THE ROSETTA PROJECT

Jan LANDSBERGEN

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

1. Origin

The Rosetta project has its roots in an earlier research project at Philips Research Laboratories, called PHLIQA (cf. Bronnenberg et al. (1)). In this project a system was developed that answered questions posed in natural English about information stored in a data base. The first component of that system had the task to convert a question into an expression of a logical language. This was done by a parser based on an attributed context-free grammar with a translation rule into the logical language coupled to each context-free rule, thus enabling a compositional translation into logic. However, because not all aspects of language could be dealt with adequately in this context-free framework, a second, transformational, component was added. First the context-free component translated into a rather hybrid representation, partially a logical and partially a deep syntactic structure. Then the second component would turn this hybrid representation into a genuine logical expression by applying transformation rules. This organization of the grammar was considered unsatisfactory, especially because of the unclear status of the hybrid intermediate representation and the transformational component. It was decided to design a new grammar which would be fully compositional, but of which the rules could be syntactically more powerful than context-free. We regarded the grammars described by Montague, e.g. the one usually referred to as PTQ (cf. Thomason (2)) as attractive examples of such an approach.

A Montague grammar specifies (i) a set of 'basic expressions', expressions with a primitive meaning, and (ii) a set of compositional rules (with well-defined meanings, formulated by means of translation rules into intensional logic), which prescribe how larger expressions and ultimately sentences can be built from these basic expressions. In PTQ, a number of the rules have concatenation as their syntactic operation and therefore can be called context-free (or categorial), but there are also rules that perform stronger syntactic operations, e.g. substitution of a noun phrase for a syntactic variable or syncategorematic introduction of an article. (Montague Grammar is often associated with pure categorial grammar, where concatenation is the only syntactic operation, but this is incorrect.) The process of deriving a sentence from basic expressions by application of compositional rules can be represented by a *derivation tree,* in which the terminal nodes are labelled by basic expressions and the non-terminal nodes by names of rules. Because of the compositionality of the grammar, the derivation tree determines both the form and the meaning of a sentence. Thanks to the power of the rules, a derivation tree of a sentence may differ substantially from its surface structure.

In order to make Montague Grammar suited for natural language processing, several problems had to be solved. The most important problem was that Montague Grammar is a completely generative formalism. Special measures are necessary to make effective analysis possible. Friedman and Warren (3) designed a parser which takes the context-free character of the majority of the rules as a starting point, but with special measures for dealing with the exceptions. I opted for a more general approach (Landsbergen (4)) and defined a version of Montague Grammar where all rules may have 'transformational power', as long as the grammar obeys a few general conditions, one of them being reversibility of the rules. In this version, called M-grammar, the rules do not operate on strings as in PTQ, but on constituent structures. Such an extension had already been proposed by Partee (5). The parser for M-grammar was implemented and tested for the PTQ grammar and several of its variants, but it was not incorporated in the question-answering system, because the PHLIQA project was ended.

In the course of 1980, I became interested in applying Montague Grammar to machine translation. There are several ways to do this. The most obvious way is to use intensional logic as an intermediate language. This approach has the advantage that it provides translation with a firm semantic base, but there are also a number of disadvantages, discussed extensively in Landsbergen (6). One of them is that after translation into a logical language many aspects of the form, which may carry translation-relevant information, are lost. An interesting alternative is the use of derivation trees as intermediate representations, for they reflect both the syntactic and the semantic structure of a sentence. It was decided to investigate this possibility and a new research activity was started, to which the name Rosetta was assigned.

2. The isomorphic grammar approach

In original Montague Grammar basic expressions have exactly one meaning. For practical purposes it is better to allow them to have more than one meaning. This leads to a distinction between syntactic and semantic derivation trees. The nodes of a syntactic derivation tree are labelled by names of syntactic rules and basic expressions, the corresponding semantic derivation trees have the same geometry, but their nodes are labelled by names of meaning rules and basic meanings.

Given these definitions, the translation relation can be defined in the following compositional way: two sentences are considered translations of each other if they have the same semantic derivation tree. So they have the same meaning, but they also have corresponding syntactic derivation trees and therefore they are similar in the derivation of their form.

In order to develop a translation system according to this definition, the grammars of the source and the target language should not be written completely independently. They have to be *attuned* to each other in such a way that for each basic expression in one grammar, there is at least one basic expression in the other grammar with the same meaning and - similarly - for each rule in one grammar there is at least one rule in the other grammar. Grammars that are attuned to each other in this way are called *isomorphic grammars.* If isomorphic grammars are written for a source language SL and a target language TL, the translation can proceed as follows. First the syntactic analysis component of SL yields a set of syntactic derivation trees of the input sentence, these can be directly converted into semantic derivation trees, which in their turn are converted into syntactic derivation trees of TL. The rules in these trees are applied and the result is a set of translations in TL (cf. Landsbergen (7) for a more detailed description).

This compositional definition of translation is more powerful than it may appear at first sight, for the following reasons.
(i) The rules can perform complicated operations on constituent trees, as we have already seen. One important property of rules is their capacity of syncategorematically introducing elements into a structure, i.e. they can introduce elements that are not arguments of the rule.
(ii) Basic expressions need not correspond to a single word, but can correspond to more complex phrases. In this way idioms can be dealt with, but the same techniques can 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.
(iii) Because the grammars are subdivided into subgrammars (e.g. subgrammars for the formation of a clause, a preposition phrase and an adjective phrase) which are mutually isomorphic, it is possible to deal in a systematic way with 'categorial mismatches'. These are cases in which a word of some syntactic category must be translated into a word of a different category. Examples are the Dutch *woonachtig (zijn)* (adjective) and its English translation *reside* (verb) and the somewhat more complicated case *graag* (Dutch adverb) vs. *like* (verb), discussed in Odijk (8).

3. The current project

For several years Rosetta was a small research activity (about two persons). During that period two small experimental systems were implemented: Rosettal and Rosetta2. The situation changed in 1985, when it was decided to start a larger project with about ten researchers.[1] The project was planned to have two phases: in the first phase we would concentrate on the fundamental linguistic aspects of translation and also develop the necessary tools, the second phase would be more application-oriented. The first phase has been finished now. The most important results are:

1. The Rosetta formalism. This was developed further in order to make it suitable for the development of large grammars (cf. Appelo, Fellinger and Landsbergen (9)). The most important differences with the original formalism are:

   - the grammars have more internal structure, they are subdivided into subgrammars and rule classes,

   - there is explicit control on the application of rules in a subgrammar,

   - a distinction is made between meaningful rules (which have to be involved in the isomorphy relation) and syntactic transformations (which can be freely chosen for each language),

   - formal notations were developed for the various types of morphological and syntactic rules that are used in the system.

2. The Rosetta environment. This includes implemented algorithms for analysis and generation, compilers for the before-mentioned rule notations and other software tools, e.g. a software management system and a number of test tools.

[1] The project was partially sponsored by Nehem (Nederlandse Herstructureringsmaatschappij) in the period 1985 - 1988.

3. The Rosetta3 phrase translator. This is an experimental translation system, which is able to translate short sentences and phrases containing a large variety of natural language constructions, for the language pairs Dutch - English and Dutch - Spanish. Rosetta3 yields all *possible* translations of a phrase, i.e. the translations that are possible on the basis of linguistic knowledge only.

   In addition, a Dutch generation component has been produced and the major parts of the English and Spanish analysis components. It is planned to complete English - Dutch and English - Spanish phrase translators in 1989.

   Rosetta3 is a research result, not intended for practical use. Its grammars were developed on the basis of a list of linguistic constructions that was compiled in the beginning of the project. No corpus of actual texts was used in this phase.

   In this period several translation problems have been studied in some detail, e.g. the translation of temporal expressions, cf. Appelo (10), the treatment of idiomatic expressions, cf. Schenk (11) and the treatment of scope and negation, cf. Van Munster (12).

   The dictionaries of Rosetta3 are still small. In the course of 1989 they will grow to a size of about 5000 frequent words.

In the second phase of the project we will first make a version of Rosetta3 that is more efficient and more robust than the current one. Then we will start the development of Rosetta4, which will be a prototype system for a real application.

4. Is Rosetta an interlingual system?

With respect to the long-lasting controversy between interlingual and transfer approaches, Rosetta has a special position. I would like to conclude this paper by making this position clear.

1. From the point of view of the system's architecture Rosetta is clearly an interlingual system. It consists of an analysis component that translates from the source language into an intermediate language, of which the expressions are semantic representations, and a generation component that translates from this intermediate language into the target language.

2. On the other hand, the intermediate language of Rosetta is not a universal interlingua, but is defined for a specific set of languages. So Rosetta is not interlingual in this strict sense.

3. In an ideal interlingual system the analysis and generation component for each language can be developed independent of the other languages. We will not discuss here to what extent this is desirable or possible, but it is clearly not the case in Rosetta, where the isomorphic grammar approach is followed. However, two aspects should be distinguished here.

   (i) The collection of languages under consideration (in our case Dutch, English and Spanish and in principle all Germanic and Romance languages) influences the design of the individual grammars. The grammars have to fit into a general scheme, which prescribes what kinds of meaningful rules there are and in what order they have to be applied. In Odijk (8) this general scheme is described. In practice, a lot of freedom is left for each individual grammar and this method is not felt as a severe constraint by the grammar writers.

(ii) In some cases there is a more direct influence of one grammar on another grammar, e.g. in the before-mentioned case *graag - like,* for which one special rule had to be added to the Dutch grammar, which would presumably not be needed for translation from Dutch to German.

In the lexicon the influence of one particular language on the grammar of another language occurs more frequently. The isomorphic approach enforces that in the intermediate language all meaning distinctions are made that are needed for any of the languages involved and that all grammars have basic expressions in all grammars corresponding to these meanings.

In 3 (ii) we are confronted with the well-known problem of interlingual systems that the translation from language A to language B is complicated by the existence of language C. If the goal is the development of a real multilingual system, where it is not known during analysis what the target language is, this seems to be the price that has to be paid. If, however, the ultimate goal is to construct a number of bi-lingual systems, these complications can be avoided by making a clear distinction between the general parts and the parts that are inserted because of translation from or into a particular language.

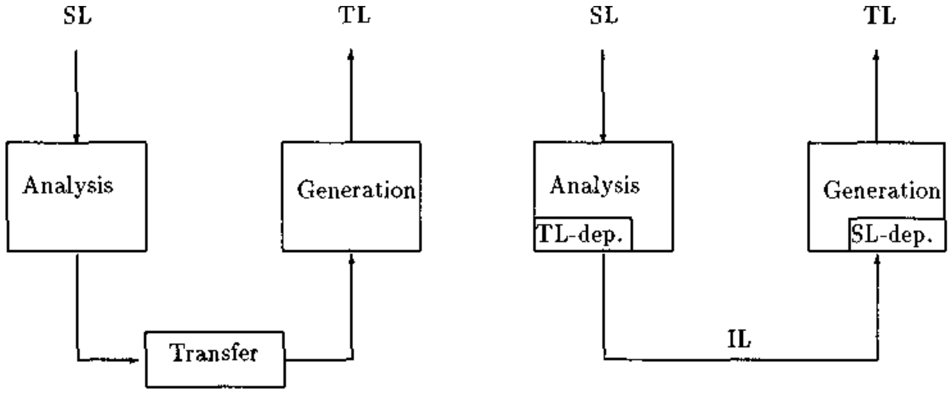

Figure 1: Transfer system and interlingual system with 'transfer parts'

Singling out the aspects that are specific for one language-pair is of course the philosophy behind transfer systems, but there it is coupled to a specific design, where the system consists of an analysis component, a transfer component and a generation component, in that order (cf. left part of figure 1). For an interlingual system like Rosetta a transfer version can be designed as follows (cf. right part of figure 1). The analysis component that translates from the source language into the intermediate language consists of two parts: a general part that is used for all languages under consideration and a part that is specific for each particular target language; the generation component consists of a general part and a part that is specific for the source language. The specific part may contain both rules and lexical entries, interleaved with general rules and lexical entries, so it is not a separate module that is applied before or after the general part.

This strategy of developing interlingual systems with 'transfer parts' may be pursued in future, more product-oriented, phases of the Rosetta project.

Acknowledgement

I would like to thank all those who contributed to the Rosetta project and especially the current team: Lisette Appelo, Franciska de Jong, René Leermakers, Elly van Munster, Jan Odijk, Joep Rous, Margreet Sanders, André Schenk and Harm Smit.

References

1 Bronnenberg, W.J.H.C., H.C. Bunt, S.P.J. Landsbergen, R.J.H. Scha, W.J. Schoenmakers and E.P.C. van Utteren, *The question-answering system PHLIQA 1,* in: L. Bolc (ed.), Natural Language Question-answering systems, Carl Hanser Verlag, München, 1980.

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

3 Friedman, J. and D.S. Warren, *A parsing method for Montague grammars,* Linguistics and Philosophy, 2, 1978.

4 Landsbergen, J., *Adaptation of Montague grammar to the requirements of parsing,* in: Groenendijk, J.A.G., T.M.V. Janssen, and M.B.J. Stokhof, Formal methods in the Study of Language Part 2, MC Tract 136, Mathematical Centre, Amsterdam, 1981, pp 399-420.

5 Partee, B.H., *Some transformational extensions of Montague grammar,* in: Partee, B.H. (ed.), Montague Grammar, Academic Press, New York, 1976, pp 51 - 76.

6 Landsbergen, J., *Montague Grammar and Machine Translation,* in: Whitelock, P. et al. (eds.), Linguistic Theory and Computer Applications, Academic Press, London, 1987.

7 Landsbergen, J., *Isomorphic grammars and their use in the Rosetta translation system,* presented at the Tutorial on Machine Translation, Lugano, 1984. In: M. King (ed), Machine Translation the state of the art, Edinburgh University Press, 1987.

8 Odijk, J., *The organization of the Rosetta grammars,* Procs of European ACL Conference, Manchester, 1989.

9 Appelo, L., C. Fellinger and J. Landsbergen, *Subgrammars, Rule Classes and Control in the Rosetta Translation System,* Procs of European ACL Conference, Copenhagen, 1987.

10 Appelo, L., *A Compositional approach to the Translation of Temporal Expressions in the Rosetta System,* Procs of the 11th Conference on Computational Linguistics, 1986, Bonn.

11 Schenk, A., *Idioms in the Rosetta Machine Translation System,* Procs of the 11th Conference on Computational Linguistics, 1986, Bonn.

12 Van Munster, E., *The Treatment of Scope and Negation,* Procs of the 12th Conference on Computational Linguistics, 1988, Budapest.