The Power of Compositional Translation

JAN LANDSBERGEN, JAN ODIJK and ANDRÉ SCHENK
Philips Research Laboratories, PO Box 80000, 5600 JA Eindhoven, The Netherlands

1. Introduction

In this paper we will discuss the compositional approach to machine translation that is pursued in the Rosetta project, at the Philips Research Laboratories in Eindhoven, in collaboration with the University of Utrecht. Rosetta is a research project, in which experimental translation systems are being developed, currently for Dutch, English, and Spanish.

In the compositional approach, the translation relation between two languages is defined as a relation between their grammars. These grammars obey the Compositionality Principle of Montague Grammar, i.e. they define a language by specifying (i) a set of 'basic expressions', expressions with a primitive meaning, e.g. content words, and (ii) a set of compositional rules (with well-defined meanings), which prescribe how larger expressions and ultimately sentences can be built from these basic expressions. The compositional translation relation is defined by means of local relations between the compositional rules and between the basic expressions of the two languages.[1] This definition of the translation relation is attractive from a theoretical point of view, especially because it provides a firm semantic foundation to machine translation, but one might be concerned about its actual power in practice. At first sight, it may appear rather restrictive, allowing only for rather trivial translation relations, where a sentence and its translation have more or less the same surface structure.

The main goal of this paper is to illustrate the power of the compositional approach by discussing—rather informally—how a number of non-trivial translation problems which may appear problematic at first sight can be solved. We will not discuss the power of the approach in the formal sense of the word. We expect that from a purely formal point of view the Rosetta system, like many other systems, is able to define any type of translation relation, if one is prepared to pay a high price for this in terms of grammar size. But the question that will be discussed here and that is of more practical interest is what translation problems can be handled elegantly and systematically in the given framework.

Basically, there are two aspects that determine the power of the compositional approach: (i) the power of the rules; this aspect is discussed in Section 3, (ii) the question of what is to be considered a basic expression; this aspect is discussed in Section 4. We start with an informal introduction to the Rosetta approach in Section 2. In Section 5 a translation problem is discussed that cannot be handled satisfactorily in the current Rosetta framework.

The paper deals only with the linguistic aspects of translation, not with the problems of disambiguation in case a sentence has more than one 'possible translation'.

2. The linguistic framework of Rosetta

The linguistic framework of Rosetta can be characterized as follows:

1. The languages are described by means of compositional grammars.
2. These grammars are reversible, i.e. they can be used for both analysis and generation.
3. The translation relation between two languages is defined as a relation between their grammars.

We will discuss these three aspects in more detail in the next two subsections.

2.1. *Compositional grammars*

The grammars of Rosetta obey the Compositionality Principle, adopted in the field of Montague Grammar (cf. Thomason, 1974 and Dowty *et al.*, 1981). This can be expressed as follows (cf. Partee, 1982):

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

Obviously, following this principle leads to an organization of the syntax that is strongly influenced by semantic considerations. As preservation of meaning is an important criterion for correct translation, this is a useful guiding principle in machine translation.

We will illustrate the Compositionality Principle and the way it is applied in Rosetta by means of a simplified example of a compositional grammar.

The **syntactic component** of a compositional grammar specifies (i) a set of basic expressions and (ii) a set of syntactic rules. The basic expressions are the smallest meaningful units, the syntactic rules define how larger phrases and ultimately sentences can be constructed, starting with the basic expressions.

A simple example grammar, G_E

The **basic expressions** are: the noun *car* and the intransitive verb *pass*, or more formally: the expressions N(*car*) and V(*pass*).

The **rules** are:

ER_1: this rule is applicable to an expression of the form N(α) and makes an indefinite noun phrase of the form NP(*a* α). (α is an arbitrary string.)

ER_2: this rule has two arguments, the first one must be a noun phrase, of the form NP(α), the second argument must be an intransitive verb of the form V(β). The result of applying the rule is a sentence in the past tense, of the form S(α β*ed*).

If ER_1 is applied to the basic expression N(*car*), the result is NP(*a car*).

If ER_2 is applied to NP(*a car*) and V(*pass*), the result is the sentence S(*a car passed*).

Note that the words in the derived sentence may correspond to basic expressions (*pass, car*), but may also be introduced syncategorematically, by a rule (e.g. the article *a*).

The actual rules in the Rosetta systems are more complicated; they operate on syntactic constituent trees, which is of vital importance for their linguistic power. We present them here in a simplified format in order to prevent the essential ideas from being obscured by notational complexities.

The process of deriving a sentence from basic expressions by recursive application of rules can be made explicit in a **syntactic derivation tree**. In Fig. 1 the syntactic derivation tree of the sentence S(*a car passed*) is given.

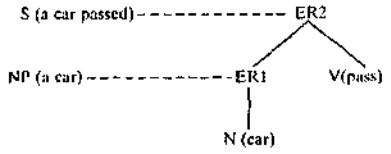

Fig. 1 Syntactic derivation tree of 'a car passed'

An effective procedure GENERATOR 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 by applying the rules in the syntactic derivation tree. If the derivation tree is not well-formed, i.e. if not all rules are applicable, this set is empty.

The **semantic component** of a compositional grammar specifies:

1. The meanings of the basic expressions (basic meanings).
2. The meaning operations corresponding to the syntactic rules.

The meaning of an arbitrary expression is then derived as follows. In parallel with the application of the syntactic rules the meaning operations associated with these rules are applied to the meanings of their arguments, starting with the basic meanings. The final result is the meaning of the complete expression. So the process of derivation of the meaning runs parallel with the syntactic derivation process and can be represented in a tree with the same geometry as the syntactic derivation tree, but labelled with names of basic meanings and meaning operations. This representation is called a **semantic derivation tree**. If we assume that the rules of example grammar G_E correspond to meaning rules, named M_1 and M_2, and the basic expressions correspond to meanings *car'* and *pass'*, the relation between the syntactic and the semantic derivation tree of *a car passed* is as in Fig. 2. If the basic meanings and the meaning operations are expressed in a logical language, as in Montague Grammar, the result of applying the meaning operations in a semantic derivation tree for a sentence is a logical

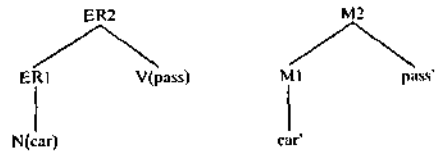

Fig. 2 Syntactic derivation tree of 'a car passed' and corresponding semantic derivation tree

expression, representing the meaning of that sentence. However, for the purpose of compositional translation we do not need this logical expression; the semantic derivation tree contains exactly the information that has to be preserved during translation, as we will show in Subsection 2.3.

Although a compositional grammar requires a close relation between syntax and semantics, this relation is not 'one-to-one'. Basic expressions were defined as the smallest meaningful units, but they may have more than one meaning. For example, a basic expression may correspond to the stem of a word like *pass*, which may have more than one reading. Because of this there is in general a *set* of semantic derivation trees corresponding to a syntactic derivation tree. On the other hand, a basic meaning may correspond to various basic expressions and a meaning rule may correspond to various syntactic rules, so there is in general a set of syntactic derivation trees corresponding to each semantic derivation tree. However, in the example we assume that there is a one-to-one relation.

2.2. *Reversibility*

The grammars used in Rosetta are reversible, i.e. the same grammar can be used for generation and for analysis of sentences. The most important requirement for reversibility of grammars is reversibility of syntactic rules.

For example, grammar G_E (in Section 2.1) the reverse rules would be:

ER'_1: the rule is applicable to an expression of the form NP (α α) and yields an expression of the form N(α).

ER'_2: the rule is applicable to an expression of the form S(α βed) and yields two expressions, the first one of the form NP(α), the second one of the form V(β).

If we apply ER'_2 to the sentence S(*a car passed*), the result is the pair NP(*a car*), V(*pass*). V(*pass*) is a basic expression.

If we apply ER'_1 to NP(*a car*), the result is the basic expression N(*car*).

An analysis is successful if it is able to reduce the sentence to basic expressions, as in this example. The analysis process can be made explicit in a derivation tree, which is the same as the one of Fig. 1.

In the previous subsection we have seen that for a grammar based on the Compositionality Principle an effective procedure GENERATOR can be defined, which maps syntactic derivation trees on to sets of sentences. In case of a reversible grammar, an effective procedure PARSER can be defined which yields for a sentence the set of syntactic derivation trees. If the

sentence is incorrect, this set is empty, if it is syntactically ambiguous the set contains more than one syntactic derivation tree. In the actual Rosetta systems the parser is more complicated, because the syntactic rules operate on constituent structures (cf. Landsbergen, 1981, 1987) for a more elaborate discussion).

2.3. *Isomorphic grammars*

In Rosetta the translation relation between languages is defined in a compositional way. Two sentences are considered translations of each other if (i) they have the same meaning, and (ii) the way this meaning is compositionally derived from basic meanings is the same too.

This definition is attractive because it captures the intuition that translation should preserve the meaning, but also the form, as far as possible. For example, it accounts for the fact that *all ravens are black* is an adequate translation of the Dutch *alle raven zijn zwart* and that the logically equivalent sentence *if something is not black it is not a raven* is not an adequate translation of this sentence. (cf. Landsbergen, 1987; De Jong and Appelo, 1987)

Having introduced the notions of syntactic and semantic derivation tree, this definition of translation can be expressed in a more technical way by means of a relation between compositional grammars:

> Two sentences are considered translations of each other if they have the same semantic derivation trees, and, hence, corresponding syntactic derivation trees.

A sentence and its translation are derived from corresponding basic expressions by applying corresponding rules (where 'corresponding' should be interpreted as 'with the same meaning').

In order to accommodate this technical definition of translation with the above-mentioned intuition about translation, we have to write the compositional grammars with translation in mind. The grammars of two languages have to be *attuned* 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. Grammars that are attuned to each other in this way are called isomorphic grammars. In Landsbergen (1987) more precise definitions of isomorphic grammars can be found. Beaven and Whitelock (1988) adopt the isomorphic grammar approach in a different syntactic framework.

As an illustration we specify an example grammar G_D for a fragment of Dutch, which is isomorphic to grammar G_E in Section 2.1.

Grammar G_D

The basic expressions are the noun N(*auto*) and the verb V(*passeer*). We assign them the meanings *car'* and *pass'* respectively, so they correspond to basic expressions *car* and *pass* of G_E.

The rules are:

NR_1: this rule is applicable to an expression of the form N(α) and makes an indefinite noun phrase of the form NP(*een* α). (α is an arbitrary string.)

The corresponding meaning rule is M_1, so NR_1 corresponds to ER_1.

NR_2: this rule has two arguments, the first one must be a noun phrase, of the form NP(*een* α), the second argument must be an intransitive verb of the form V(β).

The result of applying the rule is a sentence of the form S(*er* β*de een* α).

The corresponding meaning rule is M_2, so NR_2 corresponds to ER_2.

This grammar can derive the sentence *er passeerde een auto*. The syntactic derivation tree is given in Fig. 3. The semantic derivation tree is exactly the same as the one of Fig. 2, and thus the derived Dutch sentence is considered a translation of the English sentence *a car passed*.

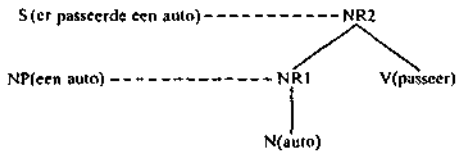

Fig. 3 Syntactic derivation tree of 'er passeerde een auto'

An important aspect of the use of *reversible* isomorphic grammars is that the translation relation can be described in a completely compositional ('generative') way, while this description still yields an effective translation procedure, consisting of an analysis and a generation component. In Fig. 4 the global design of the resulting translation system is outlined.

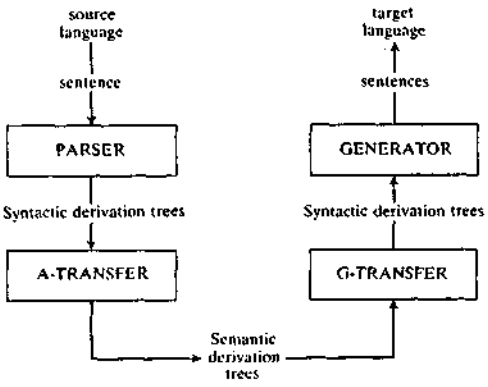

Fig. 4 Global design of the Rosetta system

We will show how the given example sentences are processed by the system, for English as the source language and Dutch as the target language. The translation procedure starts with a syntactic analysis performed

by the procedure PARSER, which applies the analytical rules of G_E to the input sentence (e.g. *a car passed*) and yields a set of syntactic derivation trees for that sentence (e.g. the tree of Fig. 1). Then a procedure A-TRANSFER (analytical transfer) is applied: it converts a syntactic derivation tree into a set of semantic derivation trees (e.g. the semantic derivation tree of Fig. 2) by means of local translation rules for the rules and the basic expressions which follow directly from the semantic component of grammar G_E. For example, syntactic rule ER_1 is mapped onto meaning rule M_1, basic expression *car* is mapped onto basic meaning *car'*. In the example the mapping is one-to-one, but in more realistic grammars it is usually one-to-many. (The semantic derivation trees can be used to perform semantic well-formedness and preference tests in order to solve ambiguities, but this is outside the scope of this paper.)

The next step in the translation process is the application of a procedure G-TRANSFER (generative transfer). G-TRANSFER maps meaning rules onto syntactic rules and basic meanings onto basic expressions, according to the semantic component of grammar G_D. In this way each semantic derivation tree is converted to a set of syntactic derivation trees (e.g. the tree of Fig. 3).

Finally, the procedure GENERATOR of grammar G_D is applied to each syntactic derivation tree. The result is a set of sentences of the target language Dutch, e.g. *er passeerde een auto*.

2.4. *Additional remarks*

We already noted that the grammar formalism of Rosetta is more sophisticated than is outlined here, but for the purposes of this paper it is not necessary to go into the technical details. In this section we will merely indicate in what respects the actual grammar formalism differs from the formalism used in the examples.

1. There is a separate morphological component, containing the detailed rules for word formation. The syntactic rules only have to specify the form of words in a formalized way, e.g. *number = plural*, and do not have to bother about the exact form of the plural, which may depend on the form of the noun. The morphological rules spell out the exact form.

2. The syntactic rules do not operate on symbol strings, as the example rules suggest, but on explicit syntactic structures, so the final result of a derivation is not a sentence, but the surface structure of a sentence.

In the sequel we will usually stick to the informal string notation that was used in the examples and only refer explicitly to the syntactic structure if this is necessary.

3. The grammars are divided into subgrammars and the order of application of rules within a subgrammar can be controlled explicitly. Isomorphy between grammars is defined by means of isomorphy between subgrammars.

4. A distinction is made between meaningful rules and purely syntactic transformation rules. Meaningful rules are involved in the isomorphy relation between the grammars, transformations are not. Therefore, transformations can be added freely to individual grammars to deal with language specific phenomena.

3. Powerful syntactic rules

As described in the preceding section, the translation method used in the Rosetta system is compositional, i.e. a sentence is translated by means of local translation of basic expressions and rules.

The compositional method might appear restrictive at first sight, but we will show that it is in fact very powerful, i.e. that it is able to characterize two sentences as translations of each other even if these sentences differ substantially from each other at a superficial level. The method is powerful, on the one hand because the rules can perform complicated operations, and on the other because basic expressions need not correspond to a single word, but can correspond to more complex phrases.

In this section we will deal with cases where compositional translation can be maintained due to the power of the rules. In all cases discussed here basic expressions correspond to single words. The next section will deal with basic expressions that correspond to more complex phrases.

One powerful property of the rules that plays an important role here is their capacity of *syncategorematically introducing* elements into a structure, i.e. they can introduce elements into a structure that are not arguments of the rule.

Let us consider the following simple example: the Dutch sentence *hij houdt van het meisje* and the English sentence *he loves the girl* are translations of each other. In the English construction there is a verb (*loves*) and a direct object (*the girl*). In the corresponding Dutch sentence there is also a verb (*houdt*), but its argument is realized as a prepositional object (*van het meisje*). In order to derive these sentences as translations of each other it is required to get the combination of *houdt* and *van* mapped somehow on to the English verb *loves*.

This translation relation is dealt with in the following way in Rosetta. There are rules to realize arguments of verbs syntactically. Such rules realize the two arguments of the verb *love* as a subject and a direct object, the single argument of the verb *dance* as a subject, the two arguments of the verb *think* as a subject and a finite subordinate clause (cf. *he thinks that he is ill*), etc. Each individual verb specifies how its arguments must be realized and the rules mentioned create the syntactic structure required by this specification. The Dutch verb *houden* (the dictionary entry with the meaning *love*) is specified in the following way: the verb takes two arguments. The first argument must be realized as a subject, the second argument must be realized as a prepositional object with the preposition *van*. If we ensure that the rule which realizes the arguments of *houden* and the rule which realizes the arguments of *love* correspond to each other, the examples mentioned can be translated compositionally.

Many cases in which it seems as if compositional translation is not possible can be dealt with in this way. We will mention some.

- The sentences treated in the preceding section (*er passeerde een auto—a car passed*) constitute another example of a non-trivial translation where the power of the rules makes it possible to maintain composi-

tional translation. Under certain conditions indefinite noun phrases cannot appear as subjects on their own in Dutch. They must be accompanied by the word *er*. In English no such restrictions hold (except for existential sentences). We can maintain compositional translation if we map the rule of English that introduces indefinite NPs as a subject into a sentence on to a corresponding rule in Dutch, which is formulated in such a way that it introduces indefinite NPs as a subject and the word *er* if required.

- The Dutch verb *scheren* requires the presence of a reflexive pronoun (*zich*) in the meaning that corresponds to the English intransitive *shave*: *hij scheerde zich—he shaved*. The exact form of the reflexive pronoun is dependent on syntactic properties (person, number) of the subject. This is accounted for in the following way. The verb *scheren* is specified as an inherently reflexive verb. There is a transformation in Dutch that syncategorematically introduces the appropriate reflexive pronoun in the syntactic structure if required by the verb. So, this translation can be achieved by simply mapping the verb *scheren* onto *shave*.

- 'Generic' plural NPs require the presence of a definite article in Spanish, though in English no article is allowed. Thus the English NP *flowers* must be translated into the Spanish NP *las flores* in sentence pairs such as *flowers are beautiful—las flores son hermosas*. For such cases there is, both in English and in Spanish, a rule that forms plural generic NPs. In English this rule takes a single noun as its argument and turns it into a generic NP by putting the noun in the plural. In Spanish the corresponding rule also takes a single noun as argument and turns this noun into a generic NP by putting the noun in the plural and introducing the appropriate plural definite article (*las*, if the noun is the feminine noun *flor*). In this way the translation can be compositional: the English noun *flower* is translated into the Spanish noun *flor*, and the English rule forming generic plural NPs is translated into the corresponding Spanish rule.

- The English verb *blow* must be combined with the particle *up* in sentences like *the soldiers blew the bridge up*. A Spanish translation of this sentence is *los soldados volaron el puente*. In this case the discontinuous unit *blew ... up* must be translated into the Spanish verb form *volaron*. This can be dealt with by specifying in the dictionary of English that the verb *blow* (when used in the meaning intended) requires the particle *up*, and by assuming that the rule that forms a sentence introduces the appropriate particle if required.

- The English sentence *the man will do it* must be translated into Spanish *el hombre lo hará*. In this case the combination of the auxiliary *will* and the main verb *do* must be translated into the single Spanish verb form *hará*. This can be accounted for by assuming rules in both languages to form future tense sentences. In Spanish this rule takes a sentence and turns it into a future tense sentence by putting the finite verb in this sentence in future tense. In English the corresponding rule puts the sentence in future tense by introducing the auxiliary *will*. More complex cases of sentence pairs such as *hij woont hier al een jaar* (lit.: *he lives here already a year*) and its correct translation *he has been living here for a year*, or *hij komt morgen* (lit.: *he comes tomorrow*) and its correct translation *he will come tomorrow* can be dealt with in a similar way. See Appelo (1986) for an extensive treatment of translational problems w.r.t. tense and aspect in the Rosetta framework.

- The English negative sentence *he does not see anyone* must be translated into Dutch *hij ziet niemand* (lit.: *he sees no one*). In this example the discontinuous *not ... anyone* must be mapped somehow onto the Dutch word *niemand*. This can be achieved in the following way. The word *anyone* is mapped onto Dutch *iemand*, English *not* is mapped onto Dutch *niet*. There is a negation rule in English that introduces the basic expression *not* into a sentence. If required, the auxiliary verb *do* is introduced as well. In this way we can derive *he does not see anyone* from *he sees anyone*. In Dutch there is a corresponding rule that introduces the basic expression *niet* into a sentence. However, if the word *niet* immediately precedes the word *iemand*, a transformation is applicable that deletes both *niet* and *iemand* and introduces the word *niemand*. Application of the negation rule to the sentence *hij ziet iemand* yields *hij ziet niet iemand*, which is transformed into *hij ziet niemand*. So, due to the power of the rules it is possible to maintain compositional translation if it is assumed that *iemand* corresponds to *anyone*, *niet* corresponds to *not*, and the Dutch rule introducing *niet* corresponds to the English rule introducing *not*. Of course, additional conditions must guarantee that *anyone* is only allowed in certain (e.g. negative) contexts and *someone* in other contexts. See Van Munster (1988) for a treatment of this problem and an extensive study of other translational problems concerning negation in the Rosetta framework.

Another class of translation problems is caused by 'categorial mismatches', i.e. cases in which a word of some syntactic category must be translated into a word of a different category. Examples of this class are; the Dutch *woonachtig (zijn)* (adjective) and its translation *reside* (verb); *(be) annoyed at* (adjective) and its Dutch translation *zich ergeren aan* (verb). For a description of the treatment of such cases, and of the somewhat more complicated cases such as *graag—like* and *toevallig —happen* (where an adverb must be translated into a verb), see Appelo, Fellinger, and Landsbergen (1987) and Odijk (1989).

4. Complex basic expressions

In this section we will discuss a number of translation problems that cannot be dealt with by merely having powerful syntactic rules, but that require in addition an extension of the notion of basic expression. In Subsection 4.1, we will discuss the treatment of idiomatic

expressions. In Subsection 4.2, we will show that the techniques developed for idioms also provide a solution to other translation problems.

4.1. *Idioms*

We can loosely define idioms as expressions consisting of more than one word, for which a literal, i.e. compositional, interpretation does not yield the correct meaning. The classic example is (1).

(1) kick the bucket

Literally this means *to hit a specific vessel with one's foot*. The idiomatic reading is approximately *to die*. It is obvious that this second interpretation cannot be derived compositionally from the parts of the expression.

Idioms occur in all languages. In most cases, an idiomatic expression in one language has an idiomatic translational equivalent in other languages, e.g. (1) corresponds to (2) in Dutch and (3a) to (3b).

(2) de pijp uit gaan
(to go out of the pipe, lit: the pipe out go)
(3) (a) spill the beans
(b) zijn mond voorbij praten
(to talk past one's mouth, lit: one's mouth past talk)

As these examples show, there is no direct relation between surface forms of idioms in different languages, so compositional translation would yield wrong results. In some cases, the most adequate translational equivalent may even be a single word, e.g. the best translation of the Dutch idioms (4a) and (5a) may be (4b) and (5b), respectively.

(4) (a) de pijp aan Maarten geven
(give the pipe to Maarten, lit: the pipe to Maarten give)
(b) opt out
(5) (a) laten zitten
(lit: let sit)
(b) ditch

Since the meaning and the translation of an idiom cannot be derived compositionally from its parts, we have to conclude that in a compositional framework an idiom must correspond to a basic expression, with a basic meaning. The problem we are confronted with then is how to represent such a basic expression, which corresponds to more than one word.

4.1.1. *Fixed idioms*

Some idioms can be treated as strings, i.e. as contiguous rows of words in a fixed order. These 'fixed idioms' are expressions consisting of more than one word in which the order of the words cannot be changed by syntactic operations and no words can intervene between the words of the fixed idiom. Furthermore, expressions of this type should be assignable to a lexical category, like noun or adjective. An example is (6).

(6) red herring

These idioms are treated in Rosetta as though they were simple words without any relevant internal structure. It is possible to apply morphological operations to them, e.g. for deriving the plural form *red herrings*. Note that there are types of fixed idioms for which it would be hard to assign an internal syntactic structure, even if one wanted to, because they are syntactically obsolete or in some other way deficient. (7) is an example of this in Dutch: a noun (*kant*) and an adjective (*klaar*) are coordinated. An English example of syntactic deficiency is (8), where a preposition and an adjective are coordinated.

(7) kant en klaar
(ready-made)
(8) by and large

4.1.2. *Flexible idioms*

For most idioms, e.g. those of (2), (3), and (4) it is impossible to treat them as strings. We will give two arguments against a string treatment here (for a more elaborate discussion cf. Schenk, 1986).

(i) The words of an idiom may be scattered over the sentence. For example, in (9a) *gave* and *the finger* have to be interpreted idiomatically, while the free argument *Mary* is intervening (the actual parts of the idiom are underlined). In (9b) a possessive pronoun that varies with the subject intervenes between the other parts of the idiom. In (9c) a temporal adverb is intervening between the verb and its complement.

(9) (a) Pete gave Mary the finger
(b) Pete lost his temper
(c) Pete gaf gisteren de pijp aan Maarten
(Pete gave yesterday the pipe to Maarten)

(ii) Idioms occur in a variety of forms that are accounted for in transformational grammar by means of syntactic transformations. See, for example, Fraser (1970). For example, the idiom in (10a) has a passive counterpart in (10b).

(10) (a) Pete broke Mary's heart
(b) Mary's heart was broken by Pete

Other examples of the transformational capacity of idioms are given in (11a-b). In (11a) the verb is in sentence-final position and in (11b) the verb is in the position following the subject.

(11) (a) Pete heeft de pijp aan Maarten gegeven
(Pete has the pipe to Maarten given)
(b) Pete gaf de pijp aan Maarten
(Pete gave the pipe to Maarten)

These examples show that a string treatment of these idioms, which we will call 'flexible idioms', would not account for all of the data. A representation in the form of an explicit syntactic structure is needed, which shows what the parts of the idiom are and where free arguments can be inserted. Furthermore, the fact that transformations apply to idioms suggests that the representation of idioms should be in a canonical form, i.e. a form to which no syntactic transformations have applied.

This strongly suggests that we should allow for complex basic expressions with an explicit constituent structure. However, a disadvantage of allowing arbitrary

constituent structures in the basic lexicon would be that this would not account for the fact that flexible idioms have regular constituent structures, which are similar to non-idiomatic constituent structures. It is desirable that the well-formedness of the syntactic structures that make up an idiom is guaranteed by the grammar. This has led us to the following solution.

An idiom is a basic expression with an explicit constituent structure. In the lexicon it is represented as a syntactic derivation tree that specifies how the canonical constituent structure of the idiom can be derived. This derivation tree contains 'normal' syntactic rules and 'normal' word stems. In this way the well-formedness of the idiomatic constituent structure can be guaranteed.

Obviously, this way of representing idioms is only one part of the solution. The other part is to organize the grammars in such a way that they can deal with these complex basic expressions. Therefore, we will go into the organization of the Rosetta grammars in the next subsection.

4.1.3. *Organization of the Rosetta grammars*

In this section we will give an outline of the actual Rosetta grammars as far as necessary to clarify the treatment of idioms. For a more extensive treatment of the organization of the Rosetta grammars, see Odijk (1989).

The Rosetta grammars are designed in such a way that arguments of a verb are introduced in a sentence in two steps. First a verb is combined with a number of syntactic variables (the number of variables equal to the number of arguments required by the verb), and later on so-called substitution rules substitute the actual arguments (NPs subordinate clauses, etc.) for these variables. This treatment is inspired by Montague grammar and makes it possible to deal adequately with scope phenomena (see Van Munster, 1988). However, it also plays a crucial role in the treatment of idioms, as we shall see. As pointed out in Section 2, the Rosetta grammars are reversible, which implies that they can be used both for analysis and for generation. In this section we discuss the grammars from the *analytical* point of view and show in particular how a basic expression corresponding to an idiom can be isolated during syntactic analysis.

We will illustrate the relevant aspects of the Rosetta grammars by showing a part of the analysis process of the English sentence *Did he kick the bucket?*, which has both a literal and an idiomatic reading.

The first rules that apply to a sentence in analysis are the so-called mood rules, i.e. rules that determine whether the sentence is interrogative or imperative or declarative, etc., and whether the sentence is a main or subordinate clause. For the sentence to be analysed it is determined that it is a main clause and a yes-no-question, and the structure is changed in such a way that the syntactic aspects expressing this (inverted order of the auxiliary and subject) disappear. The result of applying this rule is S(*he did kick the bucket*).

Next, the substitution rules must apply. In analysis, these rules 'desubstitute' elements from a sentence. They apply iteratively, i.e. they are applied zero or more times, the maximum being determined by the number of arguments in the sentence. A condition on the application of these rules guarantees that an argument is desubstituted only if no occurrences of variables to the right of it exist. In the sentence being dealt with here there are two potential arguments, i.e. *he* and *the bucket*. Now the following ways to proceed with the analysis process are possible:

no substitution rule is applied at all: S(*he did kick the bucket*)
only *the bucket* is desubstituted: S(*he did kick* x_1)
only *he* is desubstituted: S(x_1 *did kick the bucket*)
first *he* is desubstituted, and after this *the bucket* is desubstituted: S(x_1 *did kick* x_2)

Other options are not available, e.g. it is not possible to first desubstitute *the bucket* and after that *he*, because in that case a variable occurs to the right of *he* at the moment it is desubstituted. The NPs desubstituted from these structures (*he, the bucket*) are analysed themselves, and are found to be well-formed NPs of English.

The analysis process is continued for all four options. Tense and Aspect rules apply to determine the tense and aspect of the sentence and they undo the syntactic and morphological encoding of these properties (the past tense of the auxiliary verb *do* in the example being discussed). This yields the following four structures:

S(*he do kick the bucket*)
S(*he do kick* x_2)
S(x_1 *do kick the bucket*)
S(x_1 *do kick* x_2)

Next, rules apply that turn the structures given into a propositional unit with a verb as its head (called VPPs). These rules remove the auxiliary verb *do*. Voice is determined (the sentences are in active voice). This yields:

VPP (*he kick the bucket*)
VPP (*he kick* x_2)
VPP (x_1 *kick the bucket*)
VPP (x_1 *kick* x_2)

To these structures the so-called Pattern rules are applied. Pattern rules check whether the arguments of a verb are realized syntactically in the right way. Among the pattern rules there is a rule which states that a transitive verb (as the verb *kick* is) must realize its two arguments as a subject and a direct object. All four structures satisfy this requirement.

After application of the pattern rules the so-called Start rules are applied. These Start rules break the structure down into a basic expression and a number of syntactic variables. For the four structures mentioned the following candidate results can be formed:

candidate basic expression *he kick the bucket* + zero variables
candidate basic expression *he kick* + x_2
candidate basic expression *kick the bucket* + x_1
candidate basic expression *kick* + x_1 + x_2

These results will be found to be well-formed if the candidate basic expressions occur in the basic lexicon as actual basic expressions. Remember that we presented the rules in a simplified notation and that the candidate

basic expressions have in fact the form of constituent structures. The candidate basic expression in the fourth example (*kick*) is a single word and it can easily be recognized as an actual basic expression by looking whether it exists in the basic lexicon. It does, and this implies that an analysis has been found for the literal interpretation of the sentence. For the other three candidate basic expressions, it is checked whether there is a flexible idiom in the basic lexicon, i.e. whether there is a syntactic derivation tree that derives one of these expressions. The result of this process is that the candidate basic expression in the third example is recognized as a basic expression (the idiom *kick the bucket*), but the first and second example are not and so the corresponding analysis paths are rejected. Notice the role that variables play in this treatment. They indicate how many free arguments the basic expressions take. In the literal interpretation of the sentence two variables occur (x_1 and x_2) corresponding to the fact that the verb *kick* takes two free arguments. In the idiomatic interpretation one variable occurs (x_1), corresponding to the fact that the idiom *kick the bucket* takes one free argument.

4.2. *Translation idioms*

The techniques developed for dealing with flexible idioms as described in Subsections 4.1.2 and 4.1.3 can also be used to solve other translation problems, in particular if a word in one language does not correspond to a single word in the other language, but to a larger expression. This larger expression may have a compositional semantics, so it need not be an idiomatic expression from a monolingual point of view, but in spite of the compositional semantics a compositional translation is not possible. Because of the technical similarity with idioms, we will refer to these expressions as 'translation idioms'.

Examples are the Dutch and English expressions in (12a–b) and (13a–b) which have to be translated in simple verbs in Italian (12c) and Spanish (13c).

(12) (a) zachtjes neerleggen
(b) lay down with care
(c) adagiare

(13) (a) vroeg opstaan
(b) get up early
(c) madrugar

In (14), an example is given of a reflexive Spanish verb, corresponding to an idiomatic expression in English and a non-idiomatic expression in Dutch, that should be treated as a translation idiom.

(14) (a) verliefd worden
(in love become)
(b) fall in love
(c) enamorarse

These translations can be rendered by exactly the same techniques as introduced in Subsection 4.1.2 for monolingual idioms. The translation idioms are represented in the basic lexicon as syntactic derivation trees.

Translation idioms are not only useful for defining the translation relations between a word and a complex expression, but also between two complex expressions. Some examples of this are given in (15–16). In (15a–b) and (16a) a combination of a verb, an object, and a prepositional object has to be translated in a combination of a verb, an object, and a subordinate clause in which the verb takes an object (in (15c) and (16b–c)).

(15) (a) iemand om brood sturen (*Dutch*)
(b) send somebody for bread (*English*)
(c) mandar a alguien a buscar pan (*Spanish*)
(ask to someone to get bread)

(16) (a) iemand van het paard helpen (*Dutch*)
(someone from the horse help)
(b) help somebody get off the horse (*English*)
(c) ayudar a alguien a descender el caballo (*Spanish*)
(help to somebody to dismount the horse)

The examples mentioned above all involve expressions headed by a verb. The idiom techniques can be used for other constructions as well. We will give some examples without discussing them in any detail.

In (17) and (18) a combination of a noun phrase and a prepositional phrase in Dutch and English has to be translated into a combination of a noun phrase and a relative clause in Spanish.

(17) (a) de trein naar Gent
(b) the train to Gent
(c) el tren que va a Gent
(the train that goes to Gent)

(18) (a) de trein van Gent
(b) the train from Gent
(c) el tren que viene de Gent
(the train that comes from Gent)

In (19) the Dutch expression has to be translated into English and Spanish expressions containing an anaphor that has to be bound by an antecedent outside the expression.

(19) (a) voor eigen rekening
(on own account)
(b) at one's own expense
(c) por su cuenta

All these translation problems can be dealt with by allowing complex basic expressions represented by syntactic derivation trees in the basic lexicon.

5. Discussion and conclusions

In the previous sections we have shown that a number of translation problems can be solved systematically within the Rosetta framework. But, obviously, many problems are still waiting for a systematic solution, and in this final section we would like to discuss one of them. Consider the sentences given in (20) and (21). In these examples the combination of a verb and a directional prepositional phrase in English and Dutch has to be translated into the combination of a verb, an object NP, and a gerund in Spanish.

(20) (a) he swam across the river
(b) hij zwom de rivier over
(c) cruzó el río nadando
(crossed the river swimming)

(21) (a) he ran across the square
(b) hij rende het plein over
(c) cruzó la plaza corriendo
(crossed the square running)

Generally speaking, a movement verb followed by a directional prepositional phrase, headed by *across* in (a) and *over* in (b) respectively, and with a variable NP as object, has to be translated into a verb *cruzar* followed by a variable object NP and a gerund containing the movement verb. This translation problem is caused by the absence of a preposition with the meaning 'across' in Spanish.

It is possible to specify this translation relation by extending the dictionaries, using the flexible idiom method. However, if we would do this, every movement verb would have to be listed in combination with *across* in the dictionary separately, i.e. *swim across* translates into *cruzar nadando*, *run across* translates into *cruzar corriendo*, etc. This is a possible solution,[2] but unsatisfactory in our opinion, since this method does not capture the systematic character of these translation relations. An alternative, which does capture this generalization, would be to assume the existence of an abstract preposition in Spanish with the same meaning as the prepositions *across* and *over* and a rule that obligatorily turns structures containing this preposition into structures consisting of the verb *cruzar* and a gerund containing the movement verb. A disadvantage of this method, however, is that the grammar is now extended with a rule specifically written to deal with the problem of adequately translating the prepositions *across/over*, and it is clearly undesirable to write a rule solely for the treatment of a single word, because in that case each addition of a lexical item and its translations might require a change in the grammars. Nevertheless, the second approach probably is to be preferred, since the movement verbs form an open class, so that addition of a new movement verb would require addition of this movement verb plus *across* as well under the first alternative sketched. Under the second approach addition of the new movement verb itself suffices.

Summarizing, we have shown that many translation problems can be dealt with in this system of compositional translation. This is made possible on the one hand by the fact that rules can perform powerful operations, and on the other by the fact that basic expressions can have a complex structure. We also showed that certain translation relations cannot be dealt with elegantly by these methods. In cases like these we can either accept unelegant solutions (as sketched in this section) or try to adjust the compositional framework in such a way that it can deal with these problems adequately.

Acknowledgements

We would like to thank Lisette Appelo and Margreet Sanders for their comments on an earlier version of this paper.

Notes

1. Note that we use the term 'compositional translation' in a strict sense by requiring a direct relation between the grammars of the languages. A looser type of compositionality has been studied in the Eurotra project. Cf. Arnold *et al.* (1986).
2. In Isabelle *et al.* (1988) a similar solution is described in the context of a transfer system for English and French.

References

Appelo, L. (1986). 'A Compositional Approach to the Translation of Temporal Expressions in the Rosetta System', Philips Research M.S. 13.677, *Proceedings of the 11th Conference on Computational Linguistics*. Bonn.

—— Fellinger, C. and Landsbergen, J. (1987). 'Subgrammars, Rule Classes and Control in the Rosetta Translation System', Philips Research M.S. 14.131, *Proceedings of European ACL Conference*. Copenhagen.

Arnold, D. J., Krauwer, S., Rosner, M., Tombe, L. des, and Varile, G. B. (1986). 'The <C,A>,T Framework in Eurotra', *Proceedings of the 11th Conference on Computational Linguistics*, 297–303. Bonn.

Beaven, J. L. and Whitelock, P. (1988). 'Machine Translation Using Isomorphic UCGs', *Proceedings of the 12th Conference on Computational Linguistics*, 32–5. Budapest.

De Jong, F., and Appelo, L. (1987). 'Synonymy and Translation', Philips Research M.S. 14.269, *Proceedings of the 6th Amsterdam Colloquium*.

Dowty, D. R., Wall, R. E. and Peters, S. (1981). *Introduction to Montague Semantics*. Reidel: Dordrecht,

Fraser, B. (1970) 'Idioms within the Transformational Grammar', *Foundations of Language*, 6, 22–43.

Isabelle, P., Dymetman, M., and Macklovitch, E. (1986). 'CRITTER: A Translation System for Agricultural Market Reports', *Proceedings of the 12th Conference on Computational Linguistics*, 261–6. Budapest.

Landsbergen, L. (1981) 'Adaptation of Montague Grammar to the Requirements of Parsing', Philips Research Reprint 7573, in J. A. G. Groenendijk, T. M. V. Janssen, and M. B. J. Stokhof (eds.), *Formal Methods in the Study of Language*, part 2, 399–420. MC Tract 136, Mathematical Centre, Amsterdam.

—— (1987). '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, 1984, in M. King (ed.), *Machine Translation Today*. Edinburgh University Press.

—— (1987). 'Montague Grammar and Machine Translation', Philips Research M.S. 14.026, in P. Whitelock, *et al.* (eds.), *Linguistic Theory and Computer Applications*. Academic Press: London.

Odijk, J. (1989). 'The Organisation of the Rosetta Grammars', *Proceedings of the European ACL Conference*. Manchester.

Partee, B. H. (1982). 'Compositionality', in F. Landman, and F. Veltman (eds.), *Varieties of Formal Semantics*, (*Proceedings of the 4th Amsterdam Colloquium*), 281–312. Foris, Dordrecht.

Schenk, A. (1986). 'Idioms in the Rosetta Machine Translation System', Philips Research M.S. 13.508, *Proceedings of the 11th Conference on Computational Linguistics*. Bonn.

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

Van Munster, E. (1988). 'The Treatment of Scope and Negation', Philips Research M.S. 14.718, *Proceedings of the 12th Conference on Computational Linguistics*. Budapest.