14
Eurotra

This chapter describes what has probably been the biggest MT project yet, both in terms of number of personnel and amount of expenditure, and in the wide geographical distribution of the research groups involved. The Eurotra project has initiated and promoted basic research and development of the linguistic and computational requirements of complex multilingual MT system design, which will remain of lasting benefit to future developments.

14.1 Background

The increasing demands for translations in the European Community as a consequence of its multilingual policy had led the Commission to the introduction of Systran in 1976, initially for English–French translation and later for other pairs (Chapter 10). It was recognised, however, from the outset that inherent limitations in the Systran design reduced its potential as a multilingual system producing good quality output. In 1978 discussions began on a project for 'the creation of a machine translation system of advanced design (Eurotra) capable of dealing with all the official languages of the Community'. At that time these were Danish, Dutch, English, French, German, and Italian; Greek was added soon afterwards, and in 1986 Portuguese and Spanish were added. Eurotra is intended, therefore, to translate between nine languages, in all 72 language pairs.

The planning stage came to fruition with the formal approval of the Council of Ministers in November 1982 to set up a research and development programme in three phases. The first, 'preparatory', phase (1982–84) was to establish the

organisational framework, with research groups funded in each member state, and to define the basic linguistic and software specifications. The second phase from 1984 to 1988 was to be devoted to the linguistic research necessary for a small prototype system (with about 2,500 lexical items per language), to be tested on a particular corpus; and the third phase (1988 to 1990) was intended to enlarge the system to deal with more complex texts, working with about 20,000 lexical items in each language, and progressively less restricted to a particular text corpus. The domain of the prototype was limited to technical texts in the field of information technology. The final result was not to be a fully operational system but a 'pre-industrial prototype' as the basis for possible industrial development after the end of the research programme. As such, the emphasis was on quality of output —reasonable to good, without significant human involvement — and on extendibility rather than on speed of performance.

The initial Council decision was amended in November 1986 to take account of the accession of Spain and Portugal. The inclusion of Spanish and Portuguese inevitably meant a protraction of the development of a prototype and consequently, at the same time, the Council agreed to an extension of the second phase.

Progress, however, was slower than anticipated. The European Parliament set up an independent evaluation committee which reported in late 1987. The Pannenborg report was critical of failures to resolve problems of management and lack of central resources. It considered that an exclusive emphasis on linguistic foundations had led to the neglect of computational possibilities (e.g. interactive operation) and insufficient attention to dictionary compilation. It confirmed that a prototype was unlikely by 1990 and that no commercial system would appear by 1993. Nevertheless, the project was held to have made fundamental progress in the specification of interfaces and had succeeded in promoting computational linguistics research in member countries (which had been specified as a secondary aim of the programme). Its positive recommendations included more realistic deadlines, the ending of the research phase in 1988 and the establishment of closer links with industry for the development of a practical system, all of which contributed to the Council's approval in July 1988 of the transition to the third phase. By December 1989 there were 18 institutions involved in the project, located in 12 member states and comprising over 150 scientists, postgraduates and engineers (mostly active part-time only), with a central Commission staff in Luxembourg and under the general direction of Sergei Perschke.

A subsequent assessment in March 1990 (the Danzin report) confirmed the importance of the project as a stimulus to basic research in language technology, but also the unlikelihood of an operational MT system. It anticipated that the result of Eurotra would be a 'scientific definition prototype' rather than a 'pre-industrial development prototype'. It would have a limited lexicon, because not enough work has been done on terminology to achieve the planned 20,000 word coverage. However, there were already sufficient potential applications (monolingual as well as multilingual) of the Eurotra research to justify work already done, and the report recommended the continuation and broadening of the computational linguistics research framework established by the Eurotra project beyond 1991 to stimulate inter-European projects in basic and applied research and development of computer-based language tools.

Having begun as an ambitious R&D project to develop a multilingual MT system based on the latest advances in computational linguistics, Eurotra has at the time of writing entered a second round of funding, known as the 'transition programme', during which the emphases will be on laying the foundations for development of an industrial prototype by improving the performance, both linguistic and computational, and on encouraging more basic research on the theoretical foundations of MT in a multilingual framework. What is described in this chapter is not a prototype system but the specifications for a potential system which have been only partially implemented. Early articles about Eurotra concentrate on the political and administrative problems (which are in fact highly relevant to the system design), and give very general information about the nature of the translation problem to be solved. More technical articles began to appear around 1986, referring to a computational approach called the '<C,A>,T framework', later superseded, but nevertheless containing the seeds of what was finally to be adopted. Furthermore, the status of the various reports differs: some have been agreed by all participants, many exist as proposals. The basic design framework only became settled towards the end of the project, and there remain differences between researchers working in different environments. It is often difficult to establish from the writings of Eurotra-supported researchers whether they are giving an 'officially' approved view or not. A recently published set of articles can be regarded as definitive, and have informed the present description.

14.2 Organisation and system design

The requirements of decentralisation and multilinguality determined the development and design criteria of the system. While general administration and coordination was located in Luxembourg, all research was carried out by teams for each language in each Community country. Two groups had special functions not focused on a single language (Table 14.1), and other groups were involved in the project from time to time, notably ISSCO (*Istituto per gli Studi Semantici e Cognitivi*) in Geneva, and GETA in Grenoble.

Decentralisation on this scale demanded from the beginning a clear framework for the research and development of separate modules. Eurotra is the first attempt to create a genuinely multilingual system. A crucial decision, taken in 1978, was that Eurotra would be a transfer-based system, although perhaps, with the benefit of hindsight, as some observers have suggested, with originally 30, then 42, and finally 72 language pairs, it would have made more sense to choose an interlingua approach. However, it should be remembered that at that time, this approach had been so firmly discredited that the option was barely discussed (cf. section 1.3). The basic specifications for interface structures, for core formalisms and for software were drawn up by teams with members from throughout the Community; individual research groups worked on the analysis and generation modules for their own languages, and worked in pair-wise collaborations on the 72 transfer modules.

Definition of the interface structure (IS) was obviously crucial. In theory, since this is a transfer-based system, the IS for any one particular language could differ from one language pair to another, e.g. there could be one French

Task	*Group*	*Location*
Administration	DG XIII-B, CEC	Luxembourg
Danish	Københavns Universitet	Copenhagen
Dutch	Rijksuniversiteit Utrecht	Utrecht
	Katholieke Universiteit Leuven	Leuven
English	UMIST (University of Manchester Institute of Science and Technology)	Manchester
	University of Essex	Colchester
French	Université de Nancy II	Nancy
	Université de Paris VII	Paris
	Université de Liège	Liège
German	IAI (Institut für Angewandte Informationswissenschaft)	Saarbrücken
	IKP (Institut für Kommunikations-forschung und Phonetik)	Bonn
Greek	Eurotra Greece	Athens
	Panepistemio tou Rethymnou	Crete
Italian	Gruppo Dima	Turin
	Università di Pisa	Pisa
Portuguese	Universidade de Lisboa	Lisbon
Spanish	Universidad de Barcelona	Barcelona
	Universidad Autónoma de Madrid	Madrid
Terminology	Dublin City University	Dublin
Documentation and software clearing-house	CRETA (Centre de Recherches et d'Etudes en Traduction Automatique)	Luxembourg

Table 14.1 Participants in Eurotra

IS when translating into German and another French IS when translating into Spanish. However, this would have been not only less practical, obviously, but it would also have undermined the aim of having monolingual analysis and generation components which are neutral with respect to target and source languages, respectively. For this reason, there was agreed a common set of principles defining in general what the ISs should look like, and what level of abstraction of linguistic information they should contain.

In Eurotra translation is regarded as a **mapping** from a source text to a target text, where the mapping is defined in steps corresponding to **modules** of the system. In the three basic stages, 'analysis' is the mapping from source text to a representation R_s reflecting the source text, 'transfer' is the mapping of R_s onto a representation R_t reflecting the target text, and 'generation' is the mapping of R_t onto a target text (Figure 14.1).

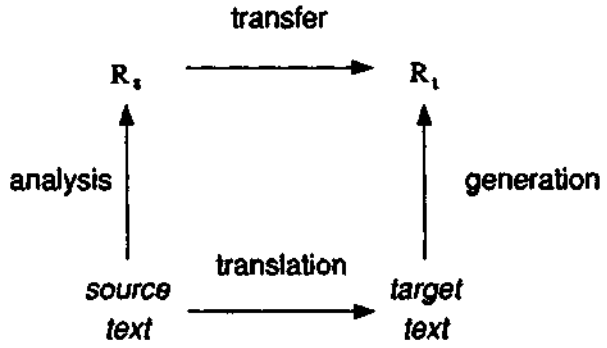


Figure 14.1 Notion of 'mapping'

With 72 language pairs there is an obvious requirement that transfer is kept as simple as possible; but as a consequence, the analysis and generation modules are correspondingly more complex than in other transfer-based systems. Because of this complexity, analysis and generation are also broken down into a series of mappings (Figure 14.2), giving the Eurotra framework its basically stratificational nature. The intermediate levels of representation are linguistically motivated, and up to five types of representation are envisaged (although some groups were able to define the sequence of mappings from text to IS in fewer stages).

The representations at the top of the figure, i.e. the final representation in analysis and the initial representation in generation, are the interface structures (ISs) mentioned above. It should be noticed that there is no stipulation that there must be the same number of levels of representation on both sides. The levels typically reflect the familiar distinctions of morphological, surface syntax (constituent), relational, and deep syntax descriptions of linguistic phenomena. They are characterised as follows:

- ETS (Eurotra Text Structure): the input text as received with formatting and publishing codes, including non-textual data and diagrams.
- ENT (Eurotra Normalised Text): the input text, stripped of all non-textual data and coding and roughly equivalent to an ASCII file.
- EMS (Eurotra Morphological Structure): a representation of words and morphemes in the form of a sequence of labelled word trees and punctuation marks.
- ECS (Eurotra Constituent Structure): a representation of surface syntactic constituency structure based on relations of syntactic categories.

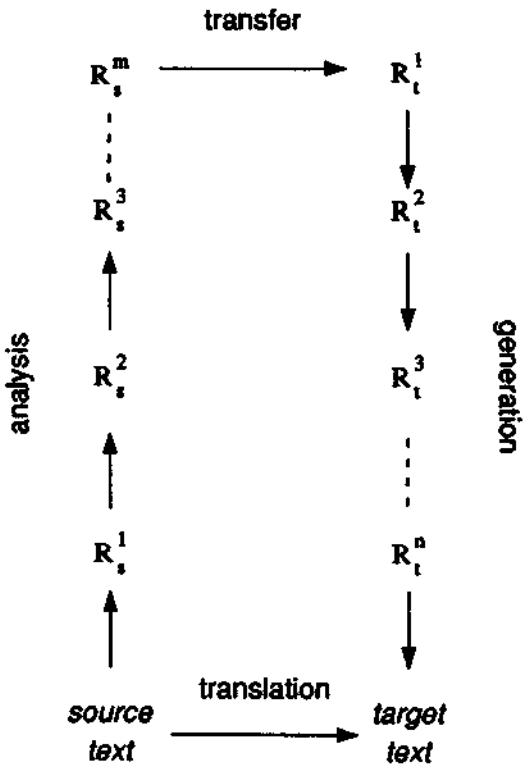


Figure 14.2 Complex mapping

- ERS (Eurotra Relational Structure): a representation of surface grammatical relations (subject, object, etc.) together with syntactic categories, broadly similar to LFG f-structures.
- IS (Interface Structure): a representation based on semantic dependency incorporating case frame structures and semantic features (animate, human, etc.) IS representations are the input to and output from transfer components.

14.3 Computational approach

In the terminology of the Eurotra project, the mapping between two levels of representation is called **translation**, an unfortunate choice of term, which should obviously not be confused with the ordinary sense of text-to-text translation (which for Eurotra is seen as a special case of 'translation' as defined here).

The Eurotra rule-writing formalism, called the **E-framework** , grew out of the earlier well-publicised '<C,A>,T' framework, and has much in common with it. The E-framework falls within the class of **unification-based** grammar formalisms (section 2.9.7), though, unlike many other such frameworks, it has not been designed with a view to dealing primarily with surface syntactic analysis. Rather, the E-framework is a more general mechanism for handling structured feature-based

linguistic objects, and fits neatly the overall stratificational linguistic approach of Eurotra.

14.3.1 Objects and structures

At each level the representations consist of primitive objects and structures built from these primitive objects. The primitive objects are **feature bundles** (cf. section 2.8.3), where features are attribute–value pairs and where the possible values of features and the possible combinations of attributes are defined by a 'feature theory' for each level of representation. For example, the feature theory for a morpho-syntactic level of representation might stipulate that if there is an object with an attribute–value pair where the attribute is 'category' and the value is 'noun', then the same feature bundle should have also an attribute 'case' with a value such as 'nominative' or 'accusative', a 'gender' attribute with a value 'masculine' or 'feminine', and so on. For each level of representation there is a defined set of features.

The legal primitive objects or categories at each level of representation are also defined by the feature theory: in practice this means that each level of representation is described by a set of 'feature declarations' which say what the features are (the attributes and their possible values), and the 'co-occurrence restrictions' on the features, which state which feature values are mutually compatible.

Turning now to the definition of **structures**, these are defined by rules which express the two structural properties of **dominance** (i.e. mother–daughter relationships) and **precedence** (i.e. ordering among sisters), cf. section 2.8.2. For example, at the syntactic level there may be a rule stating that an object having the features in (1a) might also have a structure made up of two objects, one a feature bundle including the features in (1b), and the other with the features in (1c). This kind of information is captured by the familiar type of rewrite rule.

(1a) {category=np, gender=masculine, case=nominative, number=singular}

(1b) {category=determiner, gender=masculine, case=nominative, number=singular}

(1c) {category=noun, gender=masculine, case=nominative, number=singular}

The 'rules' defining legal structures are collected to form the grammar of each descriptive level, where 'grammar' is understood in the formal sense of a set of context free rewrite rules defining the possible structural relationships of feature bundles.

It is important to note the 'declarative' nature of the feature theory and structural definitions. What the system does with these definitions is quite separate, and will now be discussed.

14.3.2 Translators and generators

The objects that are of interest to the MT system (at a computational level) are called **consolidated** objects, by which is meant structures which are well-formed according to all these rules. There are also **unconsolidated** objects, the provenance

of which will be revealed below, and which are structures which may or may not be well-formed: they are 'hypothetical structures', waiting to have their validity confirmed or denied. They may be structures which are only partially specified, or with inconsistent or incomplete feature specifications, or in need of restructuring.

The computational device which attempts to consolidate unconsolidated objects is called a 'generator'. This is again an unfortunate choice of terminology, since what a generator does here is not the 'generation' of target text. In fact, the term is used as the equivalent of a generative grammar in the mathematical sense (see section 2.9.1); hence it is declarative, and not a procedural device.

The unconsolidated objects are the results of 'translation', the process of mapping between levels of representation. Translation is done by 'translators', which take as input a single consolidated object (the output of the previous step) and process it top-down to produce as output a series of unconsolidated sub-objects which are passed to the next generator. In other words, translators take the output from a previous generator and change it in some way to provide input to the next generator; or more simply, they transform the structure and the features from one level to the next.

The generator performs the job of consolidation in a number of ways. In general terms, it can alter the dominance relations of a structure, primarily by inserting or removing nodes (though not by inverting them); and it can affect precedence either by imposing order on an unordered set of sisters or, again, by inserting or deleting sisters (but not by reversing their order).

There are three main types of rules found in a generator: structure-building rules, feature rules, and filter rules. Each of them has two variants.

Structure-building rules are either 'b-rules' or 'l-rules'. **B-rules** ('b' for 'building') are the most common type of rule, and most closely correspond to the conventional context free rewrite rule. A b-rule identifies a mother node and a list (possibly empty) of daughters, all specified as feature bundles. For example, (2) describes a simple noun phrase (perhaps for French) at the ECS level: `np` is the name of the rule (for debugging and control purposes); the '*' is a Kleene star and indicates iteration; the '^' indicates optionality. Informally, we can describe (2) as a rule defining a noun phrase with an optional determiner phrase, any number of adjective phrases (`ap`) where the adjective is prenominal (`adj_type=ante`), an obligatory noun, and any number of post-nominal adjectives; all the elements must agree in gender (`G`) and number (`N`), the capital letters indicating variables.

```
(2) np = {cat=np, gend=G, nb=N, def=DF}
         [^{cat=detp, gend=G, nb=N, def=DF},
          *{cat=ap, gend=G, nb=N, adj_type=ante},
          {cat=n, gend=G, nb=N},
          *{cat=ap, gend=G, nb=N, adj_type=post}]
```

A b-rule can be applied in four different modes, depending on the circumstances in which it comes into play. In one mode, corresponding to traditional bottom-up parsing, the daughters specified by the rule unify with a sequence of nodes, the mother node is built and the dominance relations are established. In a second mode, the rule provides confirmation of a structure; in this case both the mother and the daughters unify. In a third mode, where the

mother unifies with a node which has no daughters, the rule serves to create a new structure. And in the fourth mode, if the set of daughters is partially matched, then the rule might serve to insert an additional sister node.

L-rules ('l' for 'leaf') are also structure-building rules, but are restricted to rules which describe single feature bundles. They typically describe the atomic dictionary entries for the level concerned.

Feature rules apply only to objects in which dominance relations have been consolidated. They serve to refine such objects: by adding information (in the form of new features), in the case of **i-rules** ('i' for 'insertion'); or by causing features to propagate through the structure, in the case of **f-rules** ('f' for 'feature'). For example, the f-rule in (3) shows values for tense and aspect being passed up from the main verb (`role=gov, cat=v`) in a sentence to the governing (`cat=s`) node.

```
(3) s_tense =
          {s_tense=T,s_aspect=A,cat=s}
           [ {s_tense=T,s_aspect=A,role=gov,cat=v},
            *{}
           ]
```

The i-rule in (4) shows the insertion (identified by the '!') of a determiner in a (Dutch) noun phrase, with gender number and determiner type (`msdefs`) according to the marking on the mother node (`np`).

```
(4) np = {cat=np,msdefs=D,nb=N,nl_gender=G}
          [ ^{cat=quantp,pos=predet},
            !{cat=detp,msnb=N,nl_gender=G,msdefs=D},
            *{cat=ap},
            {cat=n},
            *{} ]
```

Filter rules, like feature rules, apply to objects where dominance has been consolidated, and they serve, as their name implies, to disallow structures which do not conform to certain specifications. In this way, they act as a safety valve by enabling the explicit suppression of structures which result from 'over-generation', i.e. the unforeseen superfluous results of rules, and by describing explicitly the conditions under which a structure can be accepted. Filter rules are, thus, of two types: **s-rules** ('s' for 'strict') specify acceptability conditions for structures, and **k-rules** ('k' for 'killer') explicitly suppress illegal or unwanted structures.

Some additional comments on filter rules are appropriate here. First, notice that their presence undermines slightly the declarative nature of generators in general. This is particularly true for the k-rules. Second, since s-rules can be regarded as restatements of the admissibility conditions for objects, it is difficult to know exactly how the description of legal objects should be divided between b-rules, i-rules, f-rules and s-rules. On the other hand, it might be seen as a strength of the formalism in that it gives the linguist a choice and the linguistic description is not confined by the tools available. However since s-rules are in effect a device to allow the grammar writer a 'second chance' in ensuring processes work as intended, they are almost as non-declarative as the k-rules. Nevertheless, if filter

rules are applied in a principled way (rather than as an *ad hoc* escape hatch), their availability should not undermine the general elegance of the E-framework.

Translators, as mentioned above, serve to map consolidated objects from one level onto sets of unconsolidated objects at the next level. Two basic principles have been developed to guide the definition of translators. The first, often referred to as the 'one-shot' principle, requires that between two representations the translator does not create any intermediate representations, i.e. elements(s) of one representation must be directly converted into element(s) of another. The second principle is that of 'compositionality', which states that interpretations of structures are functions of the interpretations of their components in a formally defined way, i.e. the translation of a complex expression should be a function of the translation of the basic expressions it contains together with their mode of combination — in this respect Eurotra shows the influence of the Rosetta project (Chapter 16) — though translators operate between representations and not (as in Rosetta) between derivation trees.

Each translator consists of a set of **t-rules** of two types. **Default t-rules**, as their name suggests, express automatic mappings from the feature theory at one level to that at the next. They are defined implicitly according the intersection of the feature theories of the two levels. **Explicit t-rules** define mappings which are not given by default, or contradict the default mappings. As such, they take priority over default t-rules when translators are activated.

Orthogonal to the explicit–default distinction is a distinction between **feature t-rules** (f-t-rules) and **structure-building t-rules** (b-t-rules), the former are used to map features, the latter to map structures. In general, f-t-rules are very simple mappings of one (or more) feature–value(s) at one level onto the next, as in (5), which maps a value for syntactic aspect onto semantic aspect (e.g. between the ECS level and the ERS level).

```
(5) t_simple = {syn_aspect=simple} =>
                {sem_aspect=perfective}
```

B-t-rules tend to be more complex. In example (6) a prepositional phrase (`cat=pp`) with the syntactic function of complement (`sf=compl`), formed with the preposition *of* at ECS, is mapped onto a simple noun phrase at ERS. The label `OBJ` is a variable name with no special significance; the '~' indicates explicitly that this node will be deleted; the '?' indicates that `cat` can have any value.

```
(6) tpp_compl =
            ~:{cat=pp,sf=compl}
               [ ~:{cat=p,gb_lu=of}, OBJ:{cat=?}]
        => OBJ:{sf=compl}.
```

As this description of the various rule types making up the E-framework demonstrates, the Eurotra linguists have available a rich computational formalism. But this richness means it is almost inevitable that different 'styles' of rule-writing emerge. For example, one group may write a large number of very simple and partly repetitive rules, while another group may try to capture similar linguistic facts with a small number of much more complex rules. Likewise, there are often alternative ways of doing the same thing, using different combinations of different rule types, and there is no firm guidance about which should be used.

14.3.3 Implementation

The importance of software tools adequate to the task of multilingual MT was recognised by the establishment of a strong team within Eurotra devoted to the development of efficient computational facilities. A major restriction was the need for software which could run on the fairly wide variety of machines favoured by the various research groups. In fact, the UNIX™ operating system was quickly agreed upon as a communal standard, and the E-framework software has been written in Prolog, in keeping with its generally declarative style and its unification basis. Needless to say, some implementations of the Eurotra software performed more efficiently than others. An example of the kind of problems encountered was the relative inefficiency of the Prolog compiler with such large programs. The Danzin report mentions an overall twenty-fold increase in performance between 1987 and 1990 thanks to an increase in computing power from 1Mips to 4Mips or 5Mips depending on the site, the introduction of the Yap™ compiler developed in Portugal to replace the standard Prolog interpreters (at least doubling and in some cases quadrupling efficiency), and a ten-fold improvement in the application software.

In the terminology of the Eurotra community itself, 'implementation' has come to mean not the installation of the basic software, but the development of the translation pairs themselves. At the time of writing, roughly the beginning of the 'transition programme' (see section 14.1), work on analysis and generation modules for all nine community languages is well under way, with transfer modules (of varying quality) for almost all 72 language pairs also under development. There is of course great variation also in the sizes of the linguistic modules amongst the different groups, not least because some began much later than others, and there is a similar variation in the sizes of the dictionaries, with lexical entries now numbering on average 4,000 in both monolingual and transfer dictionaries.

14.4 Linguistic aspects

Apart from an innovative computational approach to MT, the Eurotra project is remarkable for its significant contribution to the promotion of computational linguistics, and especially of contrastive linguistics, throughout the member states. At the beginning of the project, relatively little work had been done in a formal framework on any of the languages involved, with the obvious exceptions of English and to a lesser extent French and German. There are now large computational grammars of Greek, Danish, Dutch, Italian, Portuguese and Spanish. In the following section, we attempt to give some indication of the results of the important contrastive linguistic work undertaken in collaboration by the various Eurotra groups.

14.4.1 Research on linguistic topics

The breadth of coverage of linguistic phenomena in the project is indicated by the range of sentence types that most of the Eurotra implementations can handle: adverbial and subordinate clauses, verbless sentences, appositions, headlines and

parenthetical expressions, infinitives and control verbs, participle constructions, basic coordination, all temporal as well as modal forms, morphological analysis (inflectional and derivational), etc., with research continuing on problems such as ellipsis, negation, quantifier scope, and pronoun anaphora resolution.

Among the topics in which the Eurotra linguists can claim to have made significant advances, we may mention particularly tense and modality. These are two areas where cross-linguistic differences are notoriously treacherous. Even within the relatively small set of closely related languages that Eurotra deals with, there are great differences in the number of morpho-syntactic verb tenses and their corresponding uses (cf. the example sentence (7) in the next section). For example, the English present perfect form (*have* plus past participle) may indicate either an activity which has recently been completed or one which was begun in the past and continues. The two uses may result in different tenses in any of the other languages in the system. With so many different tense systems a pair-wise case-by-case treatment was simply impossible, and recognition of the problems triggered research into the specification of a feature system for indicating the temporal reference of any given morpho-syntactic tense and which would be common to the European languages involved, i.e. a 'euroversal' feature system for tense.

Modality (expressed in English by modal auxiliaries such as *can*, *must*, *may*, *might*, etc.) is equally complex, and, unlike time and tense, had not previously been studied extensively from a contrastive point of view. In certain respects, the problems were greater since the languages in the system do not even share the same morpho-syntactic means for expressing modality: auxiliary verbs, verb inflections, adverbials, or combinations of these. Again, a 'euroversal' system of modality features had to be worked out.

A general lexicographic problem for Eurotra was the need to develop a common perception of canonical form (cf. section 2.8.4), particularly in order that IS representations could be as interlingual as possible and that transfer could be restricted largely to lexical items. The need was to define the underlying head–modifier structures for various types of construction. It involved issues such as the role of prepositions (whether functioning as case markers or having lexical content — contrast *on* in *depend on the weather* and *sit on the floor*), the 'featurization' of modifiers such as determiners and quantifiers, and the treatment of compounds not as unanalysed units but as lexical items with structure.

In the general design of Eurotra it was important that the different language groups were assured of relative independence in matters of linguistic principle. Although all groups were constrained to use the E-framework, to aim at a common IS representation in analysis and to start from a given IS representation in generation, they were free to develop their own approaches to analysis and generation of their own language. The British group, for example, chose to skip the ERS level of representation, deriving IS structures directly from ECS representations. Likewise, there has been no centrally imposed common treatment of coordination, agreement, anaphora and so on. For these reasons, it is not possible to present a single 'Eurotra linguistic approach', since, apart from IS, there is none. We can, however, give some impression of the general stance of Eurotra in the next section, by illustrating briefly the analysis process of a sample sentence.

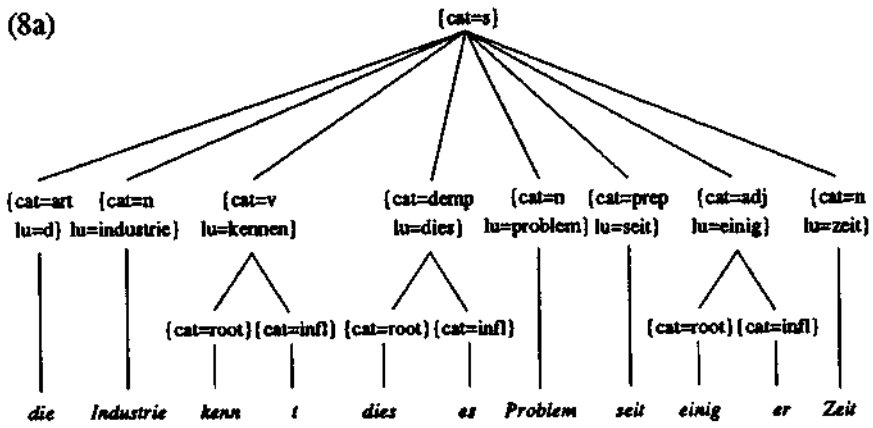
14.4.2 An illustrative example

We select for exemplification the German sentence (7) used in the most recent description of Eurotra. In the article from which this illustration is taken, the example is worked through to transfer and generation of Danish.

(7) *Die Industrie kennt dieses Problem seit einiger Zeit.*
'Industry has known about this problem for some time'

The representation after morphological analysis is a shallow structure, as in (8a). In fact the representation (as at other levels) is much richer than in the simplified tree diagram. This is shown in (8b): the feature bundles for the daughters give the lexical and morphological feature information associated with each word, where angle brackets indicate lists of daughters (the dots indicate parts of the representation omitted here for brevity, and are not part of the formalism).

(8a)

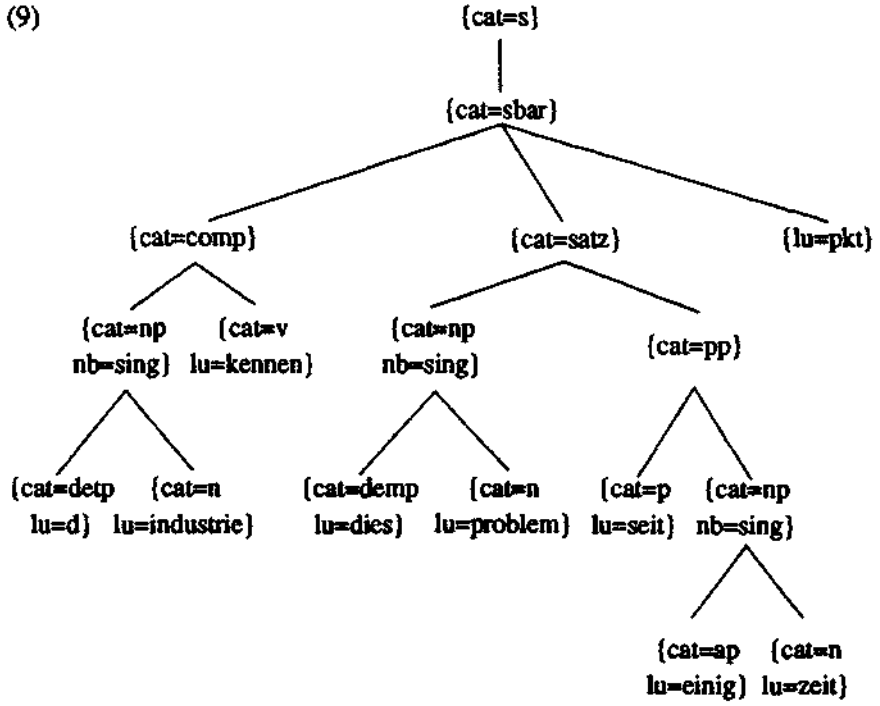


(8b)
```
{cat=s}
  <{cat=art, lu=d, msdefs=msdef, gender=fem,
    nb=sing},
   {cat=n, lu=industrie, gender=fem, nb=sing},
   {cat=v, lu=kennen, mstense=pres, nb=sing,
    pers=3}
    <{cat=root, lu=kenn},
     {cat=infl, lu=t}>,
   {cat=art, lu=dies, msdefs=msdef, gender=neut,
    nb=sing},
    <{cat=root, lu=dies},
     {cat=infl, lu=es}>,
   {cat=n, lu=problem, gender=neut, nb=sing},
   {cat=prep, lu=seit, ....},
   {cat=adj, lu=einig, ....},
    <{cat=root, lu=einig},
     {cat=infl, lu=er}>,
   {cat=n, lu=zeit, gender=fem, nb=sing}>
```

At this stage no intermediate constituents have been recognised, of course, though the treatment of inflectional morphology as a substructure should be noted. Ambiguities, such as the grammatical case and gender of *einiger*, are recognised in morphological analysis and are to be resolved at some subsequent stage.

The next stage is the translation of the EMS structure to a (set of) unconsolidated ECS structure(s), which the ECS generator then consolidates. Inflectional information is converted into features, and the lower nodes such as `root` and `infl` are discarded. The b-rules at this level are principally those of the 'bottom-up parsing' type (see above), taking sequences of nodes and creating a more structured representation, perhaps as in (9) (we show only the most relevant feature values, and for increased legibility show the structure in a more traditional tree form). Some readers might be surprised at the constituency structure of this analysis: the German ECS module is much influenced by Government and Binding theory (see section 2.10.3), hence the structure with the verb under `comp`. It is a good example of the way that representations at intermediate levels (i.e. other than IS) are not centrally legislated, but can be decided by individual language groups.

(9)



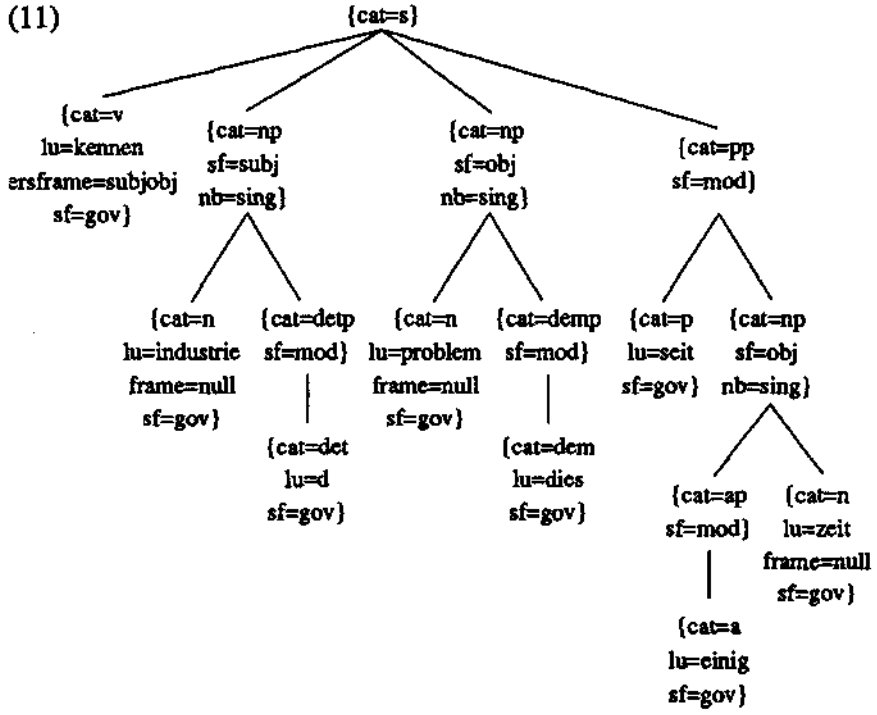
Consolidation at this level includes the 'percolation' of syntactic features from the leaves to the appropriate dominating nodes. For example, the case, gender and number of a noun phrase is given by unification of the possible values for these features on the daughters, typically by a b-rule such as (10).

```
(10) np = {cat=np, case=C, nb=N}
          [^{cat=detp, case=C, nb=N, gender=G},
           *{cat=ap, case=C, nb=N, gender=G},
           {cat=n, case=C, nb=N, gender=G}].
```

Recall that the '^' symbol indicates optionality, and the '*' repetition; capital letters indicate variables.

The representation at ERS level reflects dependency relations and syntactic functions such as governor, subject, modifier and so on; with these elements presented in a centrally legislated canonical order. The representation of our example sentence is roughly as in (11): again for clarity's sake, some feature values are not shown.

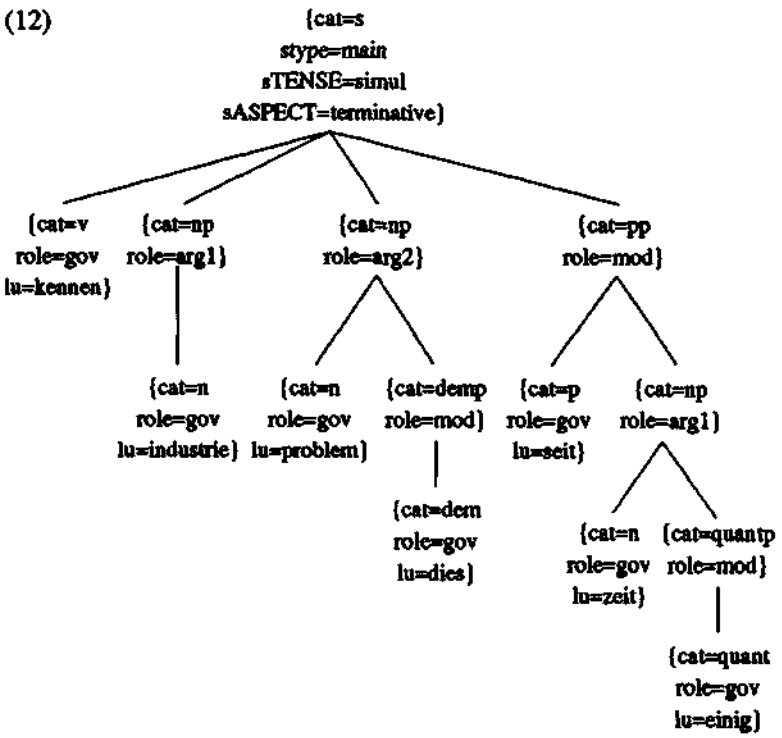
(11)



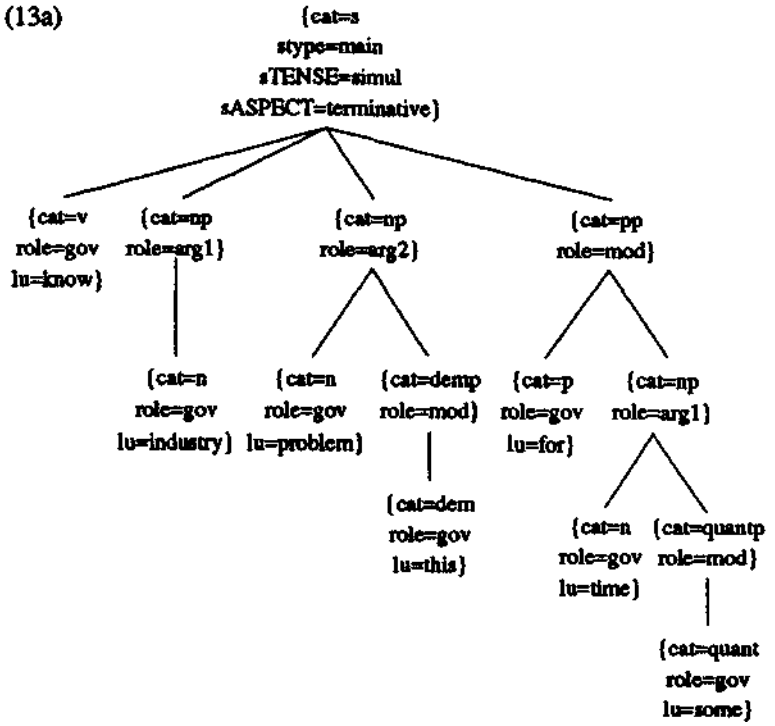
The IS structure is, in the case of this example sentence, very similar to the ERS structure, though the labellings reflect the deeper analysis (12). In fact, the full labelling of the IS tree is quite extensive: we show here, for example, that the `cat=s` node carries values for aspect, tense, sentence-type, and so on. Other labels are also present at all nodes (see below).

Transfer to an English IS consists just of replacing lexical items (13a). In our example sentence, the surface tense and aspect of the main verb will be changed from the German simple present to the present perfect in English. However, at the IS level, the time labelling for both languages is the same, the appropriate values having been calculated at a previous point in the German analysis (probably at ERS level) by a rule stipulating the values `sTENSE=simul` and

(12)


{cat=s
stype=main
sTENSE=simul
sASPECT=terminative}
{cat=v
role=gov
lu=kennen}
{cat=np
role=arg1}
{cat=np
role=arg2}
{cat=pp
role=mod}
{cat=n
role=gov
lu=industrie}
{cat=n
role=gov
lu=problem}
{cat=demp
role=mod}
{cat=p
role=gov
lu=seit}
{cat=np
role=arg1}
{cat=dem
role=gov
lu=dies}
{cat=n
role=gov
lu=zeit}
{cat=quantp
role=mod}
{cat=quant
role=gov
lu=einig}

(13a)


{cat=s
stype=main
sTENSE=simul
sASPECT=terminative}
{cat=v
role=gov
lu=know}
{cat=np
role=arg1}
{cat=np
role=arg2}
{cat=pp
role=mod}
{cat=n
role=gov
lu=industry}
{cat=n
role=gov
lu=problem}
{cat=demp
role=mod}
{cat=p
role=gov
lu=for}
{cat=np
role=arg1}
{cat=dem
role=gov
lu=this}
{cat=n
role=gov
lu=time}
{cat=quantp
role=mod}
{cat=quant
role=gov
lu=some}

'sASPECT=terminative' whenever the morpho-syntactic tense is present and there is a modifying prepositional phrase with *seit*.

In order to give a better indication of the complexity of the IS representation, we reproduce here more or less in full the English IS for this sentence (13b).

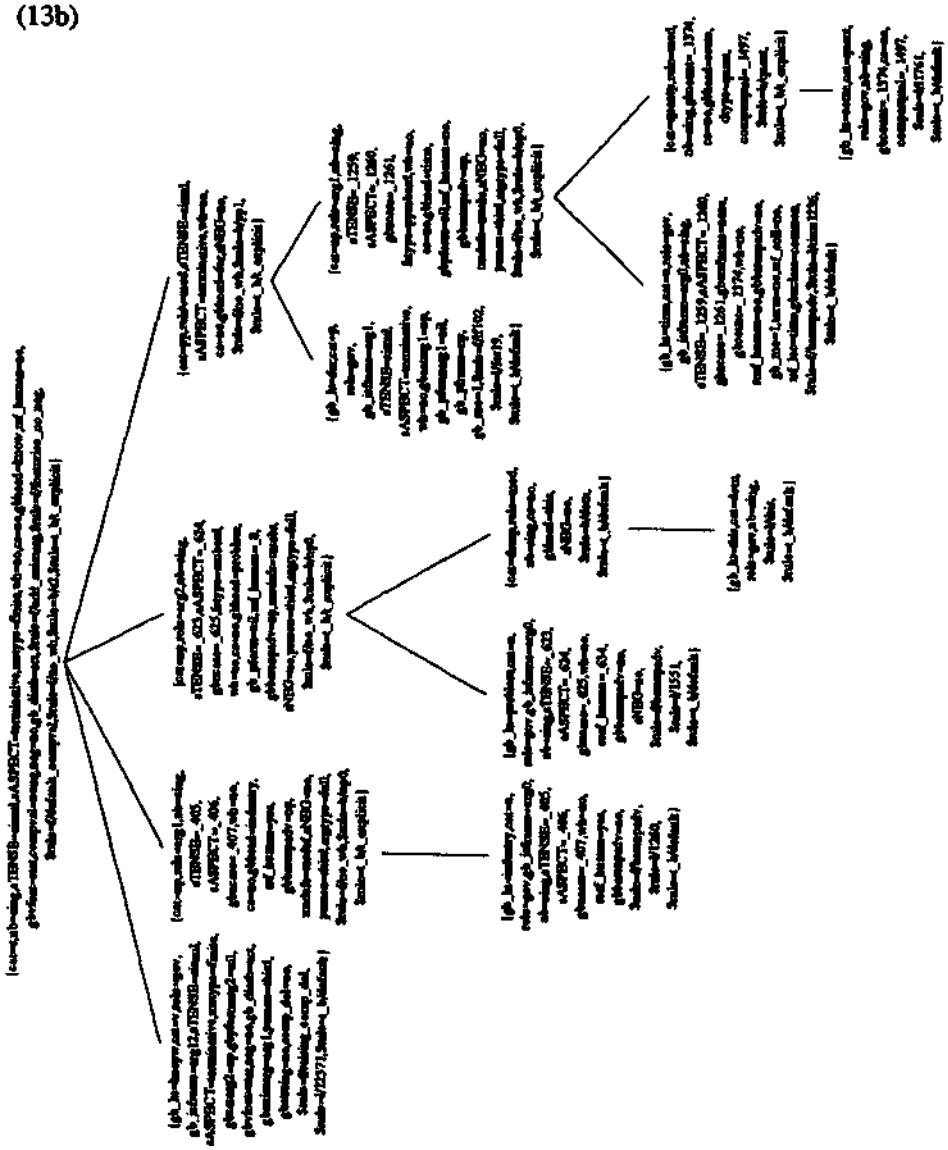
The reader's attention should be drawn to three interesting aspects of the IS representation. First is the inclusion in the feature bundles of traces of the rules which have been used to build the structures, seen as values of the feature $rule. Although clearly not part of the structure from a linguistic point of view, the inclusion of this information in the data structure is of obvious benefit to researchers working on the system. The second point concerns the features with unfilled values which show up in the IS as an underline character ('_') followed by a number. These are features which, perhaps by default, appear on each node at this level, but for which no value has been established, either by the translator, or by the unification process of the generator. From (13b) we can see that there must be a default rule that all nodes at IS level have the features sTENSE and sASPECT; but for all nodes except the mother node and its governor, the features are unvalued. The third point of interest is a related computational issue. It is the number of features with value no. These negative values are necessary because the Eurotra software is essentially unification-based: it was found that irrelevant features were being inserted into the structure by the unification process. It became necessary, therefore, at all levels (and most often at the 'lower' levels) to make explicit not only which feature–values apply, but also which ones do not. Researchers report this as a major disadvantage of the unification approach, perhaps unforeseen (or at least underestimated) by the developers of smaller experimental unification-based systems.

Obviously, the representation in (13b) is too unwieldy for the human linguist to manage comfortably, and in Eurotra, as in other systems such as Ariane and METAL (Chapters 13 and 15), some attention has been paid to making the huge data objects that the system manipulates more accessible for the researchers. For example, structures such as the ones shown in (13a) can easily be produced and investigated. What we have been able to show here is the size and complexity of the data structures created and handled by the Eurotra system, which in this respect can be said to be a typical large-scale second generation MT system.

14.5 Conclusions

Eurotra is undoubtedly the most ambitious MT project at the present time. As yet, the huge research effort has not resulted in even a 'scientific pre-development' system, but this must be seen in perspective. The basic linguistic knowledge required for a multilingual system did not exist when the project began. Linguistics research did not provide the necessary detail, and other MT systems did not organise data in ways which could be adopted for an advanced multilingual design. Now it can be said that a large part of this fundamental research has been accomplished. Much, of course, remains to be done. Eurotra has been above all a project for European linguists.

(13b)



As well as the mainstream Eurotra research, there has been some significant related 'spin-off' research, including two experimental MT systems (CAT and MiMo) based on and derived from Eurotra at an earlier stage in its development, and the development of a logic grammar formalism (CLG: Constraint Logic Grammars).

The flexibility and modularity of the basic Eurotra framework has itself stimulated linguistic research. Experimentation with different treatments of time and modality was facilitated by the independence of the interface and transfer components from the other modules. Complementary research has been sponsored elsewhere which explores topics of concern, e.g. discourse phenomena, within a somewhat looser, but still compatible, conceptual framework. The simplicity and perspicuity of the Eurotra basic design makes it an ideal 'communication medium' for linguists starting from different intellectual backgrounds.

There are limitations, of course. A major shortcoming is the too simplistic approach to semantics. In the Eurotra framework semantics is treated primarily by features attached to nodes in essentially syntax-based structures. This shallow semantics combined with deep and broad coverage of syntax has resulted in over-generation in analysis, and insufficient filtering by disambiguation. More constrained parsing and production might reduce over-generation, but possibly at the cost of damaging the stratificational design. There are alternatives to the stratificational design available with recent advances in cognitive science and in Artificial Intelligence, and it has always been accepted that Eurotra should be flexible enough to adopt new techniques, provided that they are well founded and adequately tested.

Some of this insufficiency of semantics could be minimised in a practical system by the incorporation of an interactive facility, and this has been recommended in reports to the Commission. The reports have also stressed that priority should be given to improvements in semantic representations and a greater urgency must be given to terminological and dictionary work.

Certain design questions remain unanswered. It would seem that in principle translators could be reversible, and that consequently whole modules could be used for both analysis and generation. Some Eurotra researchers are investigating the possibilities, but whether this will become part of the 'official' design is as yet unknown.

The theoretical activity of Eurotra researchers has contributed substantially to the theoretical foundations of MT. However, it remains an explicitly 'linguistic' system; it does not attempt to incorporate any knowledge bases or cognitive modelling of the AI kind. It has been criticised by some observers for its concentration on abstract formalism, for neglecting the construction of lexicons, for its insufficient empirical testing, and for what is seen as a 'narrow' exclusion of discourse phenomena and advances in knowledge-based approaches to natural language processing.

The disappointment of many external observers of Eurotra lies predominantly in the failure to produce practical results. The Danzin report in 1990 urged the Council to permit researchers to demonstrate specific applications, not to be constrained by the full multilingual requirements of the original proposals. There

could then be tangible products, monolingual in some cases, not just as spin-offs of Eurotra research but as agreed objectives: the recommendation recognises the substantial contribution to European computational linguistics of the project, and seeks to promote its potential commercial benefits.

Eurotra has successfully broadened the research base of European computational linguistics (in part by the training of manpower), it has heightened awareness amongst linguists of the need for inter-European collaboration and it has promoted awareness in governmental bodies of the growing economic and cultural importance of basic and applied research in natural language processing.

14.6 Sources and further reading

The development of Eurotra has been marked by various stages, reflected in the availability of published material on the project. Early references are very general in nature, reflecting more the political and administrative needs of the Commission, though some early reports do give an insight into the way these considerations affected the design (e.g. Johnson *et al.* 1985, Arnold 1986). The project then had perhaps three phases of scientific development, the first being now only of historical interest (and in any case only sketchily reported). The second, centring on the <C,A>,T formalism is relevant in that much of the E-framework is rooted in that formalism. Reports within the <C,A>,T framework include a special issue of *Multilingua* (Sager and Somers 1986), Arnold and des Tombe (1987), King and Perschke (1987), Raw *et al.* (1988) and Steiner *et al.* (1988).

The most up-to-date descriptions of Eurotra, from which much of the material in this chapter is adapted, are in the special issue of the journal *Machine Translation* (Allegranza *et al.* 1991). Other relevant reports include Bech and Nygaard (1988) and Varile and Lau (1988).

The Pannenborg and Danzin reports (CEC 1988, 1990) are available from the Commission of the European Communities in Luxembourg. The description in this chapter of the E-framework is specifically based on Bech *et al.* (1991), while the example of German analysis is adapted from Kirchmeier-Andersen (1991), with the assistance of members of the British Eurotra groups at UMIST and Essex. The CAT project is described in Sharp (1988), MiMo in Arnold and Sadler (1990) and van Noord *et al.* (1990), CLG in Damas and Varile (1989).

As already indicated, the writing of this chapter has benefited from the close association of one of the authors (HLS) with the project and its developers. However, it should be stressed that he left the Eurotra project in 1986 and up until the time of writing had no active involvement with it in any way. We would like to acknowledge the assistance in writing this chapter given to us by Paul Bennett, Jeanette Pugh, Nancy Underwood and Andy Way; we are of course responsible for any errors which remain.