[*Terminologie et Traduction*, no.3, 1985]

ELEMENTS OF EUROTRA TODAY

1. Introduction

The purpose of this article is to give an impression of present discussion and developments in the EUROTRA project.

Some initial acquaintance with the fundamental design of EUROTRA is assumed (comprehensive presentations are given in Machine Translation; A Series of Talks for DG IX by Prof. Margaret King,published in Terminologie Bulletin n° 40 and in EUROTRA and its objectives by M. King and S. Perschke, published in Multilingua 1-1 1982, Mouton Publishers). This means that concepts like "transfer based system", "analysis", "generation" and "interface structure" are used with no further explanation.

At the same time it should be noticed that this article is not a state-of-the-art report on EUROTRA, first of all because such a report could not be kept within the limits of an article in the present review, secondly because a genuine state-of-the-art report is being prepared under a contract with the ISSCO institute in Geneva, and thirdly because the present author is not capable of writing such a report.

2. Some elements of the EUROTRA design debate

The fundamental design of EUROTRA is based on multilinguality, modularity and extensibility.

Multilinguality means that analysis and generation of one language are independent of analysis and generation of all other languages. Bilingual relations are catered for in transfer. In translation from Greek to Italian, for example, analysis of the Greek source text proceeds monolingually (independently of the target language); in transfer the interface structure which results from the Greek analysis is transferred into an Italian interface structure, primarily by lexical transfer (i.e., substituting one lexical element for another); finally, generation transforms the Italian interface structure into an Italian target text.

Modularity means that different systems tasks are performed by different modules working, as far as possible, independently of one another. Greek analysis is performed by one module, Greek to Italian transfer by another, and Italian generation by a third module. In a system of this kind any module may be substituted by an improved version without affecting the other modules in unforeseeable ways. Moreover, big modules may

consist of small modules and thereby allow for the same process of improvement by substitution, at the same time as it makes the construction of the system easier, simply because it is easier to construct smaller elements and make sure that they work well than to do the same with bigger elements.

Extensibility is thus a result of the multilingual and modular design. It means that new languages, new linguistic theories on local phenomena or on single languages, new text types and new subject fields may be added in a way which does not invalidate or destroy those parts of the system that have already been made.

Consequently, the first EUROTRA prototype will be a small system capable only of treating a "sublanguage", characterized as non-fictional prose concerning subject fields like informatics, telecommunication, computer science and legal and policy matters. If the fundamental aims of multilinguality and modularity are achieved, this prototype will be extensible, and so it will be the basis, hopefully, of a long series of enhancements and improvements.

2.1 The software

The principles of modularity and extensibility hold for the software design of EUROTRA as well as for the linguistic design.

Traditional software design is based on the so-called top-down approach. This means that the process of writing a programme or a system of programmes is conceived as a translation of a formal description of the problem(s) to be treated into a series of statements written in a programming language (i.e. a language which is understood by the computer).

In the case of EUROTRA, however, the formal description of the problems to be treated is part of the project. The existing theories in theoretical and computational linguistics do not seem to be immediately formalizable in such a way that they may be translated into a system of programmes, and they do not cover the whole range of linguistic phenomena which has to be treated in a machine translation system. Therefore, the development of software and linguistic specifications has been designed as two parallel and mutually interacting processes: an initial set of software specifications are used to design a user language, and an initial set of analysis, transfer and generation modules are created in this user language. The creation of linguistic modules then produces some insights which are used in a revision of the user language design, and so on.

Adopting a top-down approach to this process would imply a total revision of all programmes in each revision cycle and prolong the project through several decades. Instead, the EUROTRA software designers have adopted the prototyping principle. They have defined a virtual machine (an abstract definition of a machine which may be implemented in various ways using various types of software on various types of hardware), which will support a series of different user languages. Through the use of existing software tools, which can be bought off the shelf, this approach allows for rapid prototyping, where the design of new or revised user languages is made by feeding user language specifications into the computer. On the basis of these specifications, the computer then automatically creates the user language by compiling a compiler, which will cater for the syntax, and by generating a code generator, which will cater for the semantics of this user language.

In a traditional system of programmes, different programming languages may be used for different programmes, but the system as such is stable in the sense that the languages and the basic programming techniques are chosen before the system development begins. This means that the tasks performed by means of a certain language and the relations between the languages are fixed.

The EUROTRA software design may include any programming language, as long as this language is compatible with the definition of the virtual machine, and the division of labour between the languages as well as the relations between them may be fixed and refixed according to the needs of the user language designers. This creates a considerable degree of freedom in the provision of software tools for development and implementation.

2.2 Linguistic theories

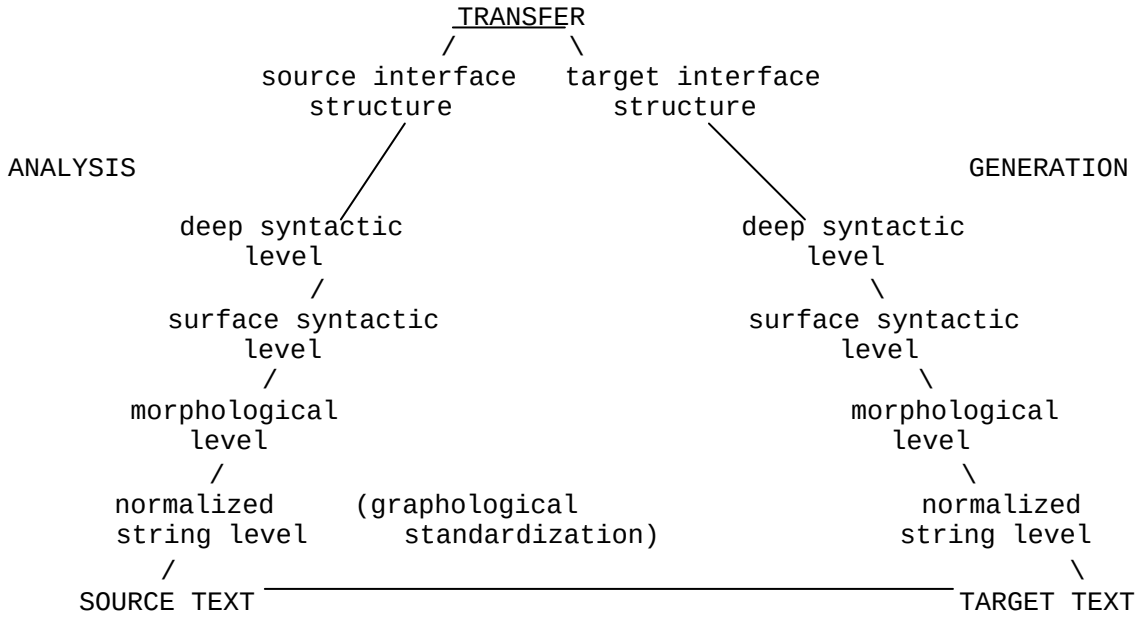
It was stated in the previous section that existing theories in theoretical and computational linguistics are not immediately formalizable in such a way that they may be translated into a system of programmes. This does not mean, however, that one theory is as good as another in the context of machine translation. Some theories have quite obviously been conceived in a much more formalizable way than others, and this is the reason why the linguistic thinking tools used in EUROTRA have in almost every case been taken out of the Northern American linguistic theories that were created in the wake of Chomsky's Transformational-Generative Grammar.

The rewrite rules of generative grammars, the semantic features of Fodor and Katz, the deep cases of Fillmore, etc. are easily recognizable in the existing draft specifications. On the basis

of these theoretical elements the EUROTRA linguistic specification designers are trying to create a complete and coherent set of specifications for the analysis and generation modules to be constructed for each of the Community languages.

If transfer has to be reduced to a minimum (cf. King, Term. Bull. 40), the analysis and generation modules will have to be very powerful, which means that there will be a long way to go between the source and target texts and the interface structure. In accordance with the modularity principle this way has been divided into a sequence of parts, each of which connects two linguistic levels.

The levels reflect the traditional subdivision of linguistic description into phonology/graphology, morphology, surface syntax, deep syntax and semantics. This gives the following conceptual picture of the EUROTRA translation process:



The diagram indicates that the "distance" between the source and target interface structures is smaller than the "distance" between the source and target texts, and it is exactly this reduction of the "distance" which justifies the enormous amount of work put into the construction of the sophisticated analysis and generation modules, because reducing the "distance" means simplifying the translation process per se, i.e. transfer.

The linguistic specification work one till now has shown that although the graphological, morphological and surface and deep syntactic levels are by no means simple, they are describable within the theoretical framework of generative grammars. The problems related to the semantic description of the interface structure are more serious, and at present only the semantic roles assigned in accordance with the principles of Case Grammar and, to a certain extent, the conversion of verbal tenses to interlingual time labels have been incorporated in a useful way. The use of semantic features and the description of modality, negation, focus, scope etc. are far from satisfactory.

The overwhelming problem, however, is the mapping of one level onto the next, e.g. how do we get from a surface to a deep syntactic description of a text? One way of attacking this problem might be the introduction of "transformation rules" saying things like:

> If the finite verb of a sentence is passive, the surface subject is transformed into a deep object;

but there are many syntactic phenomena which do not lend themselves easily to rules of this kind. One good example is ellipsis. The transformation of the sentences

> Kim went into the woods, and Sandy followed

from their surface syntactic form to a deep syntactic form must include the creation of a complete dependency structure (canonical form) for both of the juxtaposed sentences:

(Kim)	(go)	(into the woods)	(and)	(Sandy)	(follow)	(Kim)	(into...)
sub-ject	main verb	prepositional phrase		subject	main verb	object	PP

If however, we create a general "transformation rule", which covers cases of this kind:

> The second sentence of a pair of coordinated sentences related by <u>and</u> is completed with elements of the first sentence if the main verb of the second sentence has empty slots in its surface syntactic frame; (surface syntactic frame = frame describing the syntactic complements of a finite verb, e.g. subject, direct object, indirect object).

this rule will also apply to sentences like

> Kim ate the apple and Sandy left

although we have no way of knowing whether these sentences represent an elliptic construction.

To solve this problem we need semantic information of a very sophisticated kind, i.e. the probability of the main verb of the second sentence accepting any of the elements of the first sentence in its syntactic frame on semantic grounds.

Other Linguistic phenomena which create problems for the mapping are discontinuous constituents (e.g. complex German verbs: mitmachen, vorstellen), displaced dependents (e.g. negation in sentences like "I do not think Kim likes Sandy"), idioms ("to give somebody a hand") etc.

In order to overcome the problems of mapping and to get a more precise instrument of linguistic analysis and generation, the present discussion in EUROTRA concentrates on a new framework which seems to give better guarantees of computability than the existing user language, and which contains a much clearer mechanism for linking the linguistic levels.

3. New ideas

The present discussion is concentrating on two ideas: logical unification and a hierarchy of linguistic descriptors based on one single element: the feature.

Logical unification is a powerful and flexible mechanism which makes it possible to compare actual linguistic objects (texts, sentences, words) with abstract analytical objects contained in the knowledge base of the system by applying pattern matching, equality testing and feature passing. Pattern matching is the basis of the identification and processing of linguistic objects in EUROTRA.

The abstract analytical objects are hierarchic structures built of features, and features are attribute/value pairs.

This must seem pretty abstruse to any non-initiated person, and I shall try to be a bit more explicit about it.

Logical analysis of an object never works on the real object itself. It works on a "picture" of the object: the analytical object. The analysis of an object is concerned with the properties of this object, and it is quite normal in logic to define an analytical object as a collection (a bundle) of properties (or features).

Some features of some objects are not stable. This means that they may appear as different values under different conditions. In this case we distinguish between the "feature frame", which we call the attribute, and the actual value.

If we take our analytical object to be a word, we may define a feature with the name "part of speech". The frame will then be: part of speech, and the values: noun, verb, adjective, preposition etc.

In order to be able to handle enormous numbers of features, as required in linguistic analysis, we create bundles of features and call them atoms. In most cases atoms will be analytical objects corresponding to words, and the features will have names like "part of speech", "gender", "number", "person" and "case".

Atoms may also be grouped by analytical objects called constructors. A constructor could have the name "noun phrase" and group atoms with "part of speech" features which would have values like noun, adjective, pronoun and article.

By this grouping procedure classes of real objects (words, clauses, sentences, texts) are related to analytical objects and the linguistic levels are defined as sets of constructors with features of a certain kind. Surface syntactic constructors, for example, contain "part of speech" features, while deep syntactic constructors contain some "deep syntactic function" features.

Mapping one level onto another is a straightforward procedure (at least in theory), because it simply consists in translation of one (simple or complex) constructor into another. An example will illustrate this:

Let us define three atoms. One has the following features (among others):

A lexical feature (attribute: lex, value: the)
A number feature (attribute: number, value: Ø (the article the is not defined in relation to number))
A determinative feature (attribute: determination, value: determinate (as opposed to the value: indeterminate of a))

The second has the features:

A lexical feature (attribute: lex, value: window)
A number feature (attribute: number, value: singular)
A person feature (attribute: person, value: third)

The third has the features:

A lexical feature (attribute: lex, value: break)
A number feature (attribute: number, value: singular)
A person feature (attribute: person, value: third)
A tense feature (attribute: tense, value: past)

These atoms are matched against a string of characters: "the window broke", and the string and the atoms are unified.

Now, a noun phrase constructor groups the first and the second atom:

$C1_{np}$ = (the; {features}) (window, {features})

The third atom is accepted by a verb phrase constructor:

$C2_{vp}$ = (break, {features})

and the two constructors are grouped by a sentence constructor:

$C_{3\ s} = (C_1)\ (C_2)$

The sentence constructor will, in some way, be related to a rule which states that the first NP in English positive, declarative sentences is the subject, and this gives us the necessary information about the surface syntactic relations.

Going from surface to deep syntax, two atoms stay unchanged apart from some minor additions of deep syntactic relational information (from the deep level dictionary), one atom disappears, and the constructors are translated by direct relation:

(the, {features }) → Ø

($window, \{features_1\}$) → ($window, \{features_2\}$)

($break, \{ features_1$ }) → (break, {$features_2$})

$C_1 \rightarrow C_4$, $C_2 \rightarrow C_5$, $C_3 \rightarrow Ø$

This translation reflects the choice to delete the sentence constructor from deep syntax and to merge the article and the noun by adding the feature "determination" to the noun constructor C_4 (feature passing). Another choice could be the representation of deep subject as a constructor (C_4). The deep syntactic representation would then contain the following constructors:

$$C_4 = (\text{subject}, \left\{ \begin{array}{l} \text{lex: window} \\ \text{number: sing} \\ \text{person: third} \\ \text{determination: determinate} \end{array} \right\})$$

$$C_5 = (\text{gov}, \left\{ \begin{array}{l} \text{lex: break} \\ \text{number: sing} \\ \text{person: third} \\ \text{tense: past} \end{array} \right\})$$

The mapping of one level onto another by this procedure becomes a question of establishing direct relations between constructors. In order to be able to do this we may need an enormous number of constructors (in actual fact, constructors representing every imaginable type of sentence and clause), but at least it may be done in an orderly way. As against this, the mere fact of having a representation of some analytical levels based on generative devices does not imply anything about how to get from one level to another.

4. Conclusion

This presentation of some of the core elements of the discussion which goes on in EUROTRA today is very sketchy and incomplete. The examples are oversimplified and the notation was invented for the purpose of this article. The final notation that will eventually be established within the new EUROTRA framework may bear no resemblance whatsoever to the one which is used here.

Nevertheless, the article gives an outline of the principles which govern the present research and development in the project, and the first genuine EUROTRA prototype will be constructed on the basis of these principles.

Peter LAU
TAI/EUROTRA
Luxembourg

EUROTRA PROJECT MANAGEMENT AND COORDINATION

In accordance with the Council Decision on a Research and Development Program for an Advanced Machine Translation System the Eurotra Project is carried out mainly through Association Contracts with member state bodies under Commission supervision.

The actual work is done by research teams in the member states working independently on the construction of analysis and generation modules for the seven official Community languages, and it is easy to understand that this decentralized structure might lead to the creation of at least seven different or even incompatible modules (in some member states there is more than one research team, and some languages, e.g. French and Dutch, are treated by teams in more than one country).

This, of course, would be very unfortunate, especially in view of the fact that the analysis and generation modules are meant to interact through a common transfer module.

Moreover, a machine translation project has at least two aspects: a computational and a linguistic aspect, and while it may be possible to establish a computational research project on the basis of a relatively homogeneous scientific tradition, a linguistic research project will inevitably be haunted by serious differences in training and scientific background, any time you try to make linguists from more than one country and more than one scientific school collaborate.

Therefore, if you don't want to see your project ending up with seven or more incompatible "translation systems", you need a very strong project management and coordination.

The Council Decision states that the Commission is responsible for the Eurotra Program, and that it is assisted by an Advisory Committee for Program Management. In addition to this the budgetary means allocated to the program are meant to cover eight temporary agents, who should carry out the actual management and coordination work. However, due to various budgetary crises and unforeseen problems, these agents have never been hired, and during the entire preparatory phase (1983-84) all central work has been done by two Commission employees and a central team working under special contracts.

The main part of the central work during the preparatory phase, however, has not been management and coordination. In order to assure the compatibility and homogeneity of the modules developed by the language groups the work of the central team during this period has been concentrated on the elaboration of specifications for software, user language and linguistic analysis/generation.

Now the individual language groups are starting up work on their languages on the basis of these specifications, and management and coordination has become much more of a problem than before. At the same time there is no indication that the budgetary situation of the Community Institutions might ameliorate in the near future.

The central team, which was originally established in order to compensate for the lack of the temporary agents (the project team), certainly is a very small qualified team, but it is not suited to take over the day to day management and coordination tasks, simply because its members are living and working in their home countries. Managing and coordinating the central team, in fact, is a big task in itself.

Partly in order to contribute to the solution of this problem DG IX of the Commission last year offered DG XIII (directorate general responsible for the Eurotra Project) to second a group of four translators to the project. This secondment, of course, was only partly offered to solve the problems of DG XIII, as DG IX, and especially the translation service, has got problems enough itself (ever increasing amounts of texts to be translated, constantly evolving terminology etc. etc.). Another part (and no doubt the major part) of the motivation is to be found in an interest in having some people available with a certain knowledge of the system, so that the translation service may be well prepared if, some day, the Commission decides that Eurotra shall be used as a working tool by that service.

The four translators seconded to Eurotra formally belong to "Terminologie et applications informatiques" (TAI) and are, thus, still full members of the translation service. This arrangement could prove to be of great advantage provided that the seconded group is able to catch up with the project work, which is now proceeding through its eighth year, and provided the project leads to the construction of an operational system which could be used by the translation service of the Commission. This remains to be seen.

Peter LAU
Eurotra group - TAI - Luxembourg