, that the process of constructing a sentence in one language can be parallel to the process of constructing the translation of that sentence in the other language. We will explain how transfer complexity trivializes after the analysis of sentences in terms of isomorphic grammars. In fact, if the set of derivation trees of such analyses is viewed as an intermediate language, the system

effectively becomes interlingual. Isomorphic grammars form the core of the Rosetta system, and yield the relation between a sentence and the set of all of its possible translations without a semantic analysis in the A.I. sense. Here "possible translations" is to be defined as the set of translations of a sentence considered in isolation. The interlingua of the Rosetta method, however, is a good starting point for deeper analyses, as interlingual expressions can in principle be transformed into logical expressions.

2 First-level Transfer

The translation system with the most simple external structure one can think of is the one depicted in fig. 1, consisting of one component only. The task of this component would be to transfer source language (SL) sentences into target language (TL) sentences.

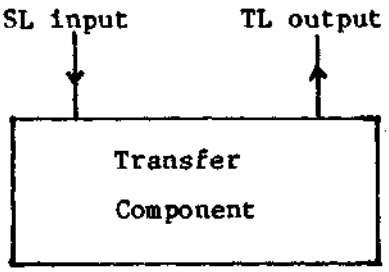


fig.1

Whereas such a system may seem preferable from an efficiency point of view, it will not excell in transparancy. For instance, it is difficult to clarify how the various levels of analyses are connected. In our opinion it is, in the present stage of MT, very important to have a transparent system in the sense that well-defined theoretical functions can be associated with each part of the system. Such a modular system is also desirable from a software engineering point of view, and it gives a solid base to incorporate future findings of linguistic and A.I. research into.

The first aspect of machine translation that we can identify in our quest for separable tasks is morphology. In order to limit the length of the dictionaries the transfer component of fig. 1 will contain some morphological knowledge in the form of morphological rules. Figure 2 gives the structure of a somewhat more structured translation system. It contains an analytical and generative morphological component which provide mappings betwwen words and lexical structures.

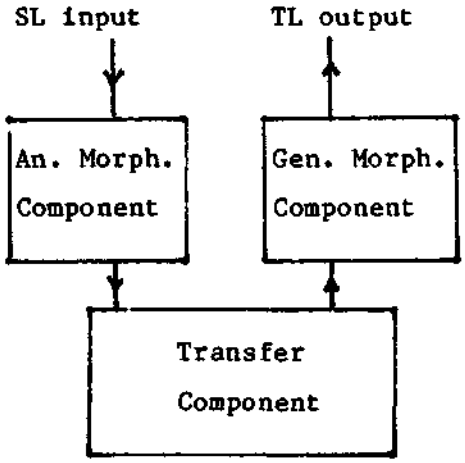


fig. 2

In most cases these mappings will be one to many. The mappings are effected by consulting a dictionary and by using morphological rules for inflection and derivation. A lexical structure contains information about the word's category, its tense, features discerning transitive and intransitive verbs, mass and common nouns, etcetera. For instance the sentence

(1) all bishops like her

would be analyzed as

(2) [DET: stem: all]
[NOUN: stem: bishop
number: plural]
[VERB: stem: like
persons: [1sing,2sing,1pl,2pl,3pl]
tense: present]
[CONJ: stem: like]
[PERSPRO: stem: she
case: accusative]

where the word "like" is analyzed ambiguously. Thus the task of the morphological component in analysis is to disambiguate the input, in the sense that it has to identify all morphological structures. The transfer component in the system of fig. 2 is to translate sequences of SL lexical structures into sequences of TL lexical structures. The generative morphological component will

turn these structures into strings of the target language. In the example (1) the translation into Dutch is

(1a) Zij bevalt alle bisschoppen

derived from the lexical structures

```
(2a)[PERSPRO: stam: zij
              naamval: nominatief]
    [VERB: stam: beval
           persoon: 3enkelvoud
           tijd: ott]
    [DET: stam: alle]
    [NOUN: stam: bisschop
           getal: meervoud ]
```

The translation complications are four-fold:

(a) The word "like" is ambiguous, and it should only be translated as a VERB.

(b) The Dutch word-order differs considerably from the English word-order.

(c) There is a non-trivial translation between lexical structures, e.g. the cases of the words "her" and "zij" are accusative and nominative, respectively.

(d) The word "bishop" has two meanings, and it should be translated into "bisschop", and not into "loper", which is the piece of chess.

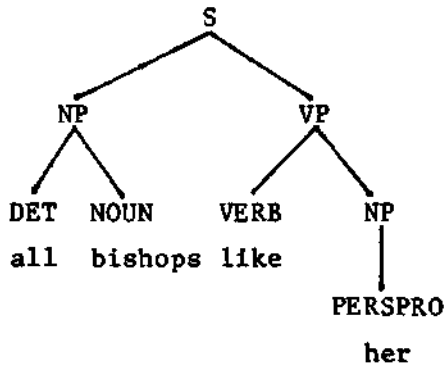
3 Second-level Transfer

Many of the structures assigned to a word by the morphological component in analysis can be rejected by looking at its context within the sentence. The transfer component of fig. 2 will have to filter out all sequences of lexical structures that make no sense. In the first place this transfer component has to know about the possible "surface structures" of the sentence, i.e. the possible groupings of words, even if surface structures of the languages in question are completely parallel.

As before, we consider it clarifying to move this transfer task, identifying the possible surface structures, to a separate component. One then gets the translation system depicted in fig. 3.

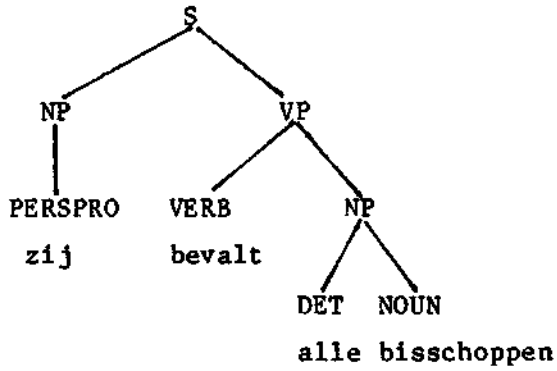
In the analysis one has a surface parser, which parses a sequence of lexical structures according to a surface grammar. The output is a surface structure syntax tree with at the leaves the input lexical structures. For instance, (2) would be analyzed as (3).

(3)



Here the lexical structure sequence containing [like:CONJ] was filtered out. Of course, in general, the surface parser will generate new ambiguities because many sentences containing unambiguous words only, still have several readings due to various possibilities of grouping words together. The task of the surface parser may thus be viewed as to disambiguate the input at a second level. It has to assign all possible surface structures to the output of the morphological component. The task of the transfer component of fig 3. is to translate a SL syntax tree into one which is correct according to the surface grammar of the target language. The surface structure of the example sentence (1a) would be (3a).

(3a)



This TL syntax tree is then mapped onto lexical structure sequence (2a) by the generative surface syntax component.
Notice that indeed one of the complications mentioned in the previous section (namely complication (a)) has disappeared after surface analysis.

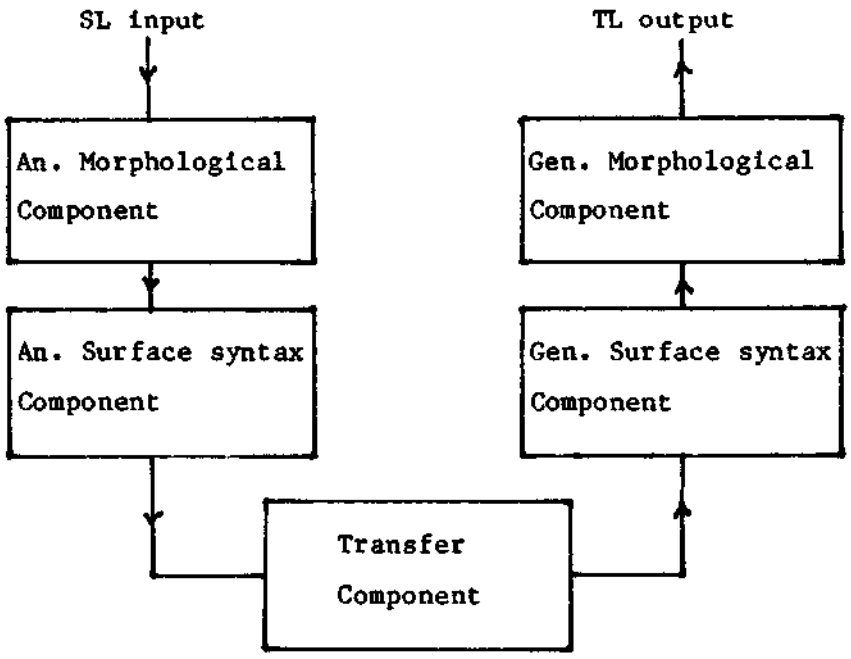


fig. 3

4 Third-level Transfer

A fundamental reason for looking for a representation of a sentence that is "deeper" than the surface structure is that if a translation is being performed at surface tree level it is very difficult to guarantee that the translated sentence indeed has the same meaning, because it is not obvious how to deduce the meaning of a sentence directly from its surface structure.

For instance, "all bishops" and "alle bisschoppen" have the same semantic rôle in the sentence, often called the agent. However, this rôle is realized syntactically different in both languages, i.e. "all bishops" is a syntactic subject, whereas "alle bisschoppen" is a syntactic object.
In most linguistic theories, semantics is attached to a "deep" syntactic structure (GPSG is a well-known exception (Gazdar(1985))). Whereas surface structures are generally accepted representations of sentences, there is no consensus about the form of deep structures. Among the possible candidates are Chomskyan deep structures (Chomsky(1981)), LFG F-structures (Bresnan(1982)) and Montague derivation trees (Montague(1974)). They have in common that they are tree-like structures, but they have different kinds of information at nodes and leaves. In any case, a transfer after deep-structure analysis should no longer be troubled by complications like (b) and (c) mentioned in section 2. The reader is referred to section 6 for Rosetta deep structure representations of sentences (1) and (1a).
In fig. 4, new modules have been added to the system, that intermediate between surface structure and deep structure. The transfer module of the fig. 4 system is to translate from SL deep structures to TL deep structures.

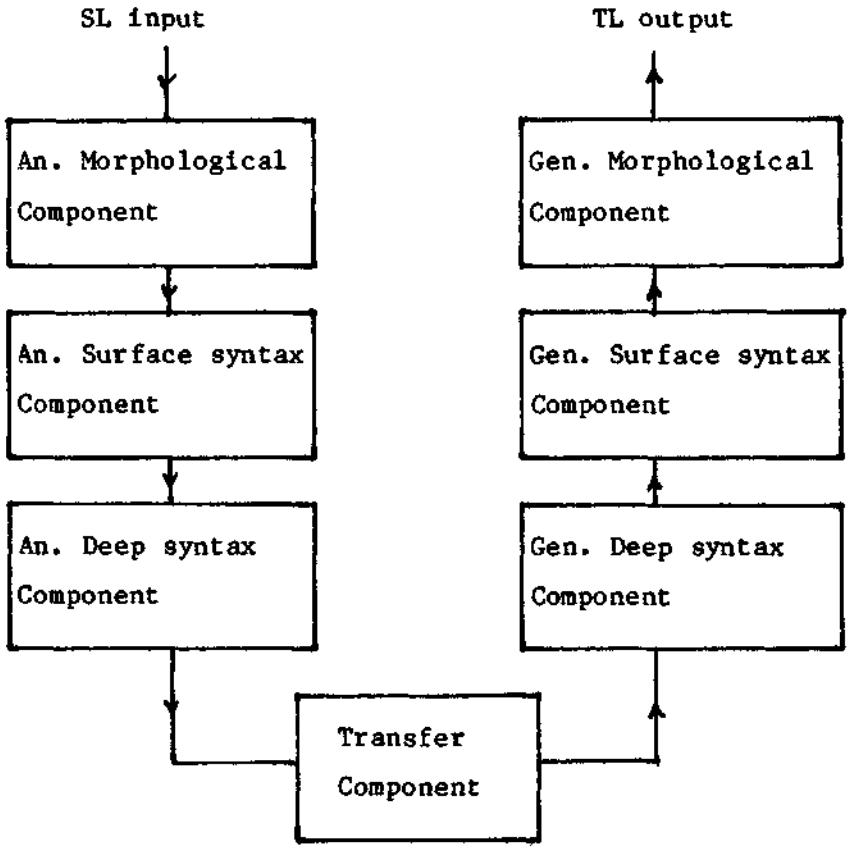

SL input
TL output
An. Morphological Component
Gen. Morphological Component
An. Surface syntax Component
Gen. Surface syntax Component
An. Deep syntax Component
Gen. Deep syntax Component
Transfer Component

fig. 4

5 Fourth-level Transfer

A transfer at the present level is still burdened by problems like complication (d) of section 2. Filtering out wrong translations now requires a semantic analysis. The precise form of semantic analyses depends on the kind of deep structures that is being used. In general, however, such structures will provide some kind of predicate-argument analysis of the sentence, and as a semantic analysis one can then do type checks to see wether predicate and arguments fit together.

In order to once more simplify the transfer task, we introduce again a new level in analysis and generation to obtain the system structure of fig. 5. The new modules intermediate between the deep structures and structures which serve as meaning representations. Analogously to the foregoing, the task of the semantic module in analysis would be to disambiguate, i.e. to find all consistent, e.g. type-checked, meaning representations from the deep structures fed into it. Type checking can only be formulated within some theoretical framework. One such framework is offered by a formal semantics, in particular by the model-theoretical semantics as proposed by Richard Montague (Montague(1974)). Also, some work has been done to establish model-theoretical interpretations for LFG F-structures (cf. Halvorsen (1983)).

In our example, type-checking should help to overcome complication (d). If "like" is considered a two-place predicate of which the agent argument type should be "living being", then the interpretation of "bishop" as a cleric would be preferred to its interpretation as a piece of chess.

In the Montague tradition, deep structures are first transformed into semantic structures, which are then interpreted model-theoretically. It seems natural to mimic this in a translation system and to have the semantic modules intermediate between deep structures and such semantic structures. For translation purposes, these structures will be close enough to genuine meaning representations, most of the time. That is, preservation of meaning can be guaranteed without an analysis in terms of logical formulae and without semantic representations of sentence contents in the traditional A.I. sense. This point is one of the central tenets of the Rosetta method.

It is reasonable to expect that after the 4-step disambiguation that we described the transfer should be easy enough to be conducted in a transparent way. We will see that in the Rosetta method it even becomes trivial.

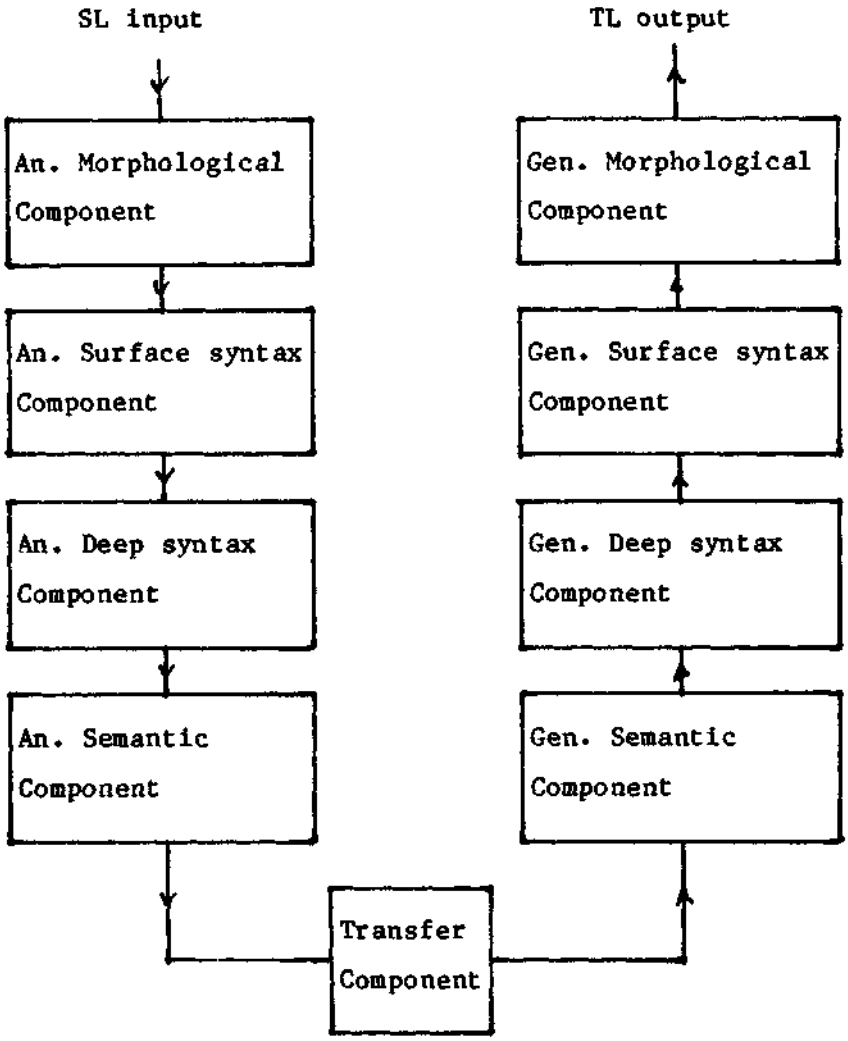

SL input
TL output
An. Morphological Component
An. Surface syntax Component
An. Deep syntax Component
An. Semantic Component
Transfer Component
Gen. Semantic Component
Gen. Deep syntax Component
Gen. Surface syntax Component
Gen. Morphological Component

fig. 5

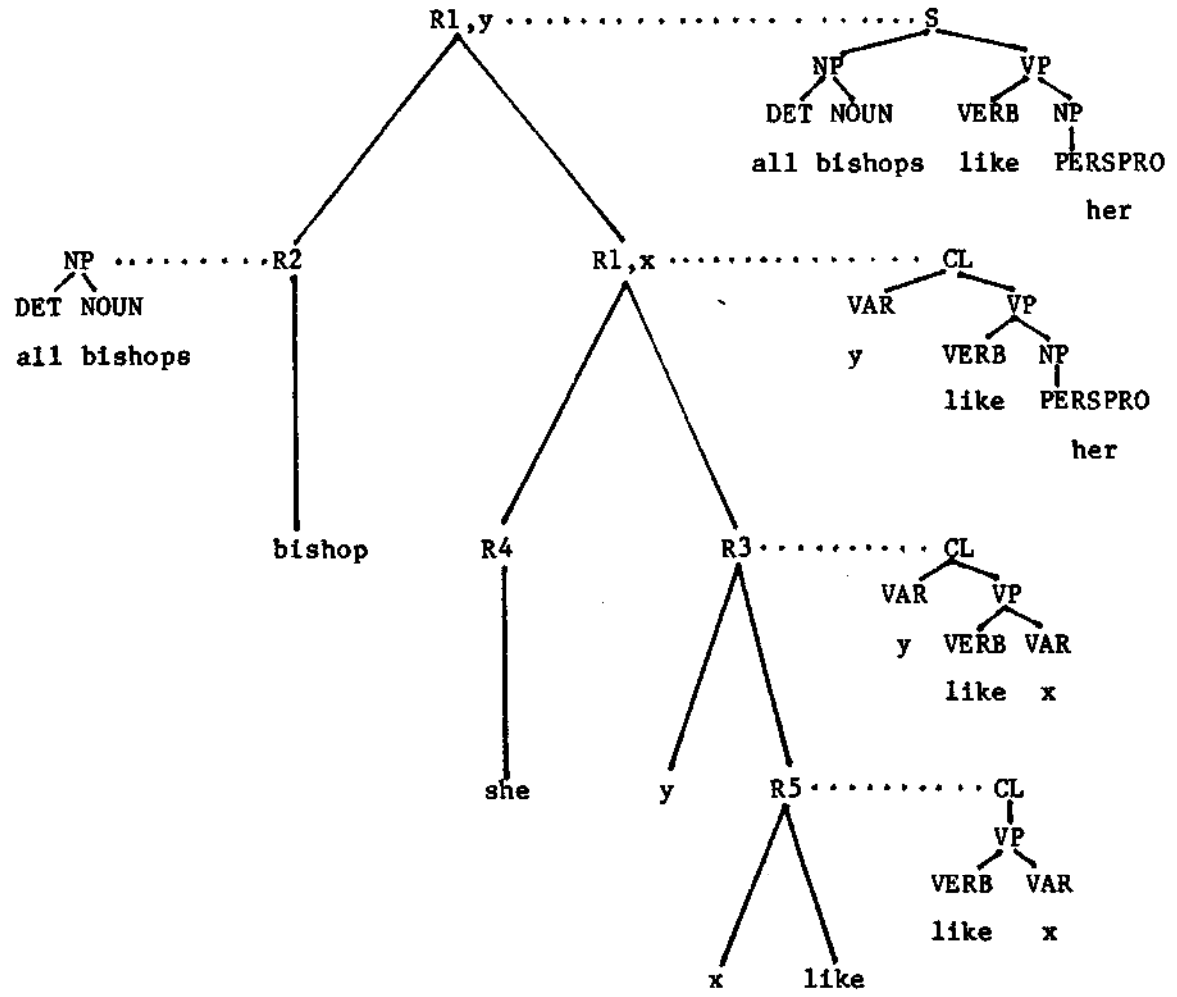
6 Rosetta

We are now in the position to discuss the principles of the Rosetta method and we will, furthermore, use these principles to fill out in more detail the general translation system depicted in fig. 5. The discussion will show in which way the Rosetta method deviates from other methods used in the field of Machine Translation.

6.1 M-grammars and Compositionality.

In the Rosetta system the deep structure analysis and generation component are based on so called M-grammars, introduced by Landsbergen (1982, 1984). M-grammars are closely related to the grammars described by Montague (1974). Grammars of this type obey Frege's Compositionality Principle. This principle says that a sentence is ultimately composed of a number of basic expressions (e.g. word stems), which are combined according to syntactic rules of the grammar into successively larger expressions. Furthermore, this syntactic process of putting things together to build a larger structure is mirrored semantically, i.e. to each basic expression corresponds a basic meaning and to each syntactic rule a semantic rule. Whereas the syntactic rule puts together expressions in order to build a larger expression, the corresponding semantic rule builds the meaning of the larger expression from the meaning of the argument expressions.
In the specific case of M-grammars, the structures figuring in the syntactic process are very similar to surface structures. At intermediate stages they may contain syntactic variables, and the order of constituents may differ from surface orders, but ultimately the correct surface structure of the sentence is constructed.
For instance, the way in which the surface structure (3) of "all bishops like her" is derived can be represented in a tree, a so called syntactic derivation tree:

(4)



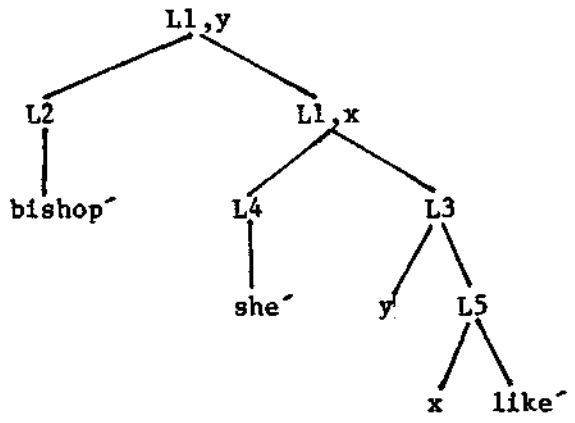
Such a syntactic derivation tree is labeled with names of syntactic rules (in this case R1,..,R5), which are called M-rules, at the non-terminal nodes and with names of basic expressions (bishop, she, like, x and y) at the leaves. In the picture above we have added a number of intermediate (surface-like) structures to give a better insight in the derivation process. Rule R5 should be interpreted to insert the semantic object, whereas R4 puts in the semantic subject (agent).

Hence, an M-grammar defines a mapping from syntactic derivation trees to surface structures. Also the reverse mapping, i.e. from surface structures to derivation trees is determined by an M-grammar. In fact, an analytical rule exists for each compositional one. It will be clear that in the Rosetta method, syntactic derivation trees serve as deep structures. The analytical module of the system, which maps surface trees unto derivation trees, is called the M-parser; the reverse component is named M-generator.

We can now describe the consequences of the compositionality principle and the

M-grammar approach for the semantic component. As mentioned before, it is the task of the semantic component to generate meaning representations for each deep structure. We already described that in M-grammars there is a close correspondence between syntax and semantics. That is, for each syntactic rule there is a semantic rule, which expresses its meaning and to each basic expression corresponds at least one basic meaning.
As an example, let the meanings of the rules R1,..,R5 be given by L1,..,L5, and let the meanings of "bishop", "she" and "like" be denoted by bishop´, she´ and like´. Then a node-to-node translation of (4) yields a semantic derivation tree for "all bishops like her":

(5)



Actually applying the rules in this tree would yield the following logical expression:

(6) FORALL(x)[bishop´(x)->like´(x,she´)]

which means: "For all individuals who are bishops there is one individual she´ that belongs to the set of individuals that are being liked by the bishop". Semantic derivation trees like (5) represent the derivation process of the meaning of the sentence. In Rosetta, the related logical formulae are not derived. Instead, the semantic derivation trees themselves are used as meaning representations. The function of the analytical semantic component of Rosetta is to map syntactic derivation trees onto semantic derivation trees. The generative semantic component performs the reverse operation; it maps semantic derivation trees onto syntactic derivation trees. Furthermore, the semantic component performs a check on the argument types of the predicates and orders the ambiguities on the basis of decreasing plausibility.

Summarizing, we can say that the module M-parser generates for each surface

structure a set of syntactic derivation trees, which are mapped onto semantic derivation trees by the semantic component. In generation the reverse process takes place. Notice that the task of the semantic component has become very transparent because of the fact that M-grammars obey the compositionality principle. Lastly, in the Rosetta system semantic derivation trees serve as meaning representations.

6.2 M-grammars and Isomorphy

In a translation system as described in fig. 5 the transfer component plays a very important role. It has to map meaning representations of the source language onto meaning representations of the target language. However, this operation can be very complex, because the semantic primitives used for source language meaning representation and those used for target language meaning representation may differ considerably.

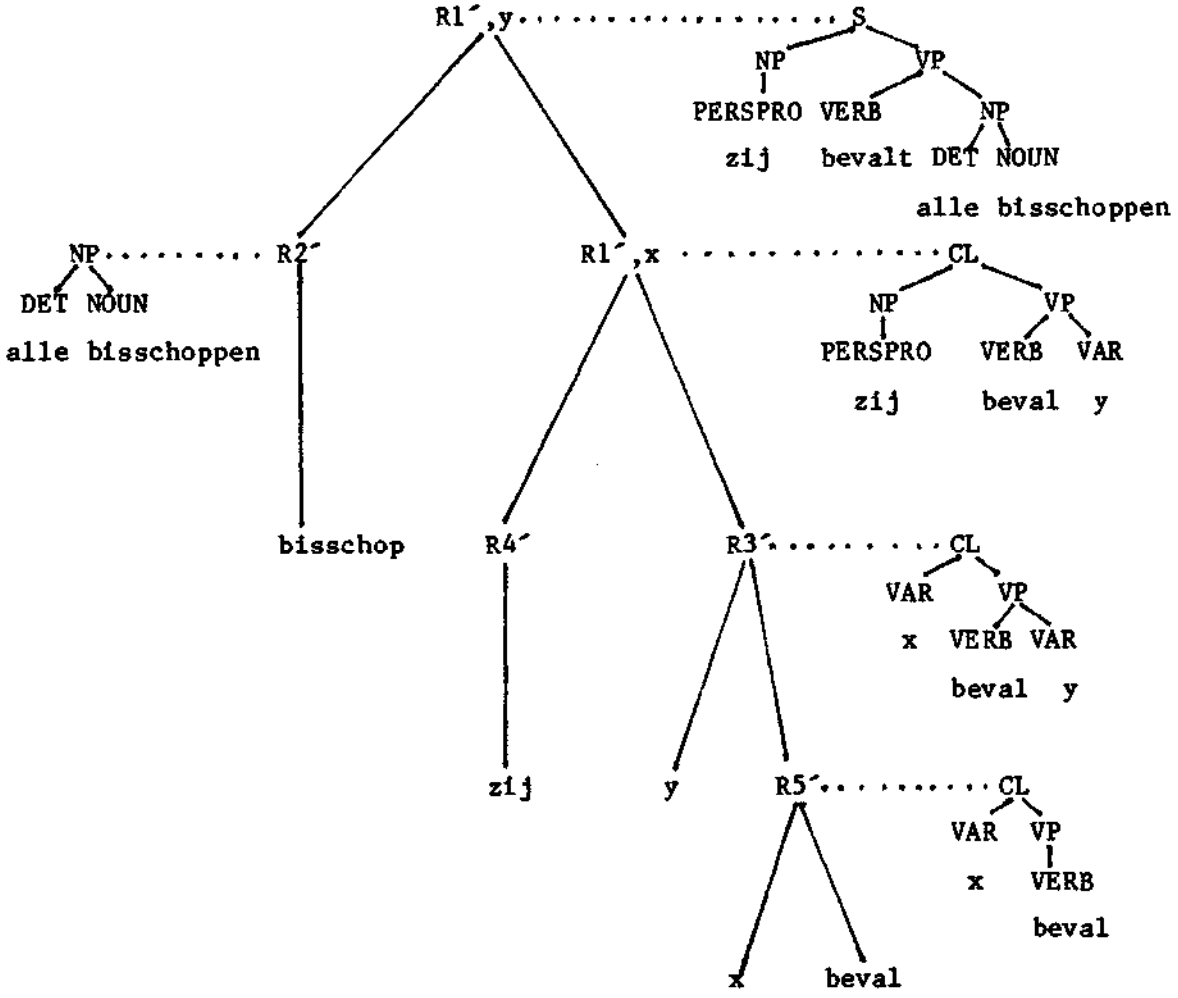
This problem has been solved elegantly within the Rosetta system by adhering to the Rosetta Isomorphism Principle. This principle prescribes M-grammars of source and target language to be isomorphic. This means that the SL and TL M-grammars have to be attuned to each other. Namely, for each syntactic rule of the SL M-grammar, which corresponds with a semantic rule expressing its meaning, there is at least one syntactic rule in the TL M-grammar with the same meaning. An SL syntactic rule and a TL syntactic rule having such a correspondence, can in principle be translated into each other.

If isomorphic M-grammars are used, the transfer component is trivial, because the semantic rules which express the meaning of SL syntactic rules are essentially the same as the semantic rules expressing the meaning of TL syntactic rules. Therefore, a semantic derivation tree, which serves as meaning representation of a SL sentence, is also the meaning representation to start with in the generation process of the TL.
In effect, the isomorphism of SL and TL M-grammars changes the general translation system depicted in fig. 5 from a transfer system into a interlingual system.
This brings us to a requirement any proper translation system should satisfy, which is the requirement of meaning preservation. It can be formulated as the demand that each sentence of the SL must have at least one meaning in common with its translated sentences in the TL. It is obvious that in the Rosetta system this condition will always be satisfied, because the SL sentence and the

translated TL sentence share the same meaning representation in the form of a semantic derivation tree.

Consider now a translation of (1) into Dutch. This translation is accomplished by translating the semantic derivation tree of fig. 5 into the following syntactic derivation tree of the Dutch language:

(7)



Both the English and the Dutch derivation tree have a similar form because the M-grammars are isomorphic. The syntactic differences between the two sentences are effected by differences between the M-rules R1,..,R5 and R1´,..,R5´. Although syntactically diverse, the M-rules have pair-wise identical meanings, i.e. R1 and R1´ have the meaning L1 whereas R2 and R2´ correspond to L2, etcetera. In particular, whereas R5´, just as R5, inserts the semantic object, syntactically it insert the subject.

6.3 M-grammars and Reversibility.

One of the most important advantages of the "interlingua" approach is the fact that if N analysis components and N generation components have been developed, in fact N^2 translation systems are available. To obtain the same number of systems in the "transfer" approach, in addition N^2 transfer components are needed. In both approaches still an analytical version and a generative version of the grammar of each language has to be written. In the Rosetta system, however, each language is described by one grammar from which an analytical and a generative system component is derived. In particular, the analytical syntactic rules and the generative versions are each others reverse. Actually, this is in line with the isomorphism principle, because both the analytical and the generative version of a rule correspond with the same meaning rule.

This Reversibility Principle is followed throughout the Rosetta system, e.g. also in the morphological and semantic components. In general, for each analytical rule there is a reverse generative rule and for each analytical dictionary there is a generative dictionary which is its reverse as far as the possible-translation relation is concerned.

In practice, the analytical version of the surface grammar rules have no reverse generative counterparts. Hence, the Rosetta surface syntax components seemingly violate the reversibility principe.
In analysis, the task of the surface parser is to build a surface tree structure on top of the lexical structure by repeatedly applying surface rules. The reverse task in generation which should theoretically be to systematically break down the surface tree, meanwhile checking whether the tree satisfies the TL surface syntax rules. Because the definition of the M-generator is such that it only generates correct surface trees, another check on the correctness would be absolutely redundant. Therefore it is formally correct to shortcircuit the process of breaking down the surface tree by simply offering its leaves to the generative morphological component.

The reversibility principle, if used correctly, guarantees a symmetry of the possible-translation relation for each language pair L1, L2, i.e.

$$[\ e \text{ in } \text{ANALYSIS}_{L1}(s) \text{ and } s' \text{ in } \text{GENERATION}_{L2}(e)\ ] \iff$$
$$[\ e \text{ in } \text{ANALYSIS}_{L2}(s') \text{ and } s \text{ in } \text{GENERATION}_{L1}(e)\ ]$$

This relation says that if a sentence s' is a translation of the sentence s in a L_1->L_2 translation system, then s will be a translation of s' in a L_2->L_1 translation system.

Of course, it is the ultimate goal of a translation system to give the best possible translation. The best-translation relation, which in itself is not symmetric, can be obtained from the symmetric possible-translation relation by imposing a plausibility ordering on the set of possible translations. Obviously, the problem of finding the correct ordering of possible translations in a given context is far from trivial.

7 Conclusions

We have presented machine translation systems at various levels. Our general strategy was to reduce the transfer complexity. We did this by rendering implicit tasks done by transfer modules explicit by moving them as new modules into the analysis and generation parts of the system, and by introducing the concept of isomorphic compositonal grammars, which was shown to be a powerful tool for developing transparent translation systems. In this way we arrived at the structure of the Rosetta system, which is effectively interlingual.
The M-grammars of the Rosetta system constitute a level of analysis that is absent in most systems. Such a level is needed for several reasons. One is that one can handle cases where syntactically very different constructions correspond to the same semantic primitive. Another is that at this level one gets a hold on translation quality. Indeed, the translation quality is guaranteed to the degree in which the related basic expressions and M-rules indeed have identical semantics. Lastly, M-rules replace the transformations on surface structure trees needed in lexical transfer systems.

The Rosetta method is based on the hypothesis that isomorphic grammars can be written for reasonably large subsets of languages. In the Rosetta project it is presently tried to further substantiate this claim by actually writing large grammars for Dutch, English and Spanish. Until now, the endeavour has yielded demonstratable Rosetta1 and Rosetta2 systems and the results are very encouraging.

8 References

Bresnan, J. (1982), Ed., The mental representation of grammatical relations, London, MIT Press.

Gazdar, G.; Klein, E; Pullum, G.K.; Sag, I. (1985), Generalized Phrase Structure Grammar, Oxford, Basic Blackwell.

Halvorsen, P.K. (1983), "Semantics for Lexical-Functional Grammar." Linguistic Inquiry, Volume 14, Number 4, Fall 1983.

Landsbergen, J. (1982), "Machine Translation Based on Logically Isomorphic Montague grammars." In COLING 82. Ed. Horeckey, J., North-Holland, pp 175-182.

Landsbergen, J. (1984), "Isomorphic Grammars and Their Use in the Rosetta Translation system", paper presented at the Tutorial on Machine Translation, Lugano, Philips Research M.S. 12.950, to appear in King, M. (ed), Machine Translation: the state of the Art., Edinburgh University Press.

Montague, R. (1974), Formal Philosophy: Selected Papers of Richard Montague. Ed. Richmond Thomason. New Haven: Yale University Press.