Complex Lexical Transfer in METAL

Gr. Thurmair
Siemens AG
Otto-Hahn-Ring 6
8000 Munich 83
W. Germany
metal@ztivax.siemens.com

0 Introduction

This paper tries to describe how complex lexical transfer in METAL is done. In a first section it explains the basic outlines and changes in the transfer concept of METAL. In the following two sections, it presents examples which show what kinds of tests and actions have to be coped with by a MT system. Chapter four shows the consequences of these cases on the system architecture of METAL, and finally, some consequences for other phenomena (like multiwords) and METAL components are sketched.

1 Changes in the transfer concept of METAL

When METAL started to develop new language pairs, it turned out that some architectural changes had to be made. Some parts of the demonstrator in the early 80ies had been designed for a German-to-English system, and relied on the similarities of Germanic languages; this holds e.g. for the verb framing analysis (see Gebruers 88). But also transfer had to be reconsidered when more languages came into play.

Moreover, at that time METAL was more or less a rule-based translation system. Every analysis rule had its target counterpart. One of the basic weaknesses of this approach is that the generation part of such a system is closely linked to the analysis language; in order to be used for other analysis languages, it has to be made analysis independent, as different languages use different analysis rules. In this situation, we did not want to follow an approach of isomorphic grammars (cf. Landsbergen 89) as it was felt that its empirical basis was too small, but rather try to separate analysis and generation parts of the translation process.

As a first result, there are now special generation grammars for the different languages which are independent of the source language analysis rules. They describe certain tree structures and conditions of operation, and they are called during transfer. METAL does not use the same grammar for analysis and generation (like Kaplan et al. 88); this approach has never been demonstrated

successfully for large coverage grammars; and our experience is that the analysis coverage must be much bigger than generation coverage, e.g. wrt word ordering in German (cf. (1) to (4)):

(1) Dann hat der Mann der Frau das Buch gegeben
(2) Der Mann hat dann der Frau das Buch gegeben
(3) Der Mann hat der Frau dann das Buch gegeben
(4) Der Mann hat das Buch dann der Frau gegeben

All these sentences have to be analysed, but as long as no formal difference can be given between them (i.e. in terms of features), generation would just produce one of them randomly. It is a better approach to produce a default generation, and if readings can be differentiated, these defaults can be modified.

The next problem was the input of the generation grammar. If a generation component is to work with different analysis grammars, these grammars have to provide the same kind of input. This does not only hold for tree structures and categories but also for features. In a multilingual system, much more information has to be made explicit than in a just bilingual approach; for instance, in a Russian-to-German system, information about definiteness of NPs is essential for the creation of German determiners, whereas in an English-to-German system this information does not necessarily have to be stated explicitly, if just the differences can be identified. But as more languages are considered, more information must be explicit. As a result, the "METAL Interface Representation (MIR)" has been designed; it specifies the interface structures between the METAL languages.

In this concept, METAL uses a kind of "markerese" as interface, but there still are transfer lexica for special language pairs. These two information sources can interact, however, in a very complex way: The transfer of lexical items may cause complex changes in the whole syntactic structure.

This is different from structural transfer: Structural transfer is defined on certain tree properties; e.g. complex German prenominal modifiers are transformed into English relative clauses:

(5) DE: Das von ihm oft gelesene Buch
EN: the book which has been read by him frequently

This operation can be described independent of the lexical items in the tree. But there are other cases where lexical items cause severe structural changes, cf. (6), (7):

(6) DE: er liest oft / vielleicht
EN: he reads often / perhaps

(7) DE: er liest gerne
EN: he likes reading

While transfer is straightforward in (6), it is not in (7), although the sentences are syntactically very similar: Here, the sentential modifier of the source language (SL) becomes the main verb of the target language (TL), and the SL main verb becomes the TL direct object.

We can follow two strategies in such a situation: Either we try to find syntactic differences between (6) and (7) and try to cover (7) by means of structural transfer (i.e. make it independent of the actal lexical filler). This requires much more linguistic insight in many phenomena than is available at present. As long as we consider cases like (7) to be lexical idiosyncracies (and there are many of those cases!), we have to provide means for complex lexical transfer. This is not a structural transfer as the change is triggered by the lexicon, not by syntactic structure, but it has severe consequences on the sentence structure.

METAL follows both strategies at the same time: As long as phenomena are considered to be lexically based, we provide means to express this, and to do the necessary structural changes, based on an lexical entry in the transfer lexicon. Once there is more linguistic knowledge or more generality, this can be expressed in MIR, the system can do structural transfer, and the transfer lexicon entry can be made simpler.

The task of the transfer phase in METAL is therefore to deliver well-defined generation input structures from analysis structures, on structural level as well as on lexical level; this can include structural changes as well.

The following paper tries to show how complex lexical transfer could be organised and integrated into the whole system in a systematic way.

2 Conditions

An entry in the transfer lexicon in METAL consists of several parts, specifying the lexicon module an entry belongs to, special customer, dialect etc. variants, and a section with tests and actions.

In the simplest case, there is just a one-to-one transfer with no further specification:

(1) EN: placebo -> DE: Beruhigungspille

But in the case of one-to-many transfers, different readings have to be distinguished. The only information available for transfer selection is the SL analysis result. The question is, however, what is really needed, and where and how it can be accessed.

The easiest way is to access all information present at the node where the transfer is done; this is usually the terminal lexical node. All transfer information is locally present in this approach.

An example of such local tests is a test for the semantic type of a noun; it can be used for disambiguation in transfer:

```
(2)  DE: Bank    ->   EN: bank      (semtyp = social_institution)
                  ->   EN: bench     (semtyp = concrete_object)
```

If this information is to be used, the semantic readings of the respective noun has to be disambiguated during analysis. But the semantic type is local, i.e. it is available at the node where the transfer is performed, as a feature.

The same holds for some syntactic features, cf. (3)

```
(3)  DE:  Schuld  ->  EN:  guilt      (number = singular)
                  ->   EN: dept        (number = plural)

(4)  EN:  allow   ->   DE:  erlauben  (voice = active)
                  ->   DE:  duerfen   (voice = passive)
```

In (4) however, things begin to become complicated: To be passive is a matter of the sentence as a whole, not strictly of its verb, as there may be more factors involved to determine sentential voice / diathesis than just the predicate. In order to write the test in (4), the respective information has to be made acccessible to the node to be transferred. Lexical transfer is not a local operation any more, it is influenced by the whole configuration of the analysis.

This becomes completely obvious in configurations where syntactic structures are involved:

```
(5)  DE:  bestehen      ->   EN:  consist of   (prep_obj  = aus)
                        ->   EN:  pass         (dobj_lu   = "Examen")
                        ->   EN:  exist        (dobj  = nil)
```

These tests trigger different transfers, based on different syntactic contexts in which the verb occurs; they test certain properties of verb frames. Moreover, it is not just the verb frames, it is also their syntactic fillers which influence transfer. In (6), different prepositions of the prepositional object cause different transfers:

(6) DE: bestehen -> EN: consist of (prep_obj = aus)
-> EN: insist upon (prep_obj = auf)
-> EN: consist in (prep_obj = in)

Not only verbs, also other categories can have different transfers according to different configurations. In (7), the existence of a genitive attribute is responsible for a change in transfer, i.e. it has a disambiguating effect here; in (8), the noun has to be translated differently depending on its position as head or specifier in a compound construction:

(7) DE: Tief -> EN: low pressure area
Tief des Mannes -> EN: depression of the man

(8) DE: Tiger-Koenig -> EN: tiger king
Koenigs-Hof -> EN: royal court

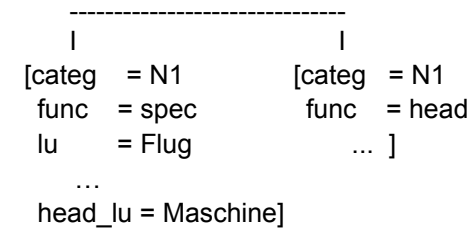
In order to state those transfer conditions, we need a structural description of a given configuration. As a result, tests in the transfer lexicon cannot be simple feature tests but must be complex descriptions of structural configurations.

This affects the concept of lexicon transfer as a whole. Lexical transfer cannot be done locally (i.e. at a given node of a tree with a given set of features/categories) if tests have to be applied which ask for structural information. In terms of X-bar categories, transfer of a noun cannot be done on N0-level if it depends on the existence of a modifier which is not available before the N1 level.

There are two ways to solve this problem: Either we copy down the relevant information to the local node (i.e. the lexical terminal node) before we call the transfer. E.g. in (9) we could tell the specifier node that the head lexical unit is "Maschine"; this could be used in transfer as test (10):

(9) [categ = N1
lu = Maschine
...]

|



(10) DE: Flug -> SP: avion (head_lu = "Maschine")

Transfer can be done locally in this case as the relevant information has been percolated down to the node in question. This transfer is called "short-sighted" transfer as it is done locally on the terminal node of the tree.

An alternative way would be to do transfer testing at the maximal projection level rather than at the X0 level. In this case, all relevant information is available at the top node, and structural relationships can be tested easily as long as they are available in the subtree which this maximal projection dominates. (11) would do transfer at N2-level, and an entry like (12) would try to match the tree in (11) using a tree comparison:

(11) DE: Hetze gegen Auslaender -> EN: agitation against foreigners
DE: Hetze -> EN: hurry

(12) DE: Hetze -> agitation (attr_pp with "gegen")
-> hurry

As this approach does transfer on the maximal projection of a category it is called "far-sighted" transfer (termini used in Gebruers 89). It would do noun transfer at NP level, verb transfer at S level, etc.

Far-sighted transfer can use all information of the tree it dominates (e.g. whether an NP has a genitive attribute, a PP attached to it, etc.); short-sighted transfer will be done locally; if it turns out that it needs information from outside, this information has to be percolated down to the respective local node.

This shows that the need for proper testing transfer alternatives causes a complex transfer strategy already (i.e. the question where the information to be tested is available).

The examples mentioned above show, by the way, that both functional and categorial information are needed for proper transfer testing. This is an obstacle for multi-layered systems where "deeper" level information might not

be available any more at transfer time.

In METAL, the transfer entry itself contains a structural pattern as test, describing a certain tree configuration (in terms of dominance and precedence relations) and a certain feature decoration for the tree nodes if possible. As METAL is a transformational system, this can be easily done in a formal way by just using the structural description part of a transformation.

3. Actions

One-to-many transfers have to disambiguate their readings in terms of SL tests. But performing these tests properly is just one of the problems we face. Very often, certain actions have to be performed in order to do good transfer. These actions again can be very complex; they influence the tree layout considerably.

Again, in simple cases, there is just replacement of one lexical unit by another one. But very often, specific actions have to be performed in order to create a well-formed TL expression. These actions do not just influence the TL lexical unit itself but also some other constituents in its neighbourhood.

The simpler cases consist in just changing feature information, e.g. voice:

(1)	DE: es besteht aus zwei Teilen	(voice = active)
	EN: it is composed of two parts	(voice = passive)
(2)	DE: es beruht auf etwas	(voice = active)
	EN: it is based on something	(voice = passive)

These features could simply be overwritten in the transfer phase, and generation would generate an English passive sentence. But (2) shows already that the verb requires specific transfer of the preposition in the prepositional object. More complex actions are needed.

The actions to be taken can be subclassified according to specific types of operations needed; the basic actions are:

- mapping lexical material into different material
- adding lexical material
- deleting lexical material
- other operations

3.1. Mapping operations

These cases can be explained by examples from verb transfer. Often, transfer

changes the syntactic functions of the verb arguments. Examples are:

(3) EN: He lacks something
DE: Ihm fehlt etwas

(4) EN: Something acts on it
DE: Etwas wirkt auf es ein

In (3), the English subject has to be mapped into the German indirect object, and the English direct object into the German subject. In (4), the verb subcategorises for special prepositions of the prepositional object. This idiosyncratic behaviour has to be stated in the transfer lexicon.

These cases show that verb transfer in fact is transfer of whole patterns, of functional structures, rather than just transfer of words (cf. Gebruers 88). It has to be specified what changes a verb transfer causes wrt the whole sentence structure.

This mapping of syntactic functions does not just hold for arguments for which the verb strictly subcategorises. Sometimes we have to transfer peripherals in a specific manner, cf.

(5) EN: He aims the gun at him
DE: Er zielt auf ihn mit dem Gewehr

(6) EN: vaporise into
DE: eindampfen in

Both (5) and (6) would not really be subcategorised for a direct object or a prepositional object: Subcategorisation is a matter of SL analysis (i.e. the analysis behaviour of the verb); we should not change monolingual facts with transfer problems in mind. Nevertheless, if such peripherals are realised, the verb controls their transfer. This phenomenon is even more frequent in the cases of NP transfer, cf.

(7) DE: Verbindung -> EN: connection
-> EN: session (ap_lu = "logisch")

(8) FR: machine -> DE: Waschmaschine (pp_lu = "a laver")

(In (8) , METAL avoids this coding by proper compound treatment). Therefore, mapping operations should be permitted for any constituent controlled by the head.

It should be noted that mapping is not restricted to the same syntactic level. In the case of factive readings, a prepositional verb complement turns into an attributive clause if it is realised sententially:

(9) DE: es besteht darin, [dass er kommt]
EN: it consists [in the [fact that he comes]]

The subordinate clause in (9) is verb argument in English but noun attribute in German.

3.2 Insertion operations

Sometimes, we need operations where new lexical material has to be inserted into a given syntactic structure. Examples are not just reflexive verbs like (10); but sometimes we have to create real syntactic functions like direct object ((11), (12)) or prepositional object (13), but also adverbials (14).
This problem does not hold just for verb transfer, cf. (15):

(10) EN: hurry
DE: sich beeilen

(11) EN: something emerges
DE: etwas bildet sich heraus

(12) DE: das Auto blendet ab
EN: the car dims the headlights

(13) EN: interconnect something
DE: etwas miteinander verbinden

(14) EN: electro-copperplate
DE: galvanisch verkupfern

(15) EN: session
DE: logische Verbindung

While (10) is not a transfer problem (the verb simply strictly subcategorises a reflexive), the other cases are. They are very frequent in the cases of transfer of German prefixed verbs. However, we do not want to talk about "inherent direct objects" etc., because we would have to change our monolingual semantic verb representation depending on new languages we consider. Therefore we have to foresee that transfer can add new lexical material together with new

syntactic functions.

The problem with insertions is, however, that in order to run the generation grammar successfully, the new structures have to be fully specified: Not only lexical units, but also definiteness, number, positional information, agreement behaviour etc. must be given by the insertion operation. In an operational system, a trade-off has to be found between information needed and the users' ability to code non-terminal linguistic information.

By the way, insertion problems show that a rule-to-rule approach in MT has its shortcomings: There would simply be no rule for the newly created transfers.

3.3 Deletion operations

This is the revert problem of insertion. Here again, we have to cope with incorporated arguments, prefixed verbs, etc., cf.

(16) EN: cut off smoothly
DE: glattschneiden

(17) DE: auf den Markt bringen
EN: launch

(18) DE: neu entwerfen
EN: redraw

(19) DE: die Luft abschnueren
EN: strangle

(26) DE: Gebrauch machen
EN: use

In all these cases, we have to delete SL lexical material in order to obtain proper transfer. Again, this material has quite different status: Sometimes we have peripherals (16), sometimes we have full arguments (19), sometimes we have "support verbs".

The problem with deletions is that we sometimes cannot simply delete the respective structure (NPs or AVPs etc.) as there may be modifiers around:

(21) DE: er bringt etwas auf den ueberfuellten Markt
(22) DE: er macht davon keinen Gebrauch

They would be deleted together with the whole constituent which leads to wrong transfer. Modifiers have to be treated differently here: Whereas in (21), the inserted adjective blocks the standard translation by adding a kind of figurative reading, could (22) simply be translated by raising the negation onto clause level. But this does not work in general, cf. (23):

(23) DE: er baut es fest ein -> EN: he incorporates it
DE: er baut es [nicht fest] ein -> EN: ?? he does not incorporate it

This area is still under investigation.

3.4 Other operations

There are still other cases which are influenced by the verb transfer, cf.:

(24) EN: [he came [to be president]]
DE: [er wurde Praesident]

(25) EN: [he runs the program]
DE: [er laesst [das Programm laufen]]

(26) EN: to default on a debt
DE: eine Schuld nicht bezahlen

(27) EN: not to default on a debt
DE: eine Schuld bezahlen

(28) EN: I do not care
DE: es ist mir egal

(29) EN: I care
DE: es ist mir nicht egal

(24) and (25) are cases where constituents have to be moved between main and subordinate clause. (26) to (29) are cases where the negation of the sentence is explicit or incorporated into the verb; it has to be reverted at the system surface.

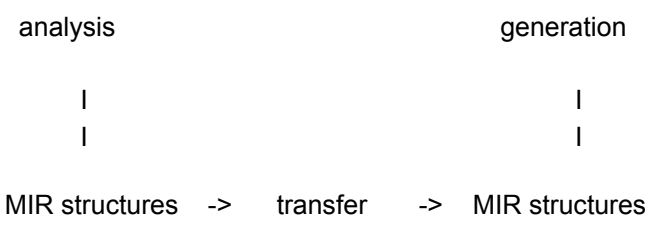
These phenomena again are triggered by the verb transfer and cause complex structural changes. They are not structural but lexical idiosyncrasies, and therefore have to be specified in the transfer lexicon of METAL.

4 Representation

The previous two sections showed that both for tests and for actions complex structural investigations and changes are needed. The question is, where, when, and how these changes should be performed, i.e. how the transfer lexicon can be integrated into the system, and how the structural tests and actions can be placed. This is again a question of system architecture.

4.1 Transfer and MIR

The basic architecture of METAL shows that transfer has well-formed MIR structures both as input and output:



This scheme shows that transfer does not do anything to the MIR if it is simple, one-to-one, and does not cause structural changes. If it is lexically complex then it is allowed to change MIR structures but it has to produce well-formed MIR structures again on which the generation components can operate. It also means that transfer should not do generation work itself (e.g. create real direct objects including their surface form); instead, it should just provide the information which is needed for the generator.

By the way, structural transfer has the same function and the same place in the system architecture. The only difference is that it is not driven by lexical transfer but by structural information; e.g. it has to transform a complex participial adjectival phrase into a relative clause when transferring into English, but this is not necessary when transferring into Dutch. Again, the result of structural transfer must be a well-defined MIR structure.

4.2 Far-sighted vs. short-sighted transfer

As the examples in chapter 3 above show, transfer is not just a local operation, replacing on lexical item by another one. Sometimes we have to change the whole structural environment, or we have to interpret it for testing cases. This can only be done if transfer is done at a higher structural level where the relevant information is really available.

This means that far-sighted transfer has to be done every time we want to apply

tests and actions to a lexical transfer, as these tests and actions refer to the structural environment. Examples show that in most of the cases, this is identical with transfer of nouns at NP level, of verbs at clause level etc., i.e. that the heads of the constituents control the transfer of their constituent.

In METAL, we do transfer of the controllers of a structure at their maximal projection level (far-sighted transfer), and transfer of the controllees locally at terminal level (short-sighted transfer). This enables us to perform all the tests and actions at the place they belong to: When transferring controllers, we should be able to use and modify the whole structure they control. Depending nodes are transferred locally; if they need external information, it can be percolated to these nodes.

This concept allows for a clear and simple architecture which fits well the X-bar scheme of grammar.

The relation between controllers and controllees can be specified in a declarative way, leaving control completely to the linguist. Such an approach has been described in Alonso 1988 and successfully implemented in the Spanish generation grammar of METAL.

4.3 Transformations

The remaining question is what kind of formal devices should be used to perform all these tests and actions.

METAL uses transformations to do them. They consist of a structure description part which specifies tree structures and the feature decoration of nodes, and of a structure changing part which specifies the target tree structure and the features decoration.

Although transformations are not used in natural language systems at present, they turn out to be the most universal formal operation available. In the area of MT especially, they are the only applicable means to do all the operations described in the previous sections, namely test structural configurations, delete, add, change structures etc.; as all these operations are needed in MT (as the examples above show), transformations seem to be the best means of performing them as they have been designed for particularly this kind of operations.

Moreover, there are many problems in transfer which have not been solved yet. It would not be a wise decision to use a formalism which is restricted in itself. This is a practical reason why METAL uses transformations.

The critical point with transformations is control of their application as they are able to create infinitely many strings. Control of application of

transformations in METAL is done by applying a kind of system-external constraint: They must fit to the MIR structures. These structures are well defined both syntactically and semantically; and although control of transformations is left to the linguist, it is rather easy to do wrt MIR structures.

In addition, METAL provides excellent testing and debugging tools for transformations in its METALSHOP grammar development system (cf. Loomis 1987). We can draw and compare structures, step through the operations etc.

5 Consequences

The approach outlined above has several consequences in different system parts; some of them should be briefly mentioned.

5.1 Multiwords

The treatment of multiwords (like (1) or (2)) can follow the same patterns as outlined above:

(1) FR: machine a laver -> DE: Waschmaschine

(2) DE: logische Verbindung -> EN: session

In contrast to other systems (cf. Rothkegel 89), METAL decided not to treat multiwords as monolingual units (except for support verbs). As METAL is syntactically based, and as multiwords are defined by having a regular syntactic structure with an idiosyncratic semantic interpretation, we would always produce two syntactic readings for a multiword expression; one for the "regular" structure, and one for the "multiword semantic" structure. This does not seem to be a good solution as it doubles analysis efforts.

Therefore, we treat multiwords at the level where semantics really comes into play; and this is in transfer, where we choose readings according to semantic properties. This means that we (more or less) do not care about multiwords during syntactic analysis; they will obtain a very regular structure, which catches the observation that they behave like standard syntactic objects in analysis.

Moreover, there are cases which clearly show that multiword interpretation is language-pair specific, i.e. really belongs into the area of transfer. Consider (3) and (4):

(3) DE: runde Klammer -> EN: round bracket -> SP: parentesis

(4) DE: eckige Klammer -> EN: square bracket -> SP: recto

(5) DE: geschweifte Klammer -> EN: curly bracket -> SP: abrazadera

In a German-to-English system, we would not have to mention anything special in transfer, as transfer can be done completely in parallel for these languages. Transfer into Spanish, however, will cause a multiword translation as there is just one expression for this concept in Spanish.

Transfer of multiwords can be done with the means of far-sighted transfer described above in METAL. We can test for certain configurations and have actions on their basis (e.g. "'Klammer', if modified by an AP with head 'rund', is translated into Spanish 'parentesis', and the AP is deleted").

5.2 Generation

As complex lexical transfer produces the same MIR interpretations as would result from simple transfer, generation is not influenced by transfer problems. Complex transfer just adds additional instructions, features, and structural changes to the generation grammar; generation then takes whatever result the transfer has, on the basis of well-defined MIR structures, and generates surface strings.

This results not only in a clearer system architecture but also in real multilinguality of a system, as generation grammars are independent of the respective analysis language, and vice versa.

5.3 User aspects

Complex lexical transfer is a very critical issue in the system as it is triggered by the transfer lexicon. Transfer lexicon is accessible to the users, which usually are not linguists but translators.

The operations explained above require a certain amount of linguistic knowledge, however; this holds in particular for the action part of a transfer entry (e.g. if a NP has to be inserted and all its non-terminal features have to be given).

METAL tries to offer a well defined set of tests and operations to the users. It is a subset of the overall functionality, and it covers the most frequent cases. They are presented to the users as coding options in the interactive coding tool of METAL (Intercoder, cf. Oppenheim 87). Examples are given for each operation. If the users select some of these options, they are transformed

internally into the respective test and action transformations.

We experimented with these patterns and found that only a restricted number of structural tests suffices for most of the cases; they are presented to the users in an easily understandable way (e.g. "is there a direct object?" "What category does it have?" "Is the sentence in passive voice?") and explicated internally.

The full power of the transformational approach is open for experts only and is too difficult to use for standard users. Therefore, only reduced functionality is offered to the outside. This approach turned out to work well as the really difficult cases are part of the system lexicon anyway and need not be coded by external users at all; the cases which users have to code are rather standardised and can be captured by a well-defined subset of the METAL functionality.

6 Literature

- Alonso, J., 1988: A Model for Transfer Control in the METAL MT System. Proc COLING, Budapest, Vol. 1

- Arnold, D.G., Krauwer, S., Rosner, M., des Tombes, L., Varile, G.B., 1986: The <C,A>,T Framework in Eurotra. Proc COLING, Bonn

- Bennett, W.S., Slocum, J., 1985: The LRC Machine Translation System. in: Computational Linguistics 11

- Gebruers, R.: Valency and MT: Recent developments in the METAL System. Proc. 2nd ACL Conference on Applied NLP, Austin, Tx, 1988

- Gebruers, R., 1989: METAL Coding Principles. (unpublished)

- Golan,I., Lappin, Sh., Rimon, M., 1988: An Active Bilingual Lexicon for Machine translation. Proc COLING, Budapest

- Heid, U., Netter, Kl., Wedekind, J.: Zur Funktionsweise des Transfers auf f-Strukturen. In: EUROTRA-D Working Papers 6, Saarbruecken 1988

- Kaplan, R., Netter, K., Wedekind, J., Zaenen, A.: Translation by Structural Correspondence. Proc. ACL Europe, 1989, Manchester

- Landsbergen, J., 1989: Rosetta. Proc MT Summitt II, Munich

- Loomis, T., 1987: METALSHOP, New Functionality. Proc 2nd METAL Workshop, Leuven (unpublished)

- Oppenheim, J., 1987: INTERCODER, description of the Software. Proc 2nd METAL Workshop, Leuven (unpublished)

- Rothkegel, A., 1989: Polylexikalitaet. Verb-Nomen-Verbindungen und ihre Behandlung in Eurotra. EUROTRA-D Working Papers 17, Saarbruecken

- Schmidt, P.: Transferprobleme in EUROTRA. In: EUROTRA-D Working Papers 6, Saarbruecken 1988

- Thurmair, Gr., 1990: Recent Developments in Machine Translation. to appear in: Computers and Humanities.