III. DLT'S LINGUISTIC OPERATION (THE 'INSIDE' OF DLT).

How will the translation process within DLT be organized? Which particular linguistic methods or translation techniques are involved? To what extent will existing approaches be adopted, and what are the important new elements?

These questions will be answered in this section, which is primarily (but not exclusively) of interest to readers more or less familiar with the specialized field of MT (Hutchins, [1978 and 1982] gives an excellent overview of the field). Descriptions of MT techniques most often gravitate towards a small number of prominent systems under continuous development, often referred to in technical articles. As DLT is a newcomer in this circle, we will try to explain some of its main features by a comparative setting amidst its "big brothers". Though this may unavoidably seem to implicate a value judgement on other systems, it will help to make clear the very characteristics, the "secret" of DLT.

1. The interlingual architecture.

In the typology of MT systems, three principal concepts exist:

i. Direct (or "language-pair")
ii. Interlingual
iii. Transfer

The above sequence represents the order in which these main types of system architecture evolved during the 30-year MT history. Currently, the Transfer type (to which EUROTRA belongs) enjoys general popularity. Its predecessor, the Interlingual type (applied during the 1960's and early 1970's) is not considered to have become obsolete, but rather to have been attempted too early, being the most ambitious of all MT architectures [Hutchins, 1982].

However, the above three keywords only serve to indicate the prevailing features of an MT system, and even then they may be fallacious. For this reason, we have to elaborate the statement that DLT is an interlingual system.

1.1. Modularity.

Direct or language-pair systems, originated before the mid-1960's, became notorious for their monolithic, unmodular structure, an obstacle for clear system understanding as

well as effective improvement and maintenance.
Surely DLT, originated in the early 1980's, cannot be reckoned to the category of direct systems in this respect. Optimizing a translation module for a given language-pair is one thing, disobeying the rules for structured design and maintainability is quite another.

1.2. Economy of multilinguality.

If a multitude of languages (n TL's and m SL's) are involved, development costs can be economized by limiting the sizes of language-pair dependent system parts, or by even totally avoiding them. The former would require a number of bilingual (language-pair) modules, which increases dramatically (n * m) with the total number of languages in the system. The latter only requires the presence of monolingual system parts: one analysis module for each SL, and one synthesis module for each TL (n + m).
DLT, in this regard, is a genuine interlingual system, made up exclusively of SL-modules (to be developed independently of any TL) and TL-modules (to be developed independently of any SL).

1.3. Inclusion of the lexical component.

Prominent interlingual systems of the past (CETA, METALS [Hutchins, 1982]) were limited to a merely syntactic interlingual structure, and therefore could perhaps better be referred to as semi-interlingual. Andreyev [1967] once proposed a fully interlingual project, including a number system to convey the lexical elements. The setting up of such a system, he argued, would require an international weighting scheme which maps words to numbers according to word chain congruency of the majority of languages. It never came off the ground. Another IL featuring a numeric system for lexical elements is SLUNT [Goshawke, 1974], considered as "...a crude MT program without any syntactic analysis at all..." [Masterman, 1979], and in which the number allocation appears to be based on English patterns mainly.

Up till now, lexemes still are the 'atomic' units of translation. Thoughts of a universal semantic interlingua, comparable to symbolic logic and capable of conveying any conceivable piece of information, were discarded by Andreyev [1967], who pointed out that natural languages are much more nearer to each other than to any logic system (besides from being much more compact: natural language still appears to be an unsurpassed invention in human communication). In CETA and METALS, "no attempt was

made to decompose lexical items into semantic primitives..." [Hutchins, 1982]. More recent projects, transfer-type systems as GETA, SUSY, EUROTRA, all rely on a 1-1 bilingual dictionary match, "the heart of any MT program" [Masterman, 1979].

DLT's position here is unique: DLT's interlingua (IL) is nothing else than a streamlined version of Esperanto, which is an existing universal auxiliary language. Though by origin Esperanto classifies as a semi-artificial language (only its word roots derive from Indo-European languages), its use as an instrument of human communication since 1887 shows no essential difference with 'natural' (i.e. ethnic) languages.
For DLT, the adoption of Esperanto means the availability of a full interlingua (IL), serving both as a lexical as well as syntactic interface between the SL- and TL-sides of the system. Not only does Esperanto provide the necessary basic vocabulary, it also offers standardized terminology for several fields and reasonable extension prospects for others [see Section V.3]. As to the comprehensiveness of the IL, DLT can therefore be considered a fully interlingual system.

1.4. Interlingual or double translation?

An opponent of DLT could argue: "First, everything is translated from SL (say English) to Esperanto; then, everything is translated from Esperanto to TL (say French); doesn't that double the translation effort?"
He would be right, but only as long his scope of evaluation is limited to just one and the same SL-TL pair. Faced with the problem to optimally solve the translation from English to French (completely disregarding the extension possibilities towards other language pairs), the introduction of an intermediate 'universal' process stage would mean an unnecessary overhead burden (both for the Anglo-French development team and for the computer's operational process). The evident justification of intermediate stages and 'standard' interfaces lies in the economy of multiple module interconnections [see paragraph 'Economy of Multilinguality' above]. The computer and software industry presents a lot of analogies: in the connection between a computer A and a terminal B, a standard protocol interface is justified because A-B is one out of many possible pairs.

In defending the need for 'double' translation, we imply that linguistically this phenomenon exists. DLT's IL, though different from common Esperanto, linguistically still qualifies as a natural language. This property is

exploited to provide development and maintenance teams easy access to the system's main interface.
However, from a system design viewpoint the IL is a semi-product, hidden to the end users. The concept of 'semi-product' is important in a system which calls itself 'distributed', both in time and location. Notwithstanding the linguistic nature of the IL, its functioning in DLT's overall system architecture, with regard to coordinative development as well as distributed operation, makes it a real interlingua.

1.5. Overall architecture vs. subsystem.

Having asserted that DLT's overall architecture is interlingual, but having admitted that linguistically one may speak of double translation, it becomes obvious to view each subsystem (either the SL-IL or the IL-TL 'half' of the system) as a bilingual module and a separate translation process. This can then again be characterized in terms of direct/transfer/interlingual, according to its own internal organization. As for the SL-IL subsystem, this most strongly resembles the direct architecture [see 4.2], whereas its IL-TL counterpart corresponds more to a transfer-type system [see 4.3].

Summarizing, we state that DLT's overall system architecture is definitely interlingual. Linguistically, one may speak of 'double' translation, with the SL-IL subsystem resembling the direct, the IL-TL subsystem resembling the transfer type.

It is interesting to quote Hutchins [1982] in this context:
"...there is now considerable agreement on the basic strategy, i.e. a 'transfer' system.... However, this apparent convergence of approaches in recent years is confined to the design of fully automatic systems... not involving any human intervention during the translation process itself."
It should be kept in mind that DLT is a semi-automatic system: the SL-IL part relies on human interaction, only the IL-TL part is fully automatic.

2. The functioning of the main interface.

We will now briefly compare the functioning of DLT's interlingua (IL) interface with the transfer mechanisms of competitive systems like EUROTRA, GETA and SUSY (see [Hutchins, 1982] for introductory remarks about these systems). This may help to uncover differences that could get overlooked otherwise, will thereby prevent confusion and better bring out the essential characteristics of DLT.

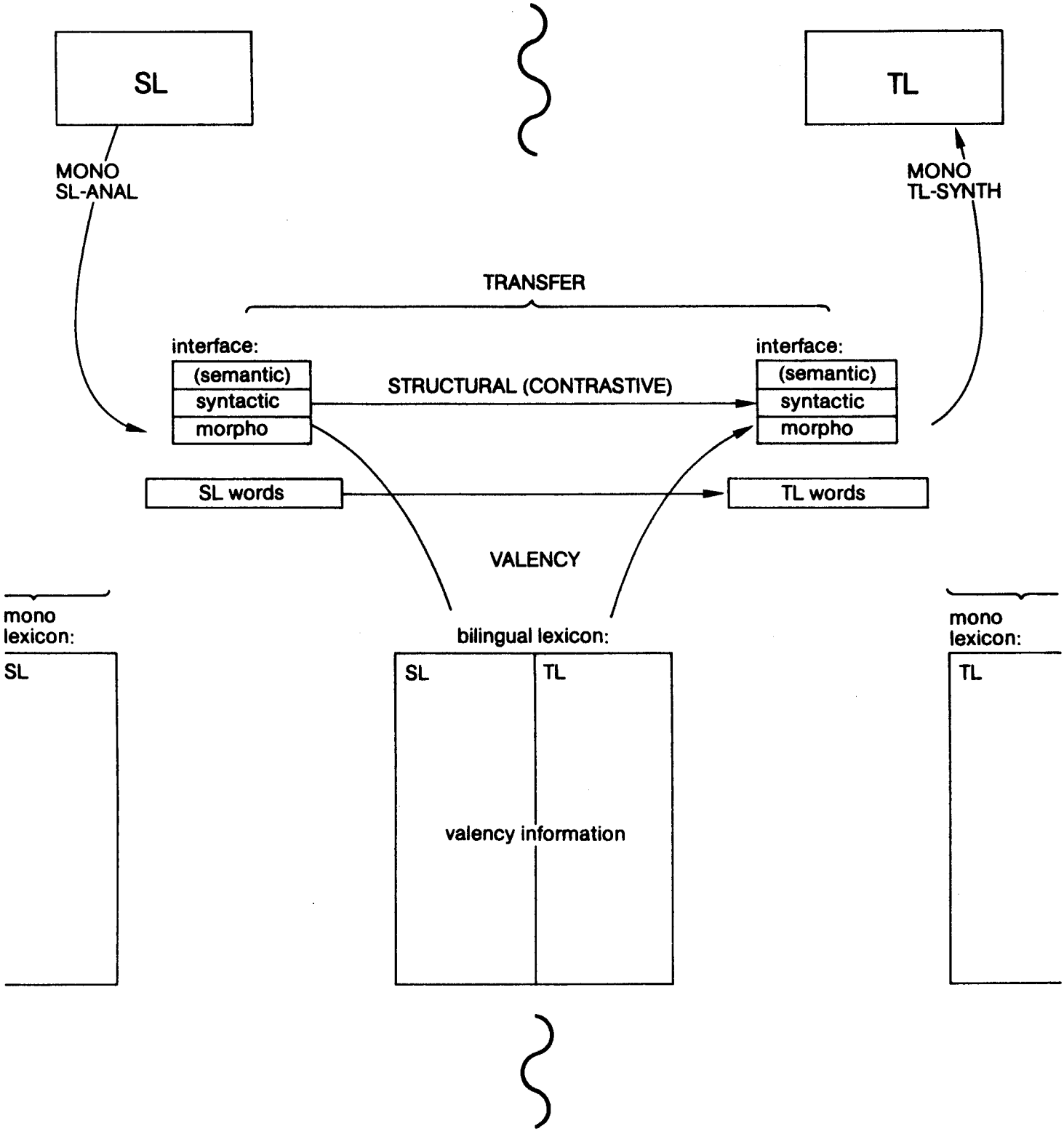


Fig. III-1. Category 1: 'Classical' transfer configuration, relying on dictionary-contained valency information in a transfer stage that includes structural as well as lexical conversions. The emphasis is on a comprehensive SL-TL dictionary. The SL-analysis and the TL-synthesis stages are monolingual, as are their dictionaries.

Judged by the form of the main interface (SL-TL) and the way of forwarding information across this interface, we distinguish 3 important configurations [see figs. III-1, III-2 and III-3] which will be briefly described and discussed now.

2.1. Transfer system with a substantial transfer stage.

The MT-systems GETA (ARIANE-78) of Grenoble and SUSY of Saarbrücken belong to this category, which precedes EUROTRA (category 2 below) in historical sequence.

The transfer stage is dominated by a structural transfer, covering the contrastive syntax of the language pair (SL-TL).
Note that clearly 3 substantial process stages (Analysis, Transfer, Synthesis) and accordingly 2 interfaces are involved: one at the SL- and one at the TL-side. The transfer stage maps the contents of the former onto the latter.

So far as the interface contents are composed of abstract formatives, they are mainly limited to morphological, morpho-syntactic (incl. syntagmatic) and syntactic-function variables, with some semantic additions recently (e.g. 'LOC QUO' to denote destination [see Luckhardt, 1983]).
Both the structural and lexical transfer rely heavily on a bilingual (SL-TL) lexicon, in particular on the valency information (of verbs, nouns, etc.) contained herein.

In this configuration, the presence of a structural, contrastive transfer and the possibility of direct dictionary-based valency translation has obviously reduced and postponed the need for an extensive abstract semantic interface.

2.2. Transfer system with a rudimentary transfer stage.

The outstanding MT-system representing this category is EUROTRA, the principal design criteria of which have been stated [King, 1982] as multilinguality, economy of development and collaboration of independent (SL- and TL-) teams.

A striking feature here is the reduction of the transfer stage to a minimal-size process: the transfer is actually limited to a 1-1 lexical substitution, based on a 'thin' bilingual (SL-TL) lexicon.
Work not done in the transfer process obviously takes place now in the SL-analysis or TL-synthesis module.

Therefore, although the transfer as a process is minute, the interface structure exchanged by it is extensive in this configuration.

The emphasis here is on the exchange of semantic information in the form of abstract formatives:

AGENT	TIME
PATIENT	SPACE
BENEFICIARY	ORIGIN
EXPERIENCER	DESTINATION
CO-AGENT	TRAJECTORY
INSTRUMENT	MATERIAL SOURCE
MANNER	etc.

These variables are attached to the various constituents (NP's, PP's, adverbs) of the sentence, i.e. to the nodes of its parse tree. They serve to explicitly convey semantic roles and relations from the SL- to the TL-modules (to be developed by 'monolingual' SL- and TL-teams respectively). Though they are supported to a considerable extent by theoretical work in the fields of general linguistics (Case Grammar [Fillmore, 1968]) and artificial intelligence [Wilks, 1973], their practical and large-scale use in a multilingual MT system is still new.

However, 'semantic relations' is only the top layer of a 4-layered interface structure of abstract formatives, which also includes the more traditional syntactic variables. The idea is that the TL-module attempts a translation based on the semantic layer first. If this layer does not provide sufficient information, the module will descend to a lower layer and translate in what one could call a fall-back mode.

Valency boundness information is provided for by a layer in between the semantic and the syntactic variables. This layer contains a constituent's valency relation (if any) to the predicate. The variable values (DEEP SUBJECT, DEEP OBJECT, ...) are assigned by the SL-module, on the basis of verb valency information in the monolingual SL-lexicon. In contrast to the category-1 (see above) type of system, valency translation plays a secondary role here. It has the status of a fall-back provision. If it is used, it is only an indirect mechanism, depending upon the exchange of abstract formatives between the SL- and TL-side. The 'thin' bilingual lexicon used in the transfer process does not contain valency information.

Structural contrastive transforms (related or unrelated with valency boundness) are entirely handled by the TL-modules. The interface structure from which these modules

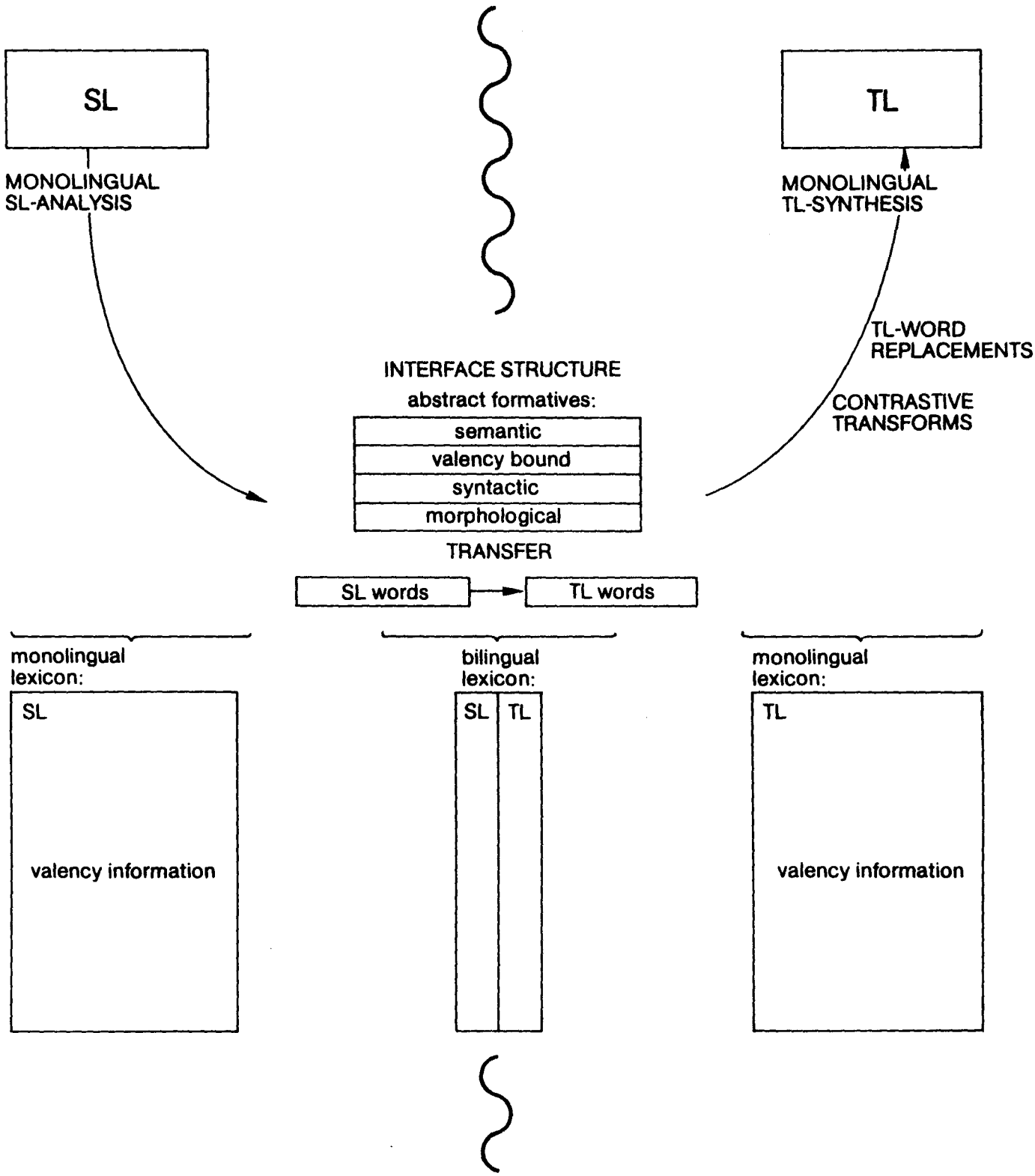


Fig. III-2. Category 2: Configuration with a minimal-size transfer stage, approaching an interlingual system. The transfer has been reduced to a 'narrow' lexical bridge (1-1 conversion of lexemes). The emphasis is on the extensive exchange of abstract formatives.

depart is a parse tree with TL-words at its leaves and extensively labeled with abstract variables from all the above-mentioned 4 layers. The underlying sentence structure is still SL, but because of the abstractness of of transferred information, the monolingual TL-module can cope with it.
The TL-modules may indeed be called monolingual, if one refers to absence of explicit SL-elements in them, and to the fact that (at least in principle) they can be developed by monolingual teams. As for the complexity of the process however, the 'synthesis' here covers more or less a 'hidden bilingual translation'. Apart from structural contrastive transforms ('hidden SL-TL transfers'), TL-words (originally substituted for SL-words by the regular and straightforward 1-1 lexical transfer process) may have to be replaced by better TL-words, in connection with valency information and style rules from the TL-module's lexicon.

Also the SL-modules are subject to higher requirements in this configuration. They have to yield values for an extensive set of abstract variables, including semantic relations as the most difficult ones. The 'depth' of abstract analysis is critical in category-2 systems.

The transfer system with only a minimal transfer stage approaches a semi-interlingual system, the interlingua consisting of abstract semantic and syntactic formatives. The effectivity of this main interface very much depends upon the sharpness of definition of many variables, especially of the semantic 'cases', which tend to show grey boundary areas. Writing about EUROTRA, King and Perschke [1982] admit that the semantic level is "quite ambitious", and that "...it would be rash to assume that an accurate semantic interpretation can always be established".

Without harmonizing on clear demarcations and adhering to them in practice, the different SL- and TL-development teams can evidently not produce a working system, except for one that will constantly be forced into a fall-back mode (translation based on the valency or even syntactic level). In itself, the built-in safety mechanism is an excellent design feature, but if it would be called upon regularly rather than exceptionally, a category-2 system (lacking a direct, dictionary-based valency translation) could compare much less favorably with the other categories.

The extensiveness of the interface structure is emphasized by a statement like the following, on EUROTRA: "...there is no upper limit on the amount of information a group may store in the interface structure..." [King,

1982]. Though the existence of a lower limit (i.e. a core common and obligatory to all groups) is affirmed, this invitation to 'proliferate' optional interface elements presents a potential danger: it could affect the overall effectivity of the structure as a standard interface amidst a multiplicity of SL- and TL-modules.

2.3. Interlingual system with a lexical-formatives interface.

This is the interface architecture pertaining to DLT.

In this category, only 2 process stages remain, clearly separated by an interlingual (IL) main interface. There is no such thing as a 3rd process in the middle, not even a rudimentary one.

The interface comprises the lexical elements, neither as SL-words, nor as TL-words, but as IL-words (or, more precisely, IL-stems). The IL has a lexicon of its own (in contrast to the so-called "interlinguas" developed for MT in the 1960's, which where limited to syntax).

Moreover, the morphological, syntactic and (the limited set of) semantic elements of the interface are not effectuated by abstract grammatical formatives, but by lexical formatives as well. This relates to the extremely regular and grammatically transparant structure of the IL, for which DLT has adopted a modified subset of Esperanto. The outstanding property of this language is the strict invariance and autonomy of its morphemes, including those corresponding with "grammatical endings". The largely isolating nature of the Esperanto language [Piron, 1978] culminates in the inclusion of function morphemes (i.e. the closed class of grammatical endings, prepositions, etc.) into the lexicon, e.g.:

morphological:	'-os'	=	FUTURE TENSE
morpho-syntactic:	'-a'	=	ADJECTIVE
syntactic function:	'-n'	=	DIRECT OBJECT
semantic relation:	'per'	=	INSTRUMENT

The systematic realization of abstract grammatical labels by lexical formatives [see ..... for much more details] also covers function words, in particular prepositions. In Esperanto, these are much more precisely defined than in other (natural) languages, and therefore prevent the need for explicit and separate (extralingual) semantic case marks. This means that all interface data are contained in one compact and consistent IL format, the comprehensive grammar definition of which [see Section IV] makes up the interface description.

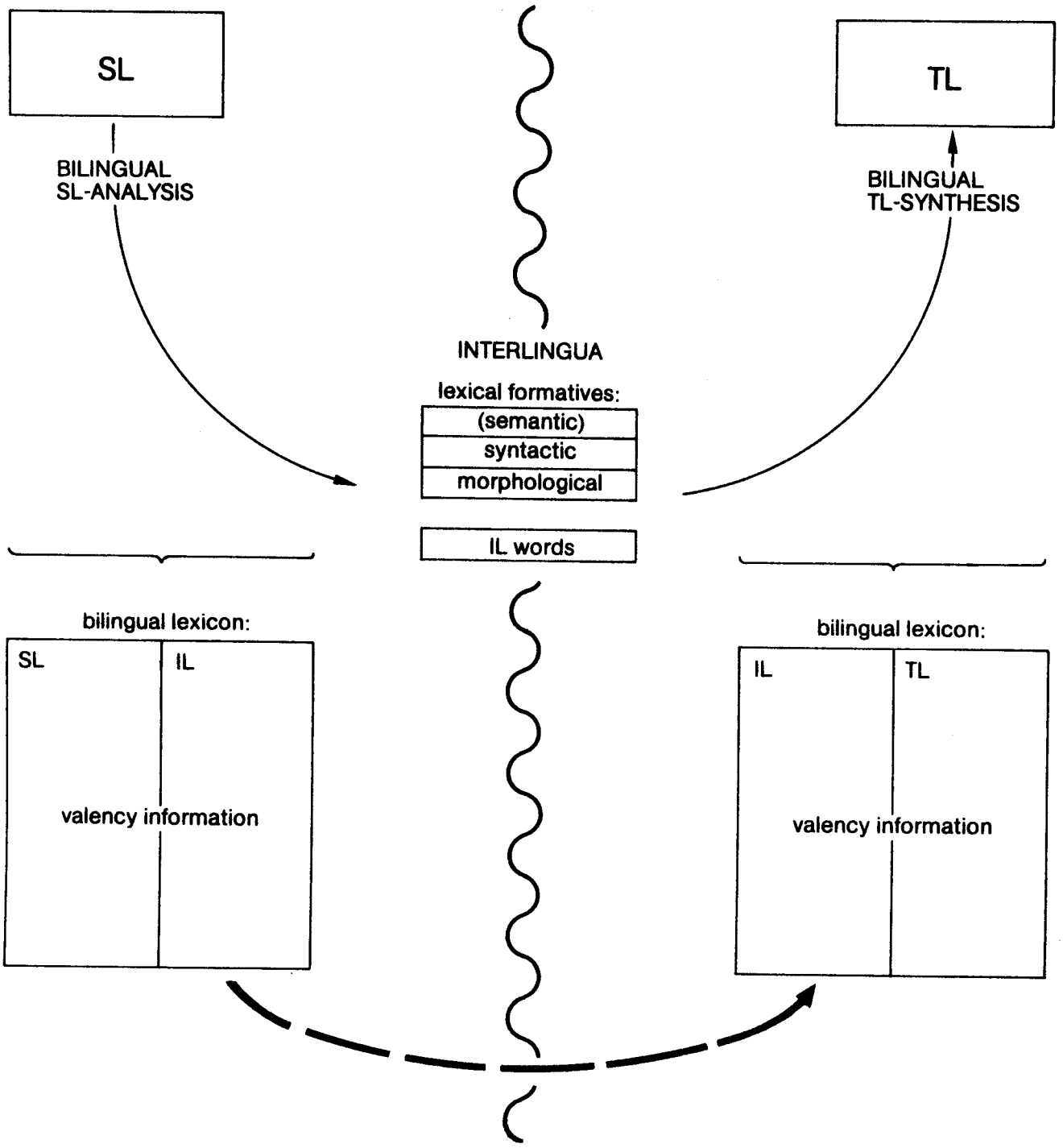


Fig. III-3. Category 3: Interlingual configuration, featuring an IL with lexical formatives only. The dashed arrow symbolizes the 'wide' lexical bridge that is formed by the presence of comprehensive IL dictionary columns at both sides of the interface. This characterizes DLT.

Each SL- or TL-module must be bilingual (SL-IL or IL-TL) in this category. However, there will only be as many such modules as there are SL- and TL-languages, so it will pay to optimize each of them.
Of course, each bilingual module relies on a bilingual lexicon (SL-IL or IL-TL). This enables the use of direct, dictionary-based valency translation within a bilingual module.
The result is a configuration, in which not so much the depth of abstract SL-analysis or the extent of the interface is involved, but much more the quality and compatibility of two bilingual lexicons is at stake.

In addition to the variable interface data passed (in the form of IL-sentences) during the translation process, the two collaborating (SL- and TL-) modules are bridged by the common presence of an IL-column in their bilingual dictionaries. This represents what one could call the fixed interface data, forming a bridge which is much 'broader' than the relatively 'narrow' channel of variable data, a situation which appears to be in accordance with the "pivotal" role of dictionaries in an MT system, as affirmed by Knowles [1982] and Masterman [1979].

Of course, a price has to be paid for the total disappearance of the 3rd process stage. Each of the two remaining stages, the SL-analysis and the TL-synthesis, now becomes bilingual, involving the IL [this will be dealt with below, see 4.2 and 4.3]. This bilinguality of dedicated SL- or TL-modules is a linguistic property that may be hidden to DLT end users, but will be unconcealed to the system's development and maintenance staff, who will profit from it [see 5.2].

Summarizing the comparison between MT-system architectures of the categories 1, 2 and 3, we can say that the role of the dictionary and valency translation is more central in 1 and 3, whereas 2 and 3 are nearer to each other regarding multilinguality.

An interesting state of relations is expressed by fig. III-4, which gives a conceptual view of the difference in interface composition for the three categories. The overwhelming presence of SL-TL bilingual components at category-1's main interface indicates that this configuration scores low on "Economy of Multilinguality".
Category 2, though much better in this respect, still has a small proportion of bilingual SL-TL data.
The diagram shows the shares of variable data, i.e. those concerned with the instantaneous text unit (sentence) being processed, and the fixed data, which includes information

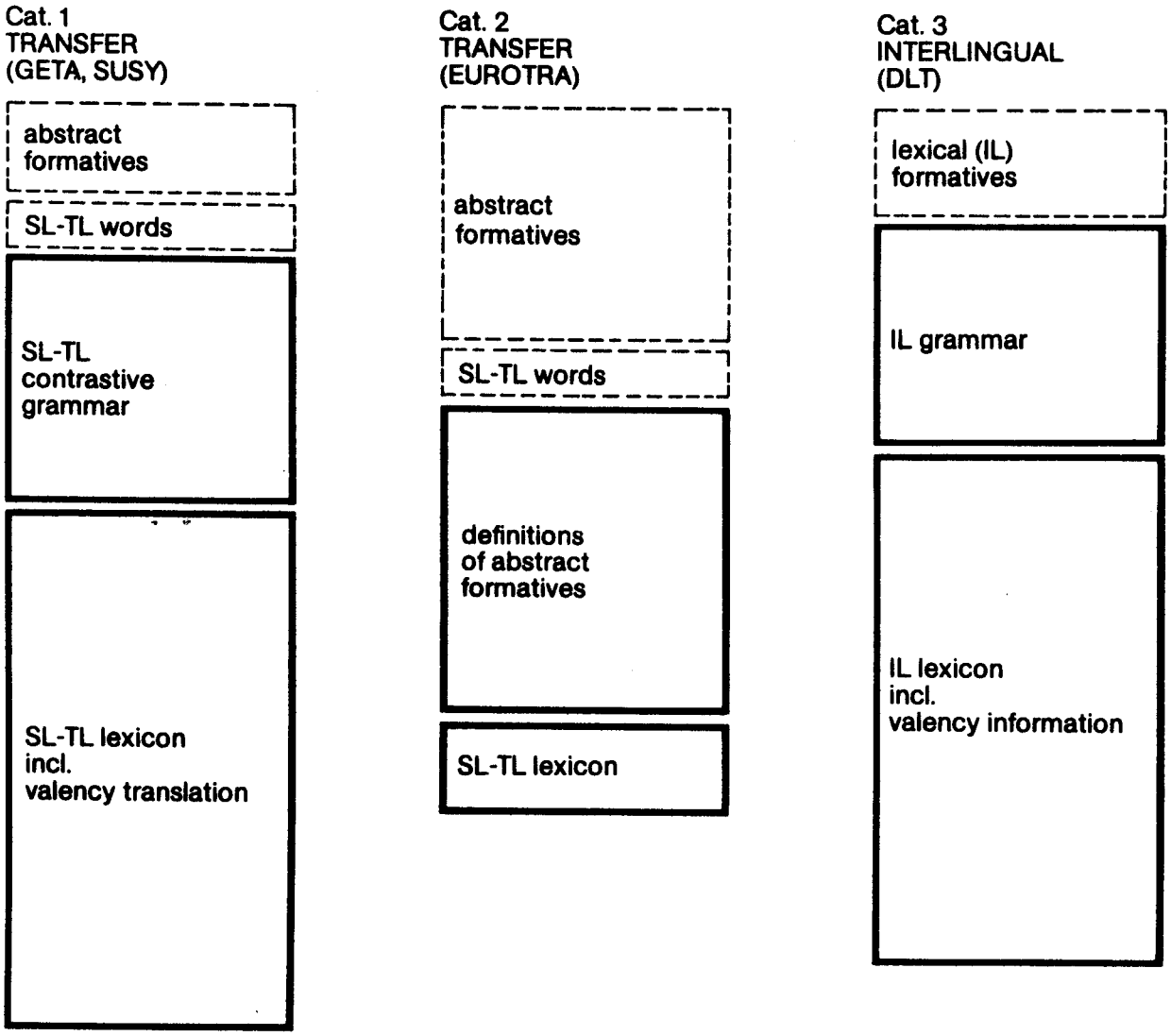


Fig. III-4. Comparison of main-interface composition. The diagram gives an impression of the proportion of 'variable' and 'fixed' interface-data, for 3 MT-system categories. The dashed boxes denote 'variable', the solid boxes 'fixed' data.

commonly used (copies of which exist) at both sides of the main interface, i.e. a grammar, a lexicon, or set of definitions of abstract formatives. In an active MT system, the variable data form a continuous flow of information, the volume of which determines both speed and cost of transmission. The fixed data are only (if at all) exchanged on a logistic support basis, e.g. on-line overnight updating of dictionary entries, and therefore do not so strongly require compactness.

As mentioned elsewhere [see Hutchins, 1982] MT was once compared with the deciphering of cryptographic code. If, for a moment, we use this analogy, not for the decoding the of SL, but for the decoding of main-interface data (e.g. IL in category 3) by the receiving TL-module, we observe the following difference: Category 2 is characterized by long, complex "messages" (with a relatively high degree of redundancy), while Category 3 has remarkably short "messages" (with a relatively low degree of redundancy) but a "key" (the fixed data) which is more extensive than the one of Category 2.
Whereas EUROTRA tends to expand the "message", DLT tends to expand the "key".

Also notice the homogeneity of the category 3 interface components: IL (sentence), IL (grammar) and IL (lexicon). This underlines the fact that DLT's interface specification is essentially the definition (by grammar and dictionary) of one coherent language (the Esperanto-based IL), and thereby profits from the decades-long experience gained with the existing auxiliary language (Esperanto) underlying it.
Compared to this, the category 2 and 3 interfaces are heterogeneous: they are partly composed of SL-TL data, partly of a system of abstract formatives, the semantic variables of which have only taken shape during the last 8 years or so, and have hardly been tried out in practice.

One could regard the choice of MT architecture as the selection of "software packages" for an intermediate language or abstract interface structure: Category 2 might then appear to offer a "better" product (in terms of linguistics or translation theory), whereas Category 3 would come out with a more conservative product which has largely been debugged and tried out in practice.

3. Convergence, divergence and disambiguation.

The proper handling of ambiguity is a major problem in MT. Therefore, the design and characterization of a new MT system would be incomplete without considering it. In this section, ambiguity will be studied in relation with MT architecture and main-interface configurations discussed above (in IV.1, ambiguity will be the theme in a contrastive overview of the IL). We will deal with ambiguity and disambiguation in a wide sense, and include the phenomena that are connected with it, such as convergence and divergence.

3.1. What is ambiguity?

The terms 'ambiguity' and 'unambiguity' (and likewise 'disambiguation') often mean different things to different people. In order to prevent confusion, we will first define our terms and briefly explain some basic concepts and distinctions. These must be taken into account as properties of language, without regard to any translation.

3.1.1. Equivocality vs. ambiguity.

First of all, we have to separate intended from unintended ambiguities. The former, also named 'equivocalities', deliberately occur in political writing, consultant's reports, advertising, word games etc. Preservation of such deliberate ambiguities in a translation, if not impossible, is definitely outside the scope of the MT-system proposed in this report [see also Section VII].

In this report, unless stated explicitly otherwise, we will assume that the ambiguities discussed are unintended. Even stronger, in many cases the ambiguities will not even be noticed by humans, who unconsciously use their knowledge of the context, situation and world, in order to smoothly write and understand a sentence. To a machine however, many words and expressions appear to be ambiguous [see also 3.1.3]. Sager [1981] uses the term 'false ambiguity' in this connection.

3.1.2. Structural vs. lexical ambiguity.

'Structural' ambiguity (also called 'syntactic' ambiguity) refers to the two or more possible readings of the same sentence, regardless of any confusion about the meaning of each content word, e.g.:

(1) Put the hammer in the toolbox on the table!

which has 3 possible readings [example from Hendrix, 1981].
To structural ambiguity we will also reckon multiple-meaning problems of function words (pronouns, prepositions, etc., the so-called closed-class words):

(2) They were surprised by the sea.

and 'part-of-speech ambiguity' (also referred to as 'homography' by some authors), such as the frequent confusion between verbs and nouns in English:

(3) Many hands make light work.

On the other hand, 'lexical' ambiguity (also called 'semantic' ambiguity), refers to multiple meanings of lexemes (open-class words), as shown by enumeration in dictionary entries, and includes 'homonymy' (non-related meanings of the same word):

(4) He took the lead.

(5) The rebels took the port last night.

and 'polysemy' (related meanings of the same word):

(6a) He played the trumpet in the morning.
(6b) He played chess in the afternoon.
(6c) He played Hamlet in the evening.

As the latter example [borrowed from Lyons, 1977] indicates, polysemy is kind of open-ended, and it is difficult to demarcate ambiguity from vagueness and unspecificity [Dik, 1979].
Also, there is no sharp distinction between polysemy and homonymy [Ullman, 1962] (we will come back upon this in section 3.2).
As our interest is in written language (the DLT design presented in this report does not include speech input), we do not consider homophones.

Within a sentence, one can have a mixture of ambiguity types, such as the structural ambiguity and the homonymy of 'saw' in the example of fig. III-5. Often - but not always - the various ambiguities within a sentence are interrelated. In case of our example, the knowledge (by indication in a lexicon entry) that 'binoculars' are no cutting instrument, would exclude the combination d.
In 4.2.3.4c we will define the concept of 'local' ambiguity, to distinguish those ambiguities that are independent of other ambiguities in the same sentence.

A very important boundary area between structural and lexical

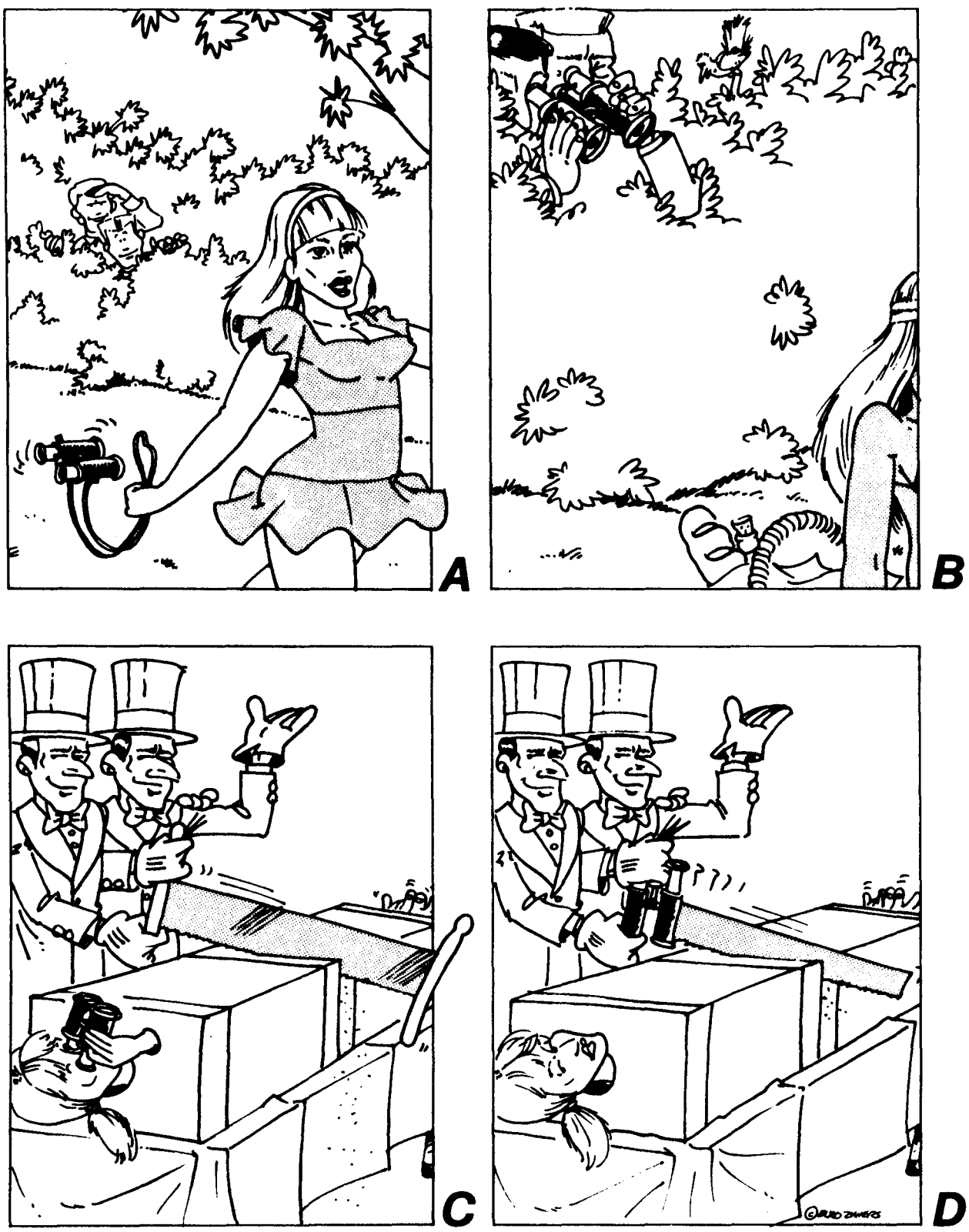


Fig. III-5. The big problem in natural language processing by computers is ambiguity. To a machine, many sentences appear ambiguous. This one shows both structural and lexical ambiguity.

ambiguity is formed by verbs whose meaning depends on the presence of certain prepositions in the microcontext:

(7a) User **sa** voiture.
(7b) User **de sa** voiture.

In these cases, disambiguation will be based on syntactic valency and strict subcategorization information contained in the verb's lexicon entry. A similar case involves discontinuities, e.g. the Dutch:

(8a) Hij merkte het varken op.
(He noticed the pig.)
(8b) Hij merkte het varken.
(He marked the pig.)

In this large boundary area between purely structural and purely lexical ambiguity, both the microcontext (the sentence and its syntax) and the lexicon (valency information) play a part in the disambiguation.

A special type of structural ambiguity, also a potential trouble source in MT, is anaphoric ambiguity:

(9) It surprised the consultant that the computer did respond to his question.

in which the anaphoric element 'his' could refer to 'the consultant', but also to another person or entity mentioned in the preceding sentences.

3.1.3. Unambiguity in terms of parsability.

We could also say: "Unambiguity in terms of the receiver's capability". What is ambiguous for the one need not be ambiguous for the other. In contrast to the intended ambiguities, humans will hardly be aware of most of the (unintended) ambiguities that could be pointed out in natural language: they do not regard these phenomena as "ambiguities". Though there are certainly interesting differences among humans (depending upon their knowledge of and familiarity with the subject, context or situation), we focus here on the difference between a human receiver and a machine, and differences among machines. The "machine" is the working combination of programs and data available for the analysis of the language, i.e. the parser, the dictionary and the knowledge-bank (if any).

One therefore can specify unambiguity in terms of a certain machine, e.g. a "simple parser (equipped with only a morpho-

syntactic dictionary so and so ...)", or a "complex parser (equipped with a deep semantics dictionary or a knowledge bank)".
Except for the weaponry used by the parser, also the degree of detail of its output needs to be specified, in order to make the definition of "unambiguity" (in terms of a machine's capability) complete. The output of a parser is usually the hierarchical constituents structure of the input sentence, i.e. a labelled tree structure. What is important is the refinement and precision put into a constituent's label or enclosed in its lexical formatives. E.g. does the parse result distinguish between various semantic categories of adjuncts:

PLACE
TIME
MANNER
INSTRUMENT

etc., and if so, is there a further subdivision such as:

PLACE	WHERE
	WHERE...TO
	WHERE...FROM

Whereas current MT systems carry this kind of information as abstract grammatical formatives within complex labels of parse tree nodes, DLT will utilize a coherent and similarly precise system of lexical formatives.
On the other hand, one could imagine a parser which just delivers

ADVERBIAL

as abstract label to a constituent composed of lexical formatives which are ambiguous with respect to the semantic category of the adverbial, e.g.:

(10) ...with the binoculars...

(which can be instrumental as well as associative).
If also the predicate's dictionary entry does not contain verb valency information to disambiguate the semantics of the preposition "with" in the sentence "he saw the girl with the binoculars", we have parser output which is not very precise.

A distinction related with the parsing process is the distinction between 'partial' and 'total' ambiguity. As long as a sentence has only been parsed partially (for a one-pass LR parse: the input pointer has not yet reached the end-of-sentence), the partial parse result at a given "snapshot" can be ambiguous, in which case we call the ambiguity 'partial'

[Agricola, 1968]. As the parse proceeds through the remaining part of the sentence, some of these partial ambiguities may be resolved automatically. Others may persist even after the sentence has been parsed totally, and are therefore called 'total' ambiguities.

In discussing ambiguity, we usually mean total ambiguity (unless we scrutinize the parsing process). As for partial ambiguity, note that it includes lexical as well as structural ambiguity: at a parse snapshot T, a lexeme may be ('partially') ambiguous; at snapshot T+1, the same lexeme may be disambiguated by the dictionary entry information of a lexeme to the right of it.
Partial ambiguity should not be confused with 'local' ambiguity. The latter refers to the concept of mutual independency of ambiguities in the same sentence [see 4.2.3.4c].

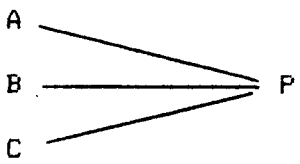
3.2. Ambiguity across language boundaries.

For language translation, we are not concerned with ambiguity as a phenomenon in each of the languages separately, but rather with the passing, resolving, inducing and preserving of ambiguity across language boundaries. The paragraphs of this section serve to explain some relevant notions and terms, still independent of a particular MT system.

3.2.1. Convergence, divergence and 1-1 correspondence.

If we follow a text unit (sentence) on its course through the entire SL-TL translation process, we can observe the following patterns (in all diagrams following in this section, the process direction is from left to right):

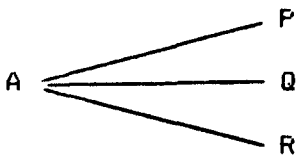
Convergence:



Two or more entities (A,B,C,...) in a given process stage are translated ("converge") into one entity (P) of the next stage. This can happen when going from SL to TL, but also (in systems with an IL) from SL to IL or from IL to TL.
The entities can be lexemes, but also grammatical values (e.g. 'Passè simple', 'Passè composè', 'Imparfait' for

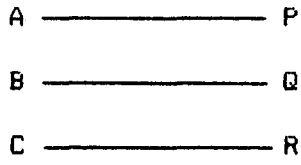
French and 'Past' for Japanese).

Divergence:



One entity (A) in a given process stage is translated ("diverges") into two or more entities (P,Q,R,...) of the next stage. Again, this can happen when going from SL to TL, SL to IL, or IL to TL, and for grammatical as well as lexical entities.

1-1 Correspondence:



As a third possibility, a 1-1 correspondence may exist between series of entities at neighboring stages.

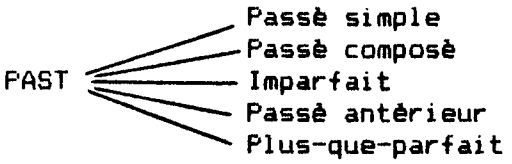
The relative frequency of occurrence of each of these three patterns depends on the process stage, MT architecture involved, and the particular language pair (SL-TL).

Evidently, 1-1 correspondence is the ideal pattern: it does not involve any complexity and still appears to preserve all information. In reality, this pattern exists for considerable parts of the vocabulary and a number of grammatical values; its importance is even increased by the fact that one will often apply 1-1 correspondence as an approximation mechanism.

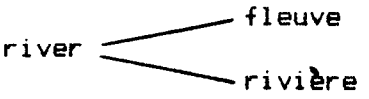
Convergence looks unproblematic if one takes into account the direction of the process (from left to right in the above diagrams). It appears to be a 1-1 mapping mechanism then. Only, one tends to suspect a loss of information, going (as it seems) from a more refined to a less refined representation stage.

3.2.2. Divergence-resolving selection and disambiguation.

Divergence implies a 1-to-2 or 1-to-many mapping. E.g. how to map:



or, taking lexical examples:



Clearly, if to be done well, these conversions require a decision mechanism. The translation quality is very sensitive to the performance of this mechanism. Some MT systems, notably in the past, simply worked without decision mechanism and showed all the possible alternatives in the TL output text (words printed above each other or separated by a slash); other systems simply produce (without any sophistication) one alternative as standard output, and the others on request only.

One could say that the decision mechanisms connected with divergence patterns represent the heart of the translation machinery. The choice out of two or more mapping alternatives is conditioned by the current context (of the entity under translation). The decision procedure itself must have been encoded inside the grammar or dictionary entries. This so-called 'fixed' information, in the form of tiny or extensive algorithms, operates on the 'variable' data. The latter is usually the microcontext (i.e. the words within the current phrase or sentence).
The procedures or algorithms involved can be referred to as 'word-choice procedures', 'tense-selection procedure' etc. In MT literature, one can also find the term 'computation' instead of 'selection', e.g. 'computation of correct tenses'.

The use of the term 'disambiguation' here is practically a matter of taste and attitude, i.e. considering parallel denotations in different languages, which one do we regard as 'normal', which one as 'extra refined' and which one as 'ambiguous' information? If the divergence is from ambiguous to normal, we will probably call it disambiguation, if it goes from normal to extra refined, we may not [see also 3.3].

In contrast to microcontext-based decision procedures, algo-

rithmics based on macrocontext and knowledge-banks are much further in the future [see 6.2]. They are related with the largely unexplored fields of discourse analysis and AI (Artificial Intelligence). In DLT, knowledge-of-the-world and macrocontext analysis will eventually support the translation process. In the long transitional stage before, DLT's SL-IL stage will rely on interactive human assistance, whereas its IL-TL stage will simply be not capable to decide correctly in cases like the 'fleuve/rivière' example given above.

3.2.3. Non-universality of homonyms.

What happens with homonyms during translation? To what extent are they 'universal' (i.e. reflected in the majority of languages to be handled in the system)?

Homonymy (unrelated meanings of the same lexical item) can be regarded as an accidental phenomenon in the orthographic development of a language, which practically precludes universality of this ambiguity type. If there is a universal homonym, it can usually be explained by the presence of a border-line territory between homonymy and polysemy, and by the fact that some present-day homonyms can be traced back to apparent cases of polysemy (often metaphors) in the past, e.g.:

Dutch	German	French	English	Swedish	Italian
pen	Feder	plume	pen	penna	penna

where all the languages share the meaning-pair: 1. writing instrument, 2. part of bird. But if the common etymological origin is in the more remote past, any 'universality' of the homonym is likely to have been resolved by distinctive orthography in many languages, e.g. the homonymy of the Dutch word 'bank' (1. piece of furniture, 2. financial institution) appears to have only the German 'Bank' as parallel ambiguity (and only for the singular):

Dutch	German	French	English	Swedish	Italian
	(Bänke)	banc	bench	bänk	banco
bank	Bank (Banken)	banque	bank	bank	banca

Thus, apart from some exceptions and boundary cases, and also apart from 'terminologic homonymy' (to be treated in 3.2.4), homonymy is a monolingual peculiarity. To make this point more clear, let us consider the following German words, the combination of which we will call 'schizonymy':

1. können
2. in Büchsen einmachen

These two German lexical units have no mutual relationship that would justify their inclusion in one common entry of a monolingual (German) dictionary. The only relationship that ties them together is a bilingual relationship, and concerns the homonymy of 'can' in English.
It is unlikely that lexicographers would take such 'schizonymy' into account when grouping words into dictionary entries. The composition of a monolingual lexicon must clearly not be mixed with the peculiarities of other languages.

3.2.4. Universality of terminologic homonyms.

There is a special class of lexical homonymy which does show 'universality' on a significant scale, and which may be very relevant in the application of modern MT systems: as soon as we enter specific fields of terminology, e.g.:

	German	French	English
Botany:			
Mathematics:	Wurzel	racine	root
Linguistics:			
Agriculture:			
Bacteriology:	Kultur	culture	culture
History:			
Meteorology:	Depression	dépression	depression
Psychiatry:			

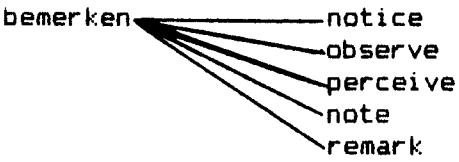
we are confronted with what we may call 'terminologic homonymy' (Horecký [1982] uses the terms 'isonymy' and 'tautonymy'), sometimes combined with a internationalization of Latin-based words (in particular between French and English). More or less related with this is the phenomenon that some languages adopt archaic (biblical) terms for 20th-century concepts (e.g. Hebrew), though the number of 'universals' of this particular type will not be substantial.
An interesting aspect of 'terminologic homonymy' is that it reduces the operational disambiguation load: the MT system needs not be directed as to the choice of a particular terminologic vocabulary [see also V.3.1].

3.2.5. The polysemy-divergence-synonymy triangle.

A frequent pattern in the course of a translation process is the potential lexical divergence with a polysemous word at the

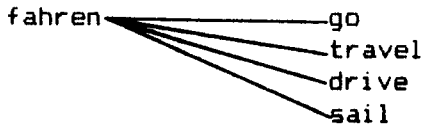
one side, and a number of 'near-synonyms' at the other side, e.g.:

POLYSEMY 'NEAR-SYNONYMY'



The word choice involved in resolving this divergence is not quite arbitrary, as indicated by the term 'near-synonyms': these words differ in shade of meaning, style or register. Often, the consequences of a random word choice in such a situation may be limited to a stylistically less perfect and somewhat unusual TL text (a well-known prey for human post-editors).
But certainly also a simply incorrect TL text could result, unless a mechanism for the proper selection of a near-synonym is made available. Such a mechanism will take into account the syntactic valency of a word (transitive or intransitive verb, prepositional objects possible) and semantic selection restrictions, based on lexical subcategorization and inspection of the word's microcontext (same phrase or sentence). In such a way, more complex 'triangles' may be resolved:

POLYSEMY 'NEAR-SYNONYMY'

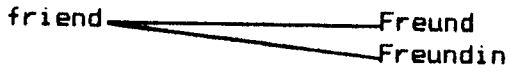


The 'disambiguation' of an SL polysemy thus corresponds with the selection of a TL 'near-synonym'.
It is interesting to quote Newmark [1981] in this connection:

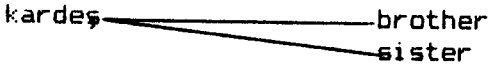
> "Where the target language has a number of synonyms to express the sense of a source language word, the translator should choose the word he considers stylistically most fitting rather than the word that most obviously translates the source language word."

In the MT field, this strategy is confirmed by the 'pivotal' role of the dictionary, as indicated - among others - by Knowles [1982] and Kelly [1982]. In effect, a dictionary entry should contain idioms, collocations and all those connections with other words that are known to be statistically significant, including of course technical terminology [see also IV.4.2 and IV.4.3].

As an extension of 'near-synonymy', one can consider 'hyponymy'. In a translation process, hyponymy can become acute when one language has a superordinate term which is lacking in the other language, e.g.:



or, taking an example with Turkish as SL:



Resolving this type of divergence further increases the requirements for comprehensiveness of MT lexicon entries, which has been defended by Knowles [1982] (partially on the basis of recent Russian progress in the field of automated lexical data bases):

> "It is the duty of the lexicographers - and that includes the elaborators of MT lexical data bases - to 'capture' all the paradigmatic relationships of a word and as many of the superordinate set-to-set relationships as are possible."

In general, the choice of a TL-hyponym will (more often than is the case with near-synonyms) depend on macrocontext and knowledge-of-the-world, and to automate such a procedure will be a very ambitious task. What this paragraph intends to make clear is that - during the translation process - the need for a selection has to be signalled from the way a monolingual dictionary entry has been composed.

In contrast to the explicit non-universality of 'accidental' homonyms discussed in 3.2.3, this paragraph emphasizes the 'universality' (by and large) of related-meaning clusters (near-synonymy, hyponymy), which is given concrete form by MT lexicon design rules ('What to include in one lexicon entry'). On their turn, the latter may yield an operational criterion for what is homonymy and what is polysemy. For instance the divergence of

suit	1. Prozess
	2. Klage
	3. Rechtshandel
	4. Gesuch
	5. Bitte
	6. Werbung
	7. Antrag
	8. Satz
	9. Garnitur
	10. Anzug
	11. Kostüm

will NOT correspond with one comprehensive entry (containing all the 11 items) of the monolingual German lexicon, simply because there is no relation whatsoever between 1-7 and 8-11, neither semantically nor from the point of view of formal identity or common etymology in German. Instead, the pattern

suit
1. Prozess
   Klage
   Rechtshandel
   Gesuch
   Bitte
   Werbung
   Antrag

2. Satz
   Garnitur
   Anzug
   Kostüm

which requires resolving a homonymy (choice between 1 and 2) and a polysemy (choice within 1 or 2) in succession, will best fulfil the requirements of passing information across the language boundary, at each side of which monolingual lexicons exist.

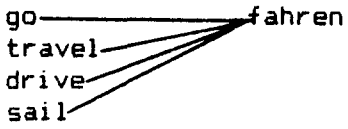
The considerations in this paragraph indicate a guideline for the design of MT lexicons (also referred to as 'lexical data bases' or 'dictionaries' in this report). One important principle that stands out is the so-called '<u>content addressability</u>', a concept of computer science which means that items similar or related in contents are stored close together in the computer's memory. This principle can be applied excellently on near-synonyms and hyponyms.

In case of multiple related lexical items in a dictionary entry, there is a clear need for lexicon-contained <u>procedures</u> that serve as selection criteria for these items. This is common to all MT systems, but they differ in the way the lexicons are partitioned (e.g. SYSTRAN has its contextual procedures in a separate dictionary), in the formalism or language in which the procedures (sometimes called 'disambiguation rules') are coded, the system of semantic categories and subcategories (e.g. 'profession', 'means of communication') upon which they rely, etc. [see also 3.3].

3.2.6. Induced ambiguity and asymmetry.

The above paragraphs 3.2.2 and 3.2.5 focussed on divergence and the reduction of ambiguity (by word selection procedures)

in the course of the translation process. As for convergence (already mentioned in 3.2.1), this will usually be the reverse of the phenomenon explained in 3.2.5: a TL has not always the differentiating power to reflect all shades of meaning of a SL near-synonym cluster, or the TL simply has a superordinate term which the IL lacks:



In such a case however, it would not be fair to call the German TL-output 'ambiguous': if at all, it is the German language, not the translation, which one should consider 'ambiguous' here. The same is true for the Turkish superordinate term 'kardeş', which is used for 'sister' as well as for 'brother'.
In fact, it is better to speak of 'vagueness' than of 'ambiguity' of the TL, in these cases [Dik, 1979].

Apart from polysemy and hyponymy, also homonomy can cause convergence. Take for instance the following translation:

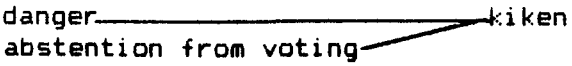
(11a) Die Dänen können wieder in der Nordsee fischen.
(11b) The Danes can fish in the North Sea again.

A criticizer could raise that the English translation is ambiguous, as it might be just as well the translation of the German SL-sentence:

(11c) Die Dänen büchsen wieder Fisch in der Nordsee ein.

We will call this 'induced ambiguity', and be aware that it is a peculiarity of the TL (sometimes connected with part-of-speech ambiguity), which would also occur in human translation. In general, induced homonymy will not cause much problems (through the ages, natural languages tended to get rid of homonyms when they proved to be a frequent source of confusion). In the attempt for FAHQT, polysemy will be a much more serious problem than homonymy.

Possibly, special attention should be given to a language like (written) Japanese (the syllabic 'kana' or the romanized 'romaji'), with a significantly higher degree of homonymities than European (written) languages, e.g.:



(examples from [Smith, 1982]). For Chinese, a phonetic writing system that includes pitch has been developed for computer purposes [Asiagraphics, 1982]; without the encoding of pitch, too many homonyms would arise.

Convergence, and especially induced ambiguity, is turned into divergence and a disambiguation requirement if one reverses SL and TL. This makes apparent the general asymmetry of the translation process: in the one direction (convergence), we observe the introduction of vagueness or ambiguity, i.e. some form of loss of information, in the other direction (divergence), we are faced with the necessity of adding information, by disambiguation or motivated word selection.

3.2.7. Neutral-form preservation.

Several important types of structural ambiguity appear to be universal, and therefore a deliberate non-disambiguation can be tempting. A typical example is PP-ambiguity concerning the subordinate relation with noun phrases, i.e. having the linear string

N PP PP PP PP PP

(N = noun, PP = prepositional phrase), the question arises: Is each PP subordinate to the same head-noun N, or are they all subordinate to the noun enclosed in their immediate predecessor? [see also IV.1.3.4].

Disambiguation always adds to the processing load during translation. If the only result of disambiguation is a temporary (i.e. during some intermediate representation stage of the translation process) refinement which is lost again after a subsequent convergence, there is clearly no much point in taking all the trouble.
In order to translate, complete analysis is not always necessary: translation is generally a less demanding process than full comprehension of a text. This applies for instance to the translation of anaphorics: in certain cases, the ambiguity of the antecedent does not prevent correct translation.
However, the problem is that it does prevent correct translation in other (not essentially different) cases, due to trivial morphological details such as the absence of separate masculine/feminine third person plural pronouns in one language, and the presence of them in another. E.g. in

(12) The soldiers fired at the women, and they fell on the ground.

(an often-cited example from [Wilks, 1979]), preservation of the ambiguity is impossible because of a language like French ('ils'/'elles').

Though some structural ambiguities appear to be universal at first sight, the snag is that they are not really, even if we limit our view to the languages covered by a multilingual MT system. For instance the clausal/phrasal PP-ambiguity in:

(13) They saw the girl with the binoculars.

is 'universal' only for English, German, Dutch, and NOT for French and Russian.

Another example is the ambiguity of premodifier and postmodifier scope in the presence of a conjunction, e.g.:

(14) Old men and women from Amsterdam.

sometime presented [Dik, 1968] as a specimen of universal ambiguity. In translation, knowledge of the exact scope can be very essential, as proven by the following - obviously incorrect - English-to-French conversion:

(15a) Pregnant women and children.
(15b) Des femmes et enfants enceintes.

(an MT error once made by SYSTRAN [Pigott, 1982]).
When French acts as SL, many French postmodifying PP's will correspond with premodifiers, compound strings or composite words in other languages (English, German):

capital à risques	risk capital risikotragendes Kapital
chargement en pontée	deck cargo Deckladung
chômage de friction	frictional unemployment friktionelle Arbeitslosigkeit
répartition des frais	cost allocation Kostenaufteilung

Especially in cases of established terminology, bilingual lexicon entries will trigger the conversion of a collocation or multi-word technical term, thus avoiding any question about modifier scope [see also 4.4.2].

Regarding the difference between 'restrictive' and 'non-restrictive' relative clauses, [Swales, 1981] states:

"...any study of scientific writing immediately throws up many examples of relative clauses that appear to be neutral in this respect...".

In a multilingual translation process, the transfer of a 'neutral' form, in addition to two alternative forms, would seem to require a ternary instead of a binary representation mechanism to handle ambiguities (e.g. for relative clauses, the absence or presence of the comma alone would not be enough). In the SL, the 'neutral' form often coincides with one of the alternatives (absence of a comma for relative clauses), which would again require disambiguation: "Do you really mean a restrictive clause or do you mean a DON'T CARE?".

In fact, the neutral form preservation is an attempt to preserve ambiguity itself. Preserving ambiguity is not the same as ignoring ambiguity. If nothing is done, ambiguity in the SL may get lost in the TL, but the result may be more specific or restricted than is desirable.
Deliberate preservation and reconstruction of ambiguities in the TL is an MT design decision, involving additional costs and provisions. Moreover, for many ambiguity patterns, the neutral form will be desired in some instances (example 14 above) but not in others (example 15). This implies the need, not only for a ternary representation mechanism, but also for a ternary selection process instead of two-way disambiguation, e.g. (in case of 14 and 15):

DO YOU MEAN:
1. old men
2. old men and old women
3. DON'T CARE

DO YOU MEAN:
1. women from Amsterdam
2. men from Amsterdam and women from Amsterdam
3. DON'T CARE

DO YOU MEAN:
1. pregnant women
2. pregnant women and pregnant children
3. DON'T CARE

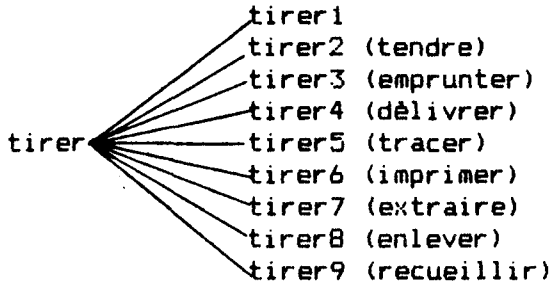
(in case of 15, the selection process could as well rely on lexicon-based selectional restrictions).

3.3. Process-stage sequence and module work-split.

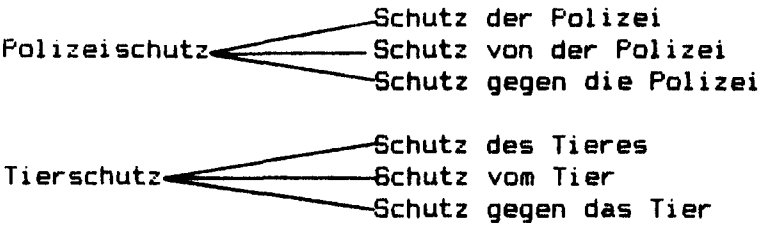
After having defined and explained our terms, we will briefly look at convergence and divergence in the DLT translation process, compared to other MT architectures (we will follow the division of MT-systems into 3 categories shown in section 3.2). This will throw more light upon the functioning of DLT, in particular with regard to the work-mix of what is done at the SL-side (SL-modules) and the TL-side (TL-modules). The division of work between modules is a matter of design and development, in its turn affecting the operational process stage sequence of an MT system.
If we follow the linguistic data (usually in units of one sentence) on their path through the entire translation process, we distinguish an SL stretch, a TL stretch and (in some systems) a SL-TL transfer stretch. Each of these stretches can again have a sequence of several process-stages, according to the work-split between the modules.
Fig. III-6 presents schematic characterizations of the divergence and disambiguation along the translation process-stage sequence (in these and subsequent diagrams, the process direction is always from left to right). Although not every occurrence of divergence implies disambiguation, the latter plays an important role here.

3.3.1. Category-1 systems.

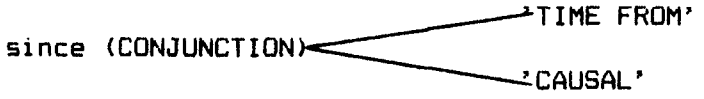
These are characterized by a monolingual divergence in the SL-analysis (e.g. the semantic disambiguation 'SEDAM' in SUSY [Luckhardt, 1983]), independent of any other language (IL or TL). This means that, within the SL stretch of the translation process, we can distinguish 2 stages, one with the lexemes as found in the input text, the other with names of lexical items that are more refined or precise. An example is:



If the SL-input contains word compositions, the second stage may employ synonyms that lend themselves better to translation (this comes down to resolving the syntactic ambiguities sometimes hidden in word compositions), e.g.:

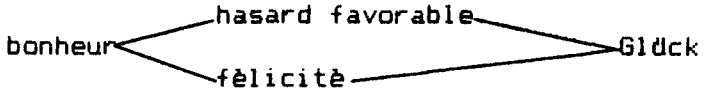


A similar 2-stage SL-stretch can occur with the second stage consisting of grammatical formatives, e.g.:

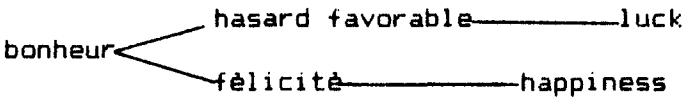


However, the existence of monolingual divergence on itself does not always imply a monolingual disambiguation in the category-1 systems. Sometimes, multiple interpretations of ambiguous words or sentences are simply transferred to the TL-side of the system, where they may all be translated successively and even appear as such in the TL-output (such as in the German-to-English translation of the METALS system [Bruderer, 1978: 259-269]).

If the divergence on the SL-stretch does lead to disambiguation there, we will call it 'unconditional disambiguation'. In principle, the disambiguation can also be postponed to the Transfer or Synthesis stage, an arrangement which we will refer to as 'postponed disambiguation': the existence of alternative readings is recognized in the monolingual SL-analysis, but no effort is spent to resolve it there.
This could be defended by pointing at the presence of occasional or even frequent (depending upon the SL-TL pair) 'universal ambiguities', which make disambiguation seem worthless in many cases, e.g.:



A symmetrical divergence-convergence pattern like this illustrates the fact that language translation is not quite as demanding as language understanding (as has been stated, among others, by Wilks [1979]): different meanings do not always result in different translations!).
But the same SL-element could necessitate disambiguation in case of a different TL:



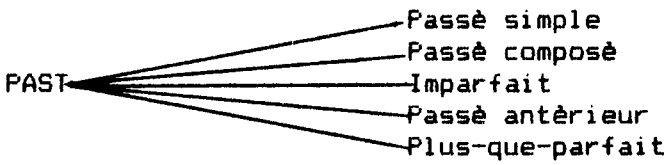
Category-1 systems typically have a substantial Transfer stage, covering all the SL-TL contrastive conversions, structural as well as lexical ones. In such a configuration, postponement of disambiguation till the Transfer stage appears to be logical, and preserves the multilinguality of the system. A lot of preparatory work and a limited amount of unconditional disambiguation can be performed within the monolingual SL-module, e.g. resolution of part-of-speech ambiguity (cfr. 'Homographenanalyse' in SUSY [Maas, 1982]). See also fig. III-6a.

3.3.2. Category-2 systems.

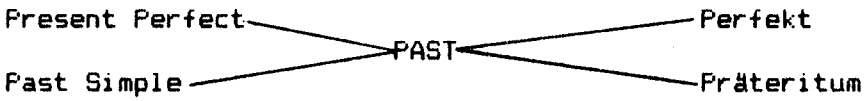
Here too we find monolingual divergence in the SL-analysis stage. For a part, disambiguation will be performed there. The remaining amount has to be postponed and arranged outside the SL-modules.

Unconditional disambiguation will be attractive if the ambiguity involved is 'non-universal' (i.e. does not exist with the majority of languages to be handled by the multilingual system).
Postponed disambiguation will be preferable for those cases of divergence that are mirrored in most other languages of the system ('universal ambiguities') and only present themselves as real ambiguities (i.e. need to be resolved) for one or two TL's. One could regard those cases as peculiarities of these TL's.

Postponed disambiguations may include grammatical elements such as the choice of proper tense in the TL, e.g.:



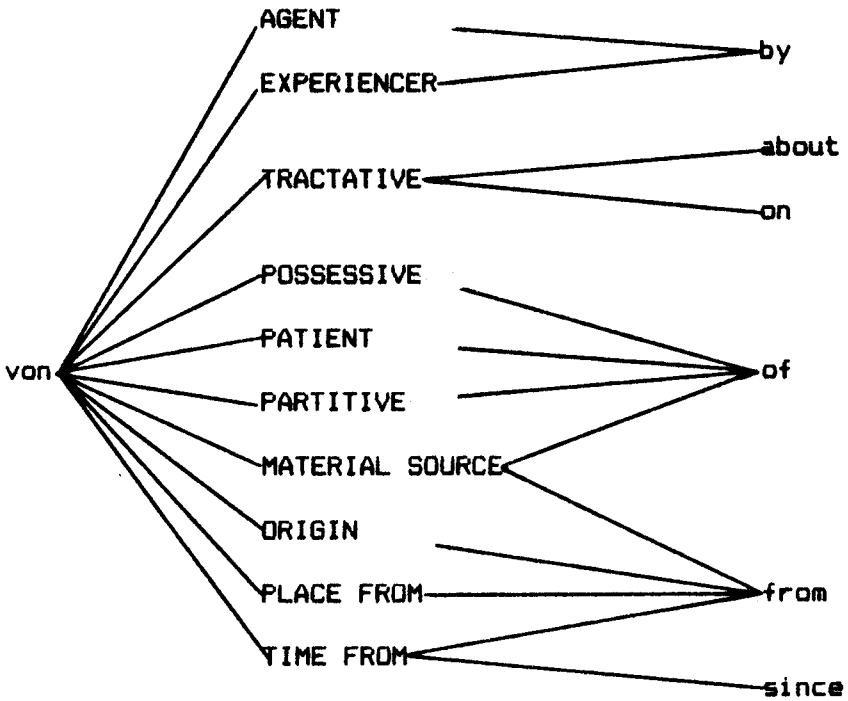
where the TL has a more refined system of past tenses, not only more refined compared to just one SL, but compared to the majority of languages in the system. This majority is reflected in the intersection of tenses, i.e. a conceptual system of tense distinctions known in all languages (PAST, PRESENT, FUTURE) and therefore serving as a communication vehicle and pivot for SL-side convergence and subsequent TL-side divergence:



This example of an English-to-German convergence-divergence pattern emphasizes the lack of 1-1 correspondence between the English 'Present Perfect', 'Past Simple' and the German 'Perfekt' and 'Präteritum' respectively. The TL-module will have to make the proper choice of past tense by taking into account the German (micro)context and style rules.

Whereas the grammatical elements (generally morphological) that express tense, verbal aspect, voice etc. are much burdened with the (historically developed) peculiarities of individual languages, and therefore lend themselves preferably to the pattern of postponed disambiguation, the contrary is true for semantic elements.
The (artificially developed) semantic layer of category-2 systems exists by virtue of its deliberate language-independency. Therefore, category-2 systems unconditionally map SL-elements onto semantic elements, i.e. unconditional disambiguation, even if this would be followed by convergence at the TL-side, as in the following English-to-German example:

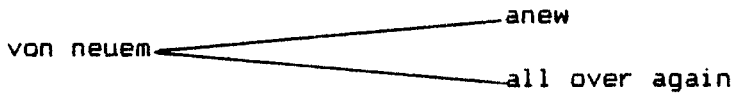
Often, the SL-side divergence will prove to have been indispensable, as in this German-to-English example:



The pattern here is predominantly divergent, though some of the abstract semantic formatives converge back onto one

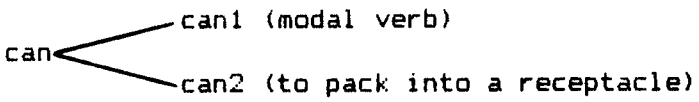
TL-element. Note the moderate divergence remaining at the TL-side ('about','on'; 'from','since'), which may seem a matter of style rather than translation precision, though the "wrong" choice could introduce TL-ambiguity unrelated with the SL-original ('He spoke on the Queen Mary'). The above diagram does not show the potential support of valency information (a fall-back provision in category-2 systems), nor the transfer of collocations and idioms:

von einem abfallen	to break with somebody
von Rechts wegen	by law
von vornherein	from the outset

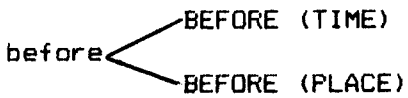


Summarizing the various disambiguations in category-2 systems, we observe the following work-split between modules:

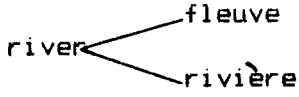
SL-analysis: a. Disambiguation of 'non-universal' (SL-specific) lexical ambiguities, e.g.:



b. Disambiguation of structural ambiguities, including the sense of SL function-words, by mapping SL-elements onto semantic formatives known to all modules:



Transfer & TL-synthesis: c. Postponed disambiguation of 'universal' lexical ambiguities for those TL's that form an exception (TL-peculiarity) on the lexical item's ambiguity pattern as observed for the majority of languages, e.g.:



d. Postponed disambiguation of grammatical formatives which correspond with more refined TL-specific structural elements.

These are mainly morphological elements (expressing tense, verbal aspect, voice etc.).

As for the work-split between Transfer and TL-synthesis, we remind that, in category-2 systems, the Transfer stage is purely lexical and must be kept to a bare minimum of bilingual operations. Therefore, item 'd' above will have to be covered entirely by the TL-synthesis modules.
Regarding point 'c' above, one can observe that those lexical ambiguities which we refer to as 'universal' (which must be understood in the practical sense of "reflected by the majority of languages likely to be dealt with in the MT system") are related primarily with polysemy and synonymy, NOT with homonymy!
Because disambiguation of lexical polysemy comes down to a choice out of TL 'near-synonyms' [see 3.2.5], the following strategy is appropriate for the work-split among modules in category-2 systems:

i. In the bilingual Transfer stage: Provisional substitution of the polysemous SL-word by just one item (presumably the top one) of the TL-bundle of near-synonyms.

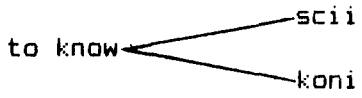
ii. In the monolingual TL-Synthesis stage: Replacement of the provisionally taken TL-word by the stylistically and collocationally most proper of its near-synonyms.

Fig. III-6b gives a schematic characterization of the above, in particular in comparison with category-1 systems.

3.3.3. Category-3 systems.

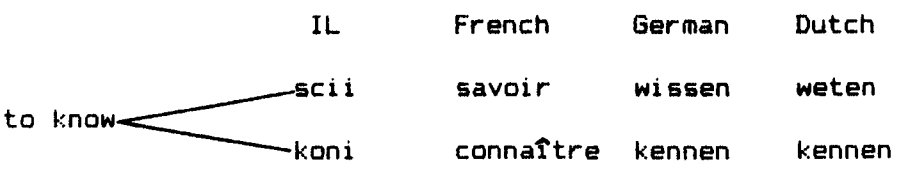
3.3.3.1. IL-directed disambiguation.

Here the IL provides the operational criterion for what is ambiguous and what is not, at the SL-IL conversion in the SL module, e.g.:



i.e. the English verb 'to know' is considered operationally ambiguous and has to be disambiguated because the SL-IL lexicon entry shows two equivalents in the IL, NOT because of any divergence with respect to TL's.
Of course, SL-IL divergence will often coincide with SL-TL divergence for several TL's, which is not a mere coincidence, taken into account the fact that the Esperanto-based IL

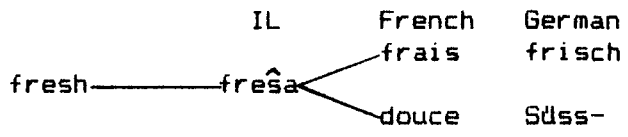
reflects prevailing lexical and structural patterns of national languages:



But, during the translation process of a category-3 system (DLT), only the SL-IL relation determines the SL's ambiguity, even if the lexical pattern in the IL would not well reflect the pattern of most surrounding languages (which would be exceptional).

The significance of the SL-IL relation as sole disambiguation criterion at the SL-side, is that it solves the difficulties arising from two otherwise vague concepts, at least concepts that are hard to define precisely: 'ambiguity' and 'universal ambiguity'.
E.g.: is 'to know' ambiguous? For the German TL-module it is, but decisions at the SL-module should not be guided by circumstances at a single TL-module. One could point to the occurence of two translations in a number of languages, and speak of 'the majority of languages in the system', but this is a dangerous criterion: if one applies it by intuition, it will be inexact and subjective, and if one attempts to apply it in an exact manner (systematically checking four or more languages), the amount of work will soon become prohibitive [in fact such an effort has gone in the creation of Esperanto and its vocabulary, a lifetime's work of the Pole Zamenhof (1859-1917) and other outstanding Esperantologists].
One could try to establish the ambiguity of 'to know' monolingually, by scrutinizing its various meanings and meaning shades in English. But this too is dangerous in a translation system, as one would risk a lot of unnecessary disambiguation of 'universal ambiguities'. The latter concept again is not easy to define for operational purposes.

The presence of the IL and the IL-directed disambiguation in DLT thus resolves the dilemma of what to disambiguate unconditionally (at the SL-side) and what to postpone for disambiguation at a later stage of the translation process: it does NOT remove the need for postponed disambiguation. At the IL-TL conversion, divergence is by all means possible, e.g.:



We will refer to this (not deliberately postponed) disambiguation at the TL-side as '<u>residual disambiguation</u>'. It results from a difference in refinement between the IL and TL. In DLT, the TL-module has to cope with this residual disambiguation fully automatically (in contrast to the SL-module, which performs the IL-directed disambiguation semi-automatically). The TL-module will strongly rely on ample IL-TL lexicon-entry information for this purpose: in the above example, the German TL-module would find the translation 'Süsswasser' for the IL collocation 'freŝa akvo'.
Thus, lexical divergence at the TL-side is generally handled as a TL-specific lexicon-supported word choice problem, which one may call stylistic as it becomes more subtle.

Fig. III-6c summarizes the above in a schematic diagram.

With all disambiguation activity determined by the IL, its degree of refinement - both structurally and lexically - will have an overwhelming effect on the whole translation process and system design. In a sense, the IL must incorporate the results of a contrastive-linguistics exploration in a well-balanced way, such that it neutralizes the syntactic differences between languages (Somers [1983: 151] considers this a more realistic target than 'an entirely language-independent theoretical representation'). This requirement is largely met by Esperanto, and further pursued by the DLT restrictions and modifications on top of it [see Section IV].

As for 'neutral form preservation' [see 3.2.7], this has consciously been given up in the design of DLT: it would considerably complicate the IL-design, and appears to be in conflict with the philosophy of a - basically - unambiguous IL. It also would complicate instead of alleviate the disambiguation procedures at the SL-side.

3.3.3.2. Over-disambiguation and 'explosiveness'.

It does not help when the IL is more refined or more abstract than the large majority of SL- and TL-candidates. In particular, any tendency towards a 'logic' IL should be avoided. As Andreyev [1967: 5] states, "<u>Human languages are much nearer to each other, than to symbols of any variation of a logical system, and consequently an effective IL must be sufficiently similar to spoken human languages</u>".

What an English receiver may experience as unambiguous, a German or a French receiver may call ambiguous:

English:	you	German:	du ? Sie ?	French:	tu ? vous ?

With regard to this very simple example, it can be said that DLT's IL has the English pattern (only 1 form for the 2nd person). The German and the French may therefore call the IL "ambiguous" on this point. This particular example is relatively harmless (the problem does not occur at all in many text types and can be isolated well). But there are much more important problems of this kind:

Articles.
The IL (following Esperanto) only has a definite, no indefinite article. This makes the IL "ambiguous" to the French, English, German TL-modules (to name a few). But to a Russian, Turkish or Japanese TL-module, the IL would seem completely unambiguous on this particular point.

Tenses.
The IL has 3 simple tenses: past, present and future. This will cause "ambiguity" if French, German, or English is the TL:

IL:	oni decidis	French:	on décida on a décidé on décidait
		German:	man beschloss man hat beschlossen
		English:	one decided one has decided

Not even all possibilities (e.g. the French 'Plus-que-parfait') have been included in this simplified scheme. On the other hand, Japanese has only 1 past tense, so for a Japanese TL-module the IL's past tense will not appear ambiguous.
Important to notice is that, among the various past tenses of French, German and English, no 1-to-1 correspondence exists. The situation is complicated by the fact that tense is partially intermixed with verbal aspect and non-temporal modalities ("Interferential" or "Narrative Past") in German and Turkish.

Due to the multitude and variety of grammatical concepts and categories in different languages (SL's and TL's), and the absence of a simple mapping mechanism between them, any attempt to create an IL which is unambiguous to each TL in every respect would result in an 'explosive' IL, loaded with a gigantic burden of grammatical distinctions (16 cases, 12 tenses, 4 genders, etc.). In [Andreyev, 1967] this type of IL is also referred to as "summarizing type" (originally proposed by Mel'chuk), and described as a disguise of a set of pairwise SL-TL translations.

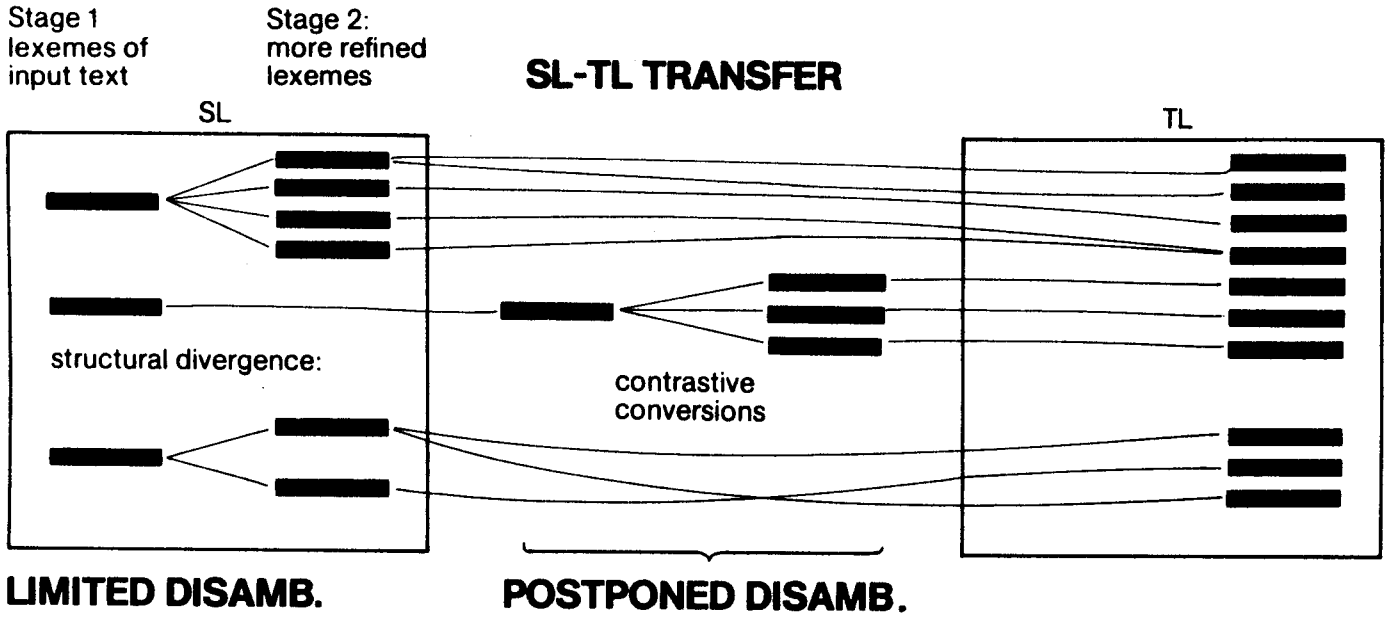


Fig. III-6a. Category-1 systems (GETA, SUSY): the disambiguation is concentrated at the Transfer stretch.

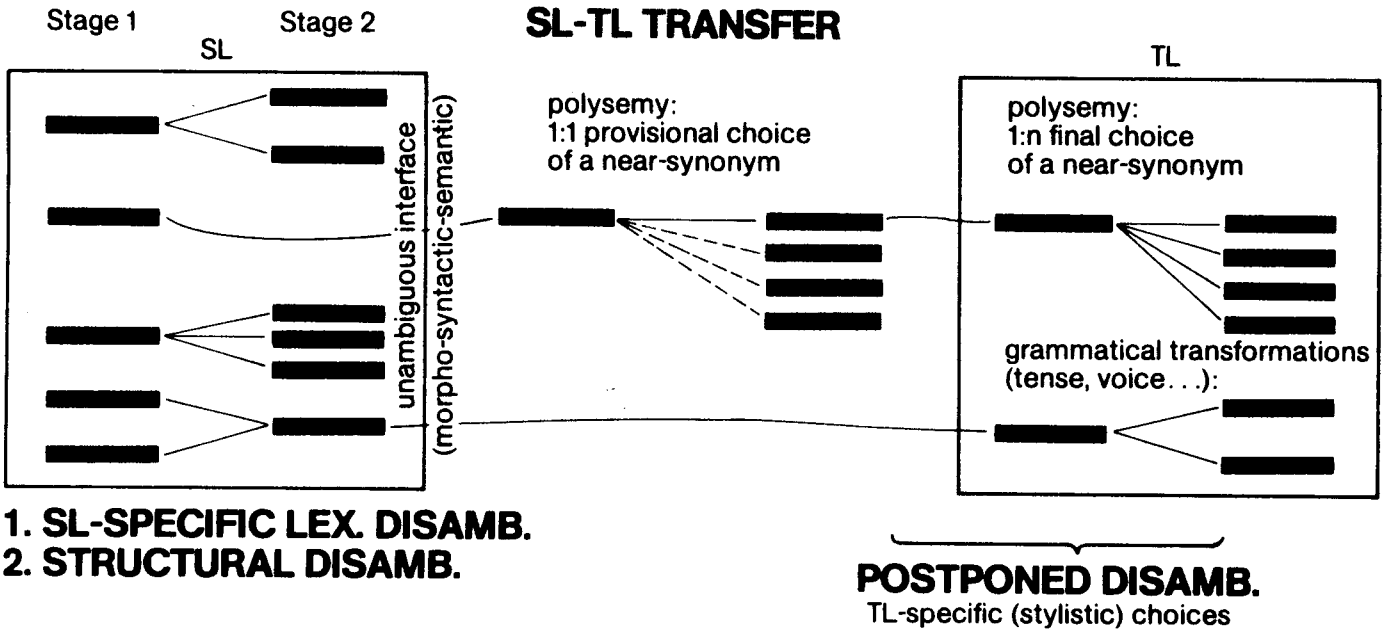


Fig. III-6b. Category-2 systems (EUROTRA): largely, a polarization of disambiguation at the SL- and the TL-stretches takes place.

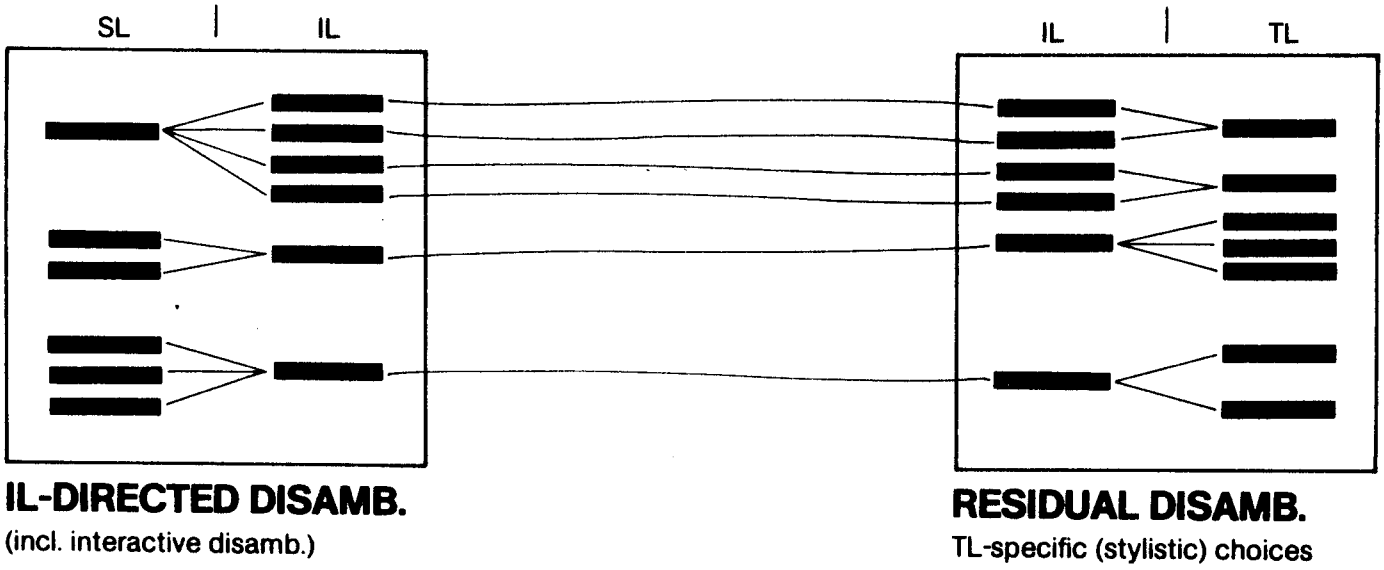


Fig. III-6c. Category-3 systems (DLT): the IL determines the degree of disambiguation at the SL-stretch, structurally as well as lexically.

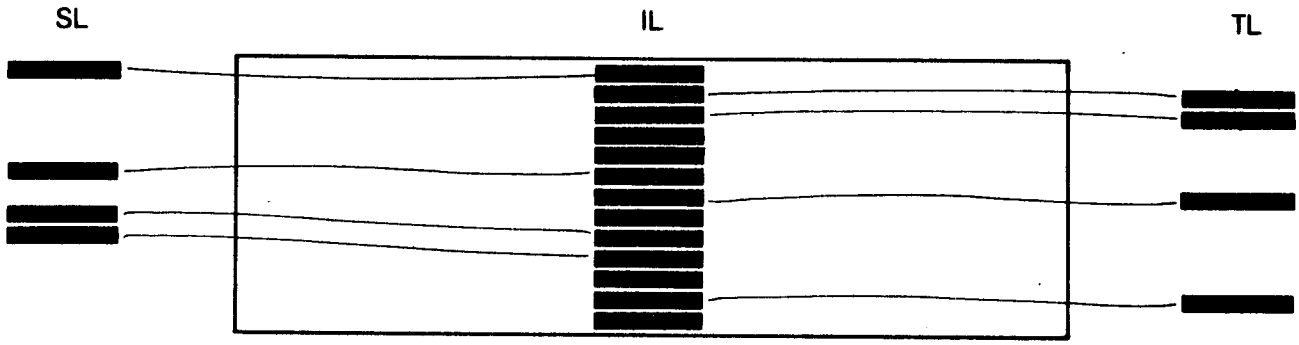


Fig. III-7a. Explosive IL (only 1-1 relations to SL and TL elements).

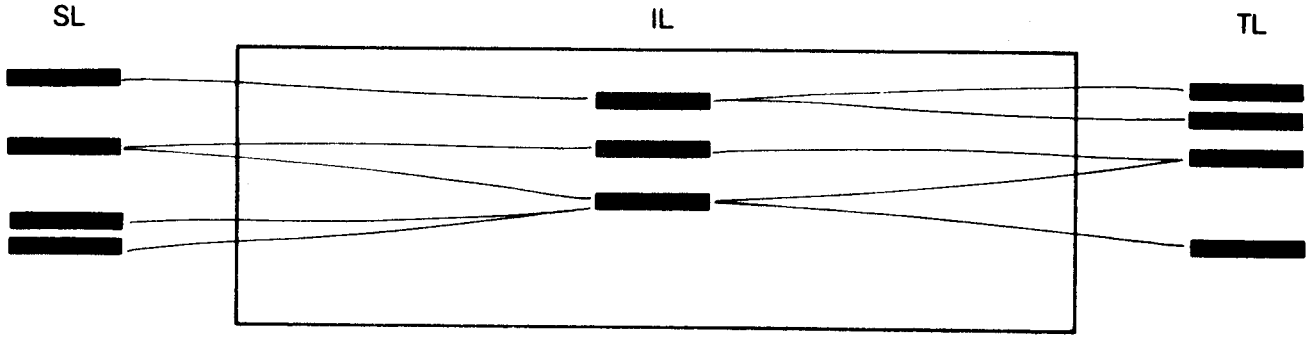


Fig. III-7b. Divergence and convergence with a non-explosive IL.

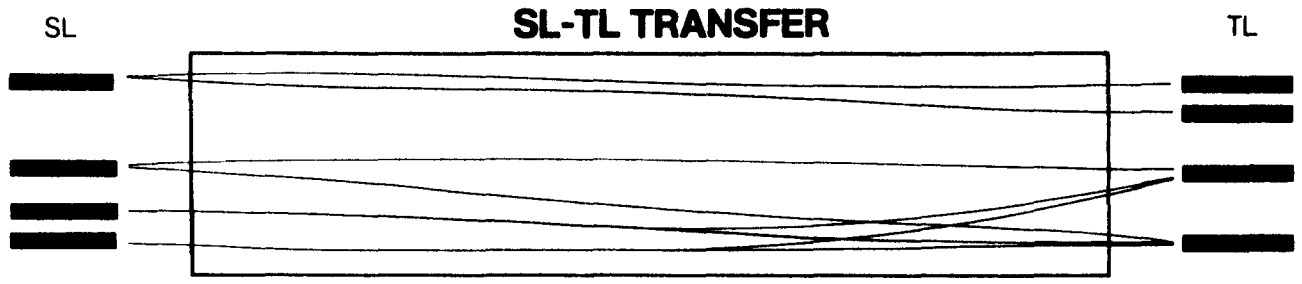


Fig. III-7c. Divergence and convergence without an IL.

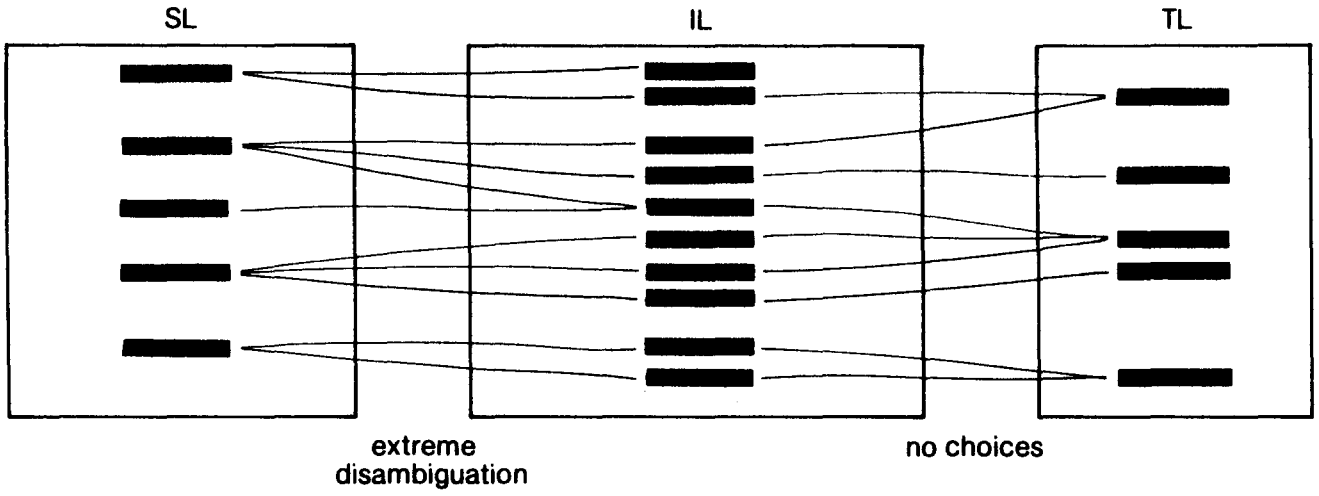


Fig. III-8a. Total avoidance of IL-TL divergence by extreme sophistication of the IL.

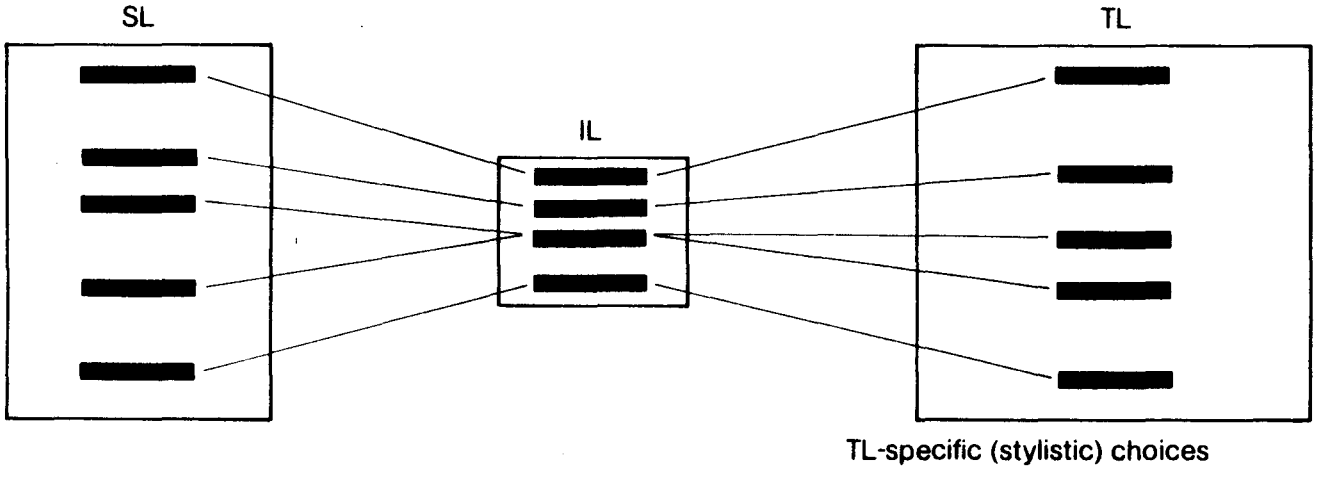


Fig. III-8b. The DLT-philosophy: accepting IL-TL divergence by sophistication of the TL-module.

An 'explosive' IL serving n SL's and m TL's would contain n+m different equivalent representations, each of them reflecting the grammatical structure (distinctions and categories) of the corresponding SL or TL [see fig. III-7a]. In a more balanced strategy, the IL would contain a limited set of grammatical distinctions and categories, common to the surrounding SL's and TL's [fig. III-7b].

The theoretical possibility of an 'explosive' IL not only exists for the grammatical, but also for the lexical translation component. Every language has certain notions which do not exist or are less refined in other languages: Arab has about 20 words for "horse", whereas Dutch has a lot of different words for "canal" ; Turkish and Russian have a more refined kinship terms, etc. Unless the IL is 'explosive', there will always be lexical conversion difficulties, which can be illustrated schematically by the example:

IL:	rivero	French:	fleuve
(English:	river)		rivière

where the IL follows the English lexical pattern. If French, which is more refined on this particular lexical item, acts as TL, the lexical divergence presents a case of residual ambiguity.

Altogether, a basic or non-explosive IL with a certain degree of convergence-divergence at the SL-IL and IL-TL transitions [fig. III-7b] appears to be preferable over an explosive IL. Opponents of the interlingual approach should keep in mind that a similar amount of convergence-divergence is inherent to the SL-TL interface of a transfer system [fig. III-7c].

3.3.3.3. DLT's work-split strategy.

The structural-lexical IL in the middle of the DLT translation process separates SL-side and TL-side disambiguation well from each other, leaving SL- and TL-teams only to be concerned with their own language (and, of course, its relation to the IL). The polarization of ambiguity handling, which is already largely visible in the category-2 systems [fig. III-6b], has been further pursued here.

Another viewpoint - of primary importance in the design of DLT - is expressed in figure III-8. It involves the sophistication of the IL vs. the sophistication of the SL- and TL-modules.

One design strategy could be to make the IL more and more refined, in order to prevent any divergence at the IL-TL transition [fig. III-8a]. This would permit a relatively

simple TL-module (no residual disambiguation required), but at a very high price: not only would the IL have to be extended to a level of structural and lexical refinement far away from its Esperanto-based level [as presented in this report, see also Chapter IV], but also would the SL-side be charged with disambiguations that in fact originate from TL-peculiarities (such as 'fleuve/riviere'). In this strategy, TL-specific problems are shifted to an earlier stage in the translation process, the SL-TL transition, thereby overloading the work required (both automatic procedures and human interactions) at the SL-side. This forwarding of TL-specific problems to the SL-side would have to be channeled by an extremely refined and sophisticated IL. In a multilingual system with many IL's, the IL then would tend to grow into explosive proportions, while the system as a whole would become top-heavy (too much work on the SL-side and in the IL-design).

The other design strategy, and the one adopted by DLT, is illustrated by fig. III-8b. The IL is kept to basic proportions, regarding the degree of refinement and sophistication. For sake of the argument, one could characterize the IL here as: a natural language, void of morpho-syntactic irregularities, idiosyncracies and ambiguities. Divergence at the IL-TL transition is, however, by all means accepted, both lexically (e.g. 'fleuve/rivière') and structurally (e.g. verbal tense and aspect). The TL-module has the task to solve all IL-TL divergence, by use of the TL-grammar and an extensive IL-TL lexicon (including valency information, collocations, etc.). In particular, TL word choices will be guided by TL-information (TL-microcontext), instead of being predetermined by artificial IL-refinement.

Summarizing:
The DLT design thus not only preserves the separation of SL- and TL-specific work at their respective sides, but aims at a fair balance of work: sophisticated SL-specific SL-modules, a compact IL, and sophisticated TL-specific TL-modules.
Whereas SL-modules involve semi-automatic SL-IL translations, the TL-modules consist of automatic IL-TL translations. In this multilingual double-translation system, the compact IL is more regular, less ambiguous and roughly similarly refined in comparison with the surrounding languages.

4. DLT's internal architecture and operation.

After the major architectural characteristics of DLT have been shown in comparison with other MT systems, highlighting the main SL-TL interface [III.2] and the handling of ambiguities [III.3], this section will further describe the design and internal working of DLT. After an introductory explanation of the different steps and modules involved in the total translation process, the principles of operation of the SL- and TL-modules will be dealt with separately (the fact that the SL-module description requires much more space than its TL-counterpart, reflects the more innovative character of the former).

4.1. The overall translation process sequence.

In DLT, the translation of a text unit (normally a sentence) from keyboard-entered SL to visually displayed TL is a process which, as we have seen [fig. III-3], consists of two major steps or stages:

I. Analysis of the SL
II. Synthesis of the TL

clearly distributed over sending (text-generating) and receiving (text-displaying) equipment.

For further understanding the essentials of DLT, a more comprehensive break-down of the SL-TL translation process into 6 intermediate steps may be clarifying [see fig. III-9]. The purpose of this is not so much an elaboration of the SL- and TL-modules separately as rather to demarcate a sort of "IL-kernel" in the process trajectory, and to point out the mutual symmetry of the modules with respect to this kernel. The 6 intermediate steps are:

Step 1: SL-analysis proper
Step 2: Tree-ordering
Step 3: Tree-to-string conversion
Step 4: IL-recognizing
Step 5: IL-parsing
Step 6: TL-generation.

By no means do these steps represent equal portions of the translation work (neither in developmental nor in operational terms). Practically the whole load of translation work is inside Steps 1 and 6.

Step 1 covers the semi-automatic analysis (lexical and grammatical) of the SL, including disambiguation (which may involve an interactive dialogue). Step 1 [which is the subject of section 4.2] results in a tree representation of the

SL-sentence. This tree, with IL syntactic categories (FADJ, Ovv, Ag etc. [see IV.1.4]) at its nodes, and IL words at its leaves, is considered an (unordered) IL-tree and serves as input for the next step.

Step 2 and Step 3 together form what we call the IL-coder. They are SL-independent, and represent an estimated 10% of the total analysis effort. Their output is IL proper, a compact linear and readable string.

Step 4 has a special place in the DLT system. Theoretically it is redundant, in practice it will be vital. Its function is to check the IL-stream on grammatical and lexical integrity, and to reject any sentence with errors. Of course, Steps 1 through 3 are supposed to produce correct IL. Step 4 must therefore be seen as designed for system reliability reasons, according to the principle of dual programming [Gilb, 1973]. Besides, it will be a component of primary importance during the first development phase of DLT (the pilot system).

It should be realized that the input to Step 4 will not always consist of output from Steps 1 through 3. At the interface between Step 3 and Step 4, manually generated IL (for test purposes, but also for certain applications [see Chapter 2]) can be offered. Step 4 has to prevent under all circumstances that incorrect IL spreads through a DLT network and enters into receivers. Therefore, Step 4 is an integral part of each DLT sender module, i.e. the SL-analysis module in its widest sense.
Step 4, which of course is entirely SL-independent, is appropriately referred to as 'IL-recognizer'. It is the filter that protects the DLT network.

The main DLT interface, consisting of verified (recognized) IL, separates the intermediate process Steps 1 through 4 at the SL-side from Steps 5 and 6 at the TL-side.

Steps 5 and 6 together make up the TL-synthesis module, in which Step 5 is certainly the minor part of the effort. While Steps 2 and 3 (together) form the IL-coder, Step 5 serves as IL-decoder: it parses the incoming linear IL-string and reconstructs an IL-tree from it, for the convenience of Step 6. So Step 5 is nothing else than an IL-parser. Note that this parser is ensured (by the presence of Step 4) of getting correct IL only. Also, Step 5 is TL-independent.

Step 6 is the generation of a TL-sentence, departing from a (canonically) ordered IL-tree. This process step is fully automatic (unlike Step 1, it does not involve any interaction dialogue). It includes the lexical and grammatical transfer from IL to TL, and the syntactic and morphological synthesis of the latter.

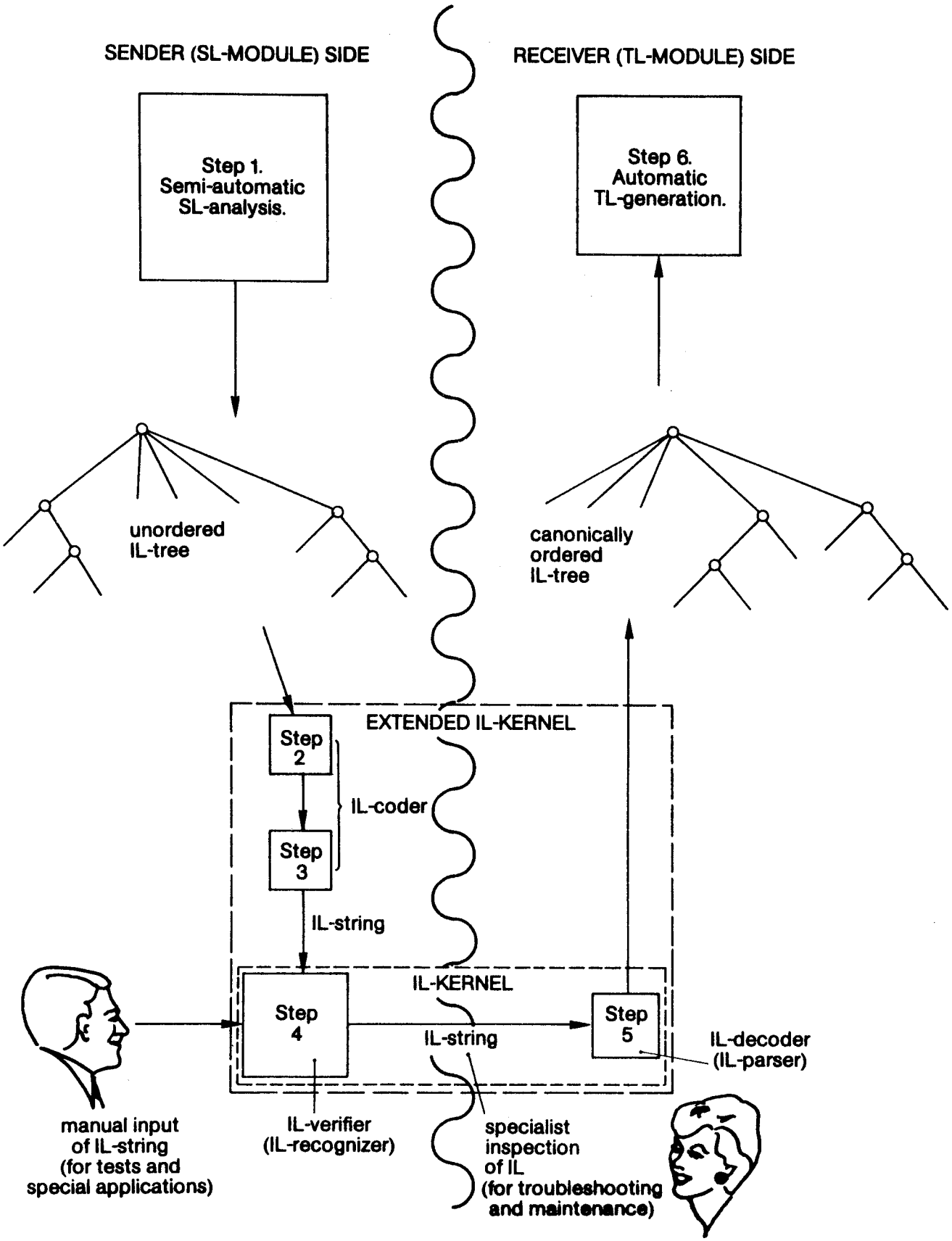


Fig. III-9. DLT process break-down into 6 steps, showing the monolingual (extended) IL-kernel and its various interfaces.

The above process break-down into 6 steps highlights the middle part of the sequence, which is SL- and TL-independent:

Step 1: SL- and IL-dependent (bilingual)
Step 2:)
Step 3:) IL-dependent only
Step 4:) (monolingual)
Step 5:)
Step 6: TL- and IL-dependent (generation)

The distribution of process steps over the 'sending' (text generating) and 'receiving' (text consuming) modules reflects the relatively heavy load on the sender and the modest load on the receiver (as is commensurate with typical information distribution situations [see II]):

SL-module (sender):	TL-module (receiver):
Step 1	Step 5
Step 2	Step 6
Step 3	
Step 4	

For conceptual and organizational reasons we call the combination of Step 4 and Step 5 the 'IL-kernel' of DLT. These steps are important, because they incorporate the IL-grammar to the fullest extent: at one side to recognize (i.e. to formally check) IL, at the other side to parse it (to build tree structures).
If we add Steps 2 and 3 to Steps 4 and 5, we have the 'extended IL-kernel'.

The IL is the cornerstone of DLT's 2-stage architecture; the (extended) IL-kernel is the communication mechanism between SL- and TL-modules, in the sense of network transmission as well as team collaboration.
One could imagine a system without Steps 2,3,4 and 5. The IL-tree resulting from Step 1 would then immediately be passed to Step 6. Although this would preserve the interlingual architecture and valency-based 'double' translation [see III.1.4 and III.2.3 above] of DLT, it would mean giving up the IL's characteristic compactness and readability, attained by a canonically ordered IL with lexical formatives only.

One could regard Steps 2 through 5 as an elaborate but effective 'packing' and 'unpacking' of IL-trees. The 'packed form' (i.e. linear string) is expedient to transmission, quick inspection and fast processing (string-matching) in future intra-IL operations [see II, III.5.2 and III.6].

4.2. Principles of the SL-analysis.

4.2.1. Parsing 'on the fly'.

Step 1, the SL-analysis proper, is dominated by the following design principles, closely related with the operational environment of DLT:

- single-pass, LR (left-to-right) parse;
- intervalwise, piecemeal parse progress, following the rhythm in which the SL-words arrive into the system during the standard text-entry procedure, i.e. from a word processor keyboard (besides, another input mode is feasible [see 4.2.4 and Chapter VI]);
- dedicated hardware: each word-processor console will have its own SL-module boards, containing (multiple) processors and storage [see VI]; there is no such thing as time-sharing.

The rationale underlying these principles is:

- natural language parsing, because of its essentially undeterministic nature, demands a large amount of processing time;
- an average text-entry rate of 2 characters per second, with a maximum rate of 4, has been assumed; this assumption is based on known figures [Martin, 1973: 176] for 'continuous-keying' operators, taking into account that for DLT, typing of SL-text has to be interrupted frequently because of the disambiguation dialogue [see 4.2.4]; also, text-entry by casual typists can be expected more and more in the future [see II];
  the assumed keying rate results in a typical input speed range of 0.5 - 4 seconds per word and 5 - 70 seconds per sentence (this accounts for the variation in word and sentence lengths, depending on the language and the particular type of text);
- 5-70 seconds is an abundance of processor time!
- the mere entering of text, including usual word processor functions (editing facilities), hardly takes up any processor capacity, i.e. the dedicated console processors, unless used for translation, are idle most of the time;
- among a variety of parsing strategies and techniques, methods for single-pass LR parsing do exist.

Efficiency, in terms of computation time, storage space and and amount of machinery (parallel processors) appears to remain a factor of concern even at the present state-of-the-art in natural language parsing [Johnson, 1983; Sampson, 1983].

So what is more evident than running the time-consuming parse process 'on the fly', along with the typing of the SL-text? In this manner, the wide time-slices left by keyboard input will profitably be used by DLT's Step-1 process, the SL-analysis.

Without this 'parallel' operation, all parsing work would accumulate for processing after the entering of a sentence. The outcome of a sentence analysis may be a clarification request to the typist, in the form of a computer-initiated interactive dialogue, and must therefore always be awaited before the next sentence can be entered. We have a potential response-time problem here. According to generally accepted standards [Martin, 1973], response time in situations as routine entering of text should be in the less-than-two-seconds range. Postponing of all parsing work till the end-of-sentence could just cause an unacceptable delay.

The DLT hardware design [see Chapter VI] gives further details on the implementation of parsing 'on the fly' in microprocessor-based desk-top equipment.

4.2.2. The lexical and the syntactical parse level.

The parsing of SL-text as it is entered takes place on two major levels:

I. Lexically: SL-words come in character by character; simultaneity between manual typing and automatic dictionary look-up is largely attained by using the first 3 or 4 characters of a word as pointers of a multi-level index, which exploits the inter-character time-gaps for memory accesses and leaves only small-size buckets to be scanned when the word is completed.

   Matching of typed words with dictionary entries is a complicate process in itself, due to:

   a. The occurrence of input errors, either spelling errors or typing mishits; in order to adequately deal with these (taking into account also 'b' and 'c' below), a sophisticated mechanism for semi-automatic spelling correction will be required.

   b. SL-words not included in the module's dictionary.

c. Special elements in the input text: acronyms, proper names, words not to be translated, number strings etc.

d. Morphological 'blurring': suffixing and inflexion of word tokens, vs. forms in the lexicon (stems).

e. Ambiguities: part-of-speech ambiguity, homonymy and polysemy. Part-of-speech ambiguity requires more than lexical analysis alone. Note that any interactive disambiguation will not be initiated before end-of-sentence, after automatic disambiguation based on syntactical parsing and additional dictionary entry information (valency, idiom, microcontextual procedures) has been attempted.

f. Word compositions: these force to develop 2 or more parallel word parses in many cases. In some pathological cases, more than 1 parse-track will remain after the end-of-word (e.g. the German word "Wachtraum" represents 2 possible compositions: "Wacht-raum" and "Wach-traum" [Brandt Corstius, 1978]), which means another type of ambiguity.

g. Syntactic issues concealed within words. As the whole character string enclosed between spaces is considered a 'word' here, we will be faced with things like:

1. river-crossing equipment
2. a face-saving campaign
3. earthquake-plagued countries
4. chromium-plated steel
5. a science-based organization
6. science-trained people
7. a wall-hanging apparatus

Though the general strategy in DLT will be to include collocations, technical terms etc. in the SL-IL lexicon [see also IV.4.2 and IV.4.5], this will never be possible for all compounds. Therefore, lacking the explicit enumeration of a compound word in the lexicon (in the entry of either of its composing elements, a word-syntactic analysis is required as 'default procedure'.
This analysis can be done on the basis of the valency of the compound's verbal element (the valency information in the verb's lexicon entry). In some cases, the syntactic or semantic role of the non-verbal element can be determined unambiguously, in other cases a consultation of the human operator (in the interactive dialogue) will prove necessary.

(The syntactic or semantic role will be 'recorded' in the form of an IL preposition or accusative; for the above examples: 1 and 2: accusative; 3: 'far'; 4: 'per'; 5: 'sur'; 6: 'pri'; 7: 'sur' .)
This arrangement enables correct translations like:

3. von Erdbeben heimgesuchte Länder
7. appareil suspendu à un mur

II. Syntactically: upon occurrence of a 'word-interval', i.e. immediately after an SL-word has completely been entered (noticeable by the occurrence of a space or a punctuation mark), the syntactic parse level is activated. Making use of the set of SL grammar rules (present as ATN or in some other formalized form, to be discussed in 4.2.3), the syntactic parse results developed at previous word-intervals, and information in the dictionary entries of the current and previous input words, another step is made in the syntactic parse of the sentence, just as far as corresponds with the new information obtained with the current input word (speaking in ATN terms, such a step could consist in the 'consumption' of an input symbol, but also in using it for 'look-ahead' purposes).
The syntactic parse step activated at a word-interval can overlap with the duration of keyboard-input of the next word. The time available for a syntactic parse step therefore equals the typing time for a word: typically 1-4 seconds. From this amount of time, one should deduct an allowance for garbage removal (e.g. 0.05 seconds [Charniak, 1983]) and the time required by the lexical parser, if both parsers share one processor.
The syntactic parsing will be further discussed in 4.2.3.

Of the lexical and the syntactical parser, the former triggers the retrieval of SL dictionary entries, and makes them available to the syntactical parser for the whole duration of the text unit (sentence) processing. Only the syntactical parser makes use of the SL grammar (the lexical parser may be equipped with a small separate grammar for analyzing the structure of compound words not found in the lexicon). The syntactical parser is activated at the occurrence of a space or punctuation mark, i.e. after each orthographic word [see also fig. III-10].

For the rest, there is no sharp boundary between the lexical and the syntactical parser. In fact, they have a lot in common:

a. both proceed intervalwise, in the rhythm of text input;

b. both make use of the SL lexicon; the recognition of multi-word units (English compound strings: technical terms, collocations) is part of the syntactical parser's work (the lexical parser's scope of control is limited to the character string between two blanks);

c. both are faced with the occurrence of alternative parses, which require the development of parallel parse traces (as preferred to backtracking);

d. both may find unresolved ambiguities at the end of a word (lexical parser) or sentence (syntactical parser), in the form of survived parallel parse traces; for the lexical parser, this is the ambiguity of word composition mentioned under I.f above;
these ambiguities may require treatment by an interactive dialogue at the end of the sentence;

e. an advanced spelling correction facility will make use of syntactical (subject-verb agreement etc.) as well as lexical information (moreover, it will take into account keyboard layout, SL digraph frequency etc. in order to guess the right substitute for a mis-hit);
some spelling errors may raise questions in an interactive dialogue, on the syntactical as well as the lexical level;

f. both parsing levels must keep their intervalwise results on an equally intervalwise 'throw-away' basis, in view of always possible character or word deletions by the person entering the text at keyboard; this flexibility requirement (connected with the usual editing facilities such as cursor backmoves) is the price to be paid for the simultaneity of input and parsing.

It has been pointed out [Lawson, 1982] that - in practical experience with MT systems - a high percentage of translation errors could be traced back to errors in the input text. This underlines the importance of integrating error-screening facilities into the design of the SL-module, as indicated above.
To serve as a user-friendly system, commensurate with its generally innovative and long-term goal-setting, a DLT SL-module should not merely protect itself against intrusion of input errors, but also try to correct the error without bothering the user (the person who enters the SL text at the keyboard).
For many errors, involving just one missed key-hit or slight grammatical issues (use of the comma, subject-verb agreement in some SL's, etc.), automatic correction is feasible. For others, the person at the keyboard must be consulted or is

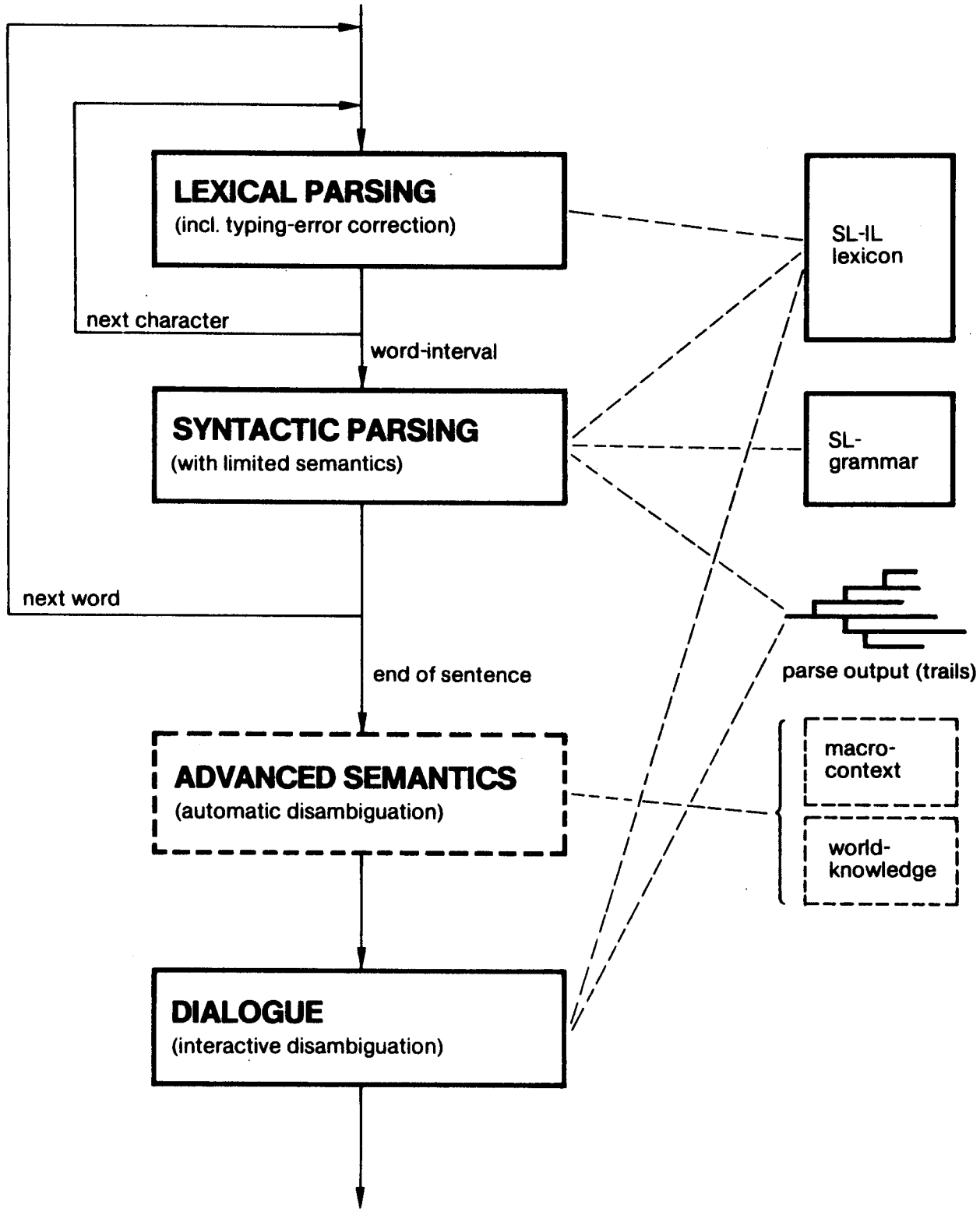


Fig. III-10. Overview of the main flow of control (solid arrows) and the most important relations (dashed lines) inside Step 1 (SL-analysis). Note the nested loops for the lexical and syntactical parse levels. The dashed boxes indicate future extension.

asked to check the computer's correction. Important is that the parsers have a certain built-in ability to cope with wrongly typed words and with sentences that are grammatically not quite correct (a syntactic parser of this type has been referred to as 'semi-grammatical' [Charniak, 1983]).

In the total framework of the DLT design, the projected SL-module capability of a tolerant and flexible response to errors in the SL input text, is completely in line with the strategy adopted for SL-ambiguity handling. The system first attempts to sort out both problems (input errors and ambiguity) automatically, and only after that attempt has failed, resorts to human assistance via a computer-initiated interactive dialogue [see also fig. III-10, which gives an overview of the total Step-1 process].
The tolerance towards SL-errors should NOT be confused with the strict filtering of IL-sentences in Step 4 [see 4.1 above], where any IL-error will cause an immediate rejection of the sentence. Therefore, in contrast to an SL-parser, the IL-parser used in Step 5 (in the receiver) needs to process correct input only.

4.2.3. Parse strategy, grammar model and data structure.

4.2.3.1. General considerations.

This section will deal with the heart of the SL-module, the technical process of syntactical parsing. Though presented earlier as just a component of Step 1 in a 6-step total translation sequence, the syntactical parsing of a natural language covers most of the work and difficulties that have made MT such a hard subject to tackle.
In setting out the contours of the parsing technique envisaged for DLT's SL-modules, the following practical considerations play a role:

- SL-modules will not be built in a first realization phase or as part of the pilot project scheduled after the current feasibility study; this is because of general concensus of the fact that, in a system as DLT, SL-modules represent a higher degree of difficulty than TL-modules; section VII of this report recommends a separate study of SL-modules before the first of them will be developed in a trial system;

- as will be indicated in section VII, French is proposed as the first SL to be part of a DLT trial system; nevertheless, the examples below, illustrating the fundamentals of DLT's SL-analysis, show English as SL; because English excells in part-of-speech ambiguity

(nouns that can also be verbs, postmodifying participles that can also be past tenses, etc.), it could be more or less regarded as a 'worst case' for the study of SL-analysis;

- the area of natural language parsing, partly considered a domain of linguistics, partly of AI, has been in continuous development since the 1960's, and will probably continue to do so; not only will new techniques be invented, but techniques that are considered new and insufficiently tried out NOW, may become just acceptable for practical purposes after a few years ("...one would not want to use Marcus' theory to design an artificial parsing system for some practical purpose, such as translating, ...at least not until this theory has been tested over a long period..." [Sampson, 1983]).

As design considerations that especially determine the choice of an SL parsing technique, we can mention the following constraints and degrees of freedom:

1. Single-pass LR operation, with intervalwise parse progress.
2. Plenty of computing time (per word-interval) and storage space available.
3. Error-prone and ungrammatical input to be handled.
4. Capability to signal and produce alternative parses (in case of ambiguities).
5. No 'deep' semantics capability required during the parsing.
6. Full IL-directedness with respect to the parse output.
7. Long-term easy maintenance and extendibility.

Of these - mainly operational - criteria, number 4 and 5 relate to the presence of an interactive disambiguation dialogue after the input of a complete SL sentence. This semi-automatic design feature [which will be separately described in 4.2.4], not only determines the outside appearance of DLT's SL-analysis, but also dominates the internal working and choice of the parsing method.
In particular, the possibility of recourse to human assistance will significantly relaxe the need for advanced semantic capabilities of the ('syntactical') parser, capabilities that would largely exceed the state-of-the-art of general purpose natural

language parsers, not only now but also for the next decades. It should be noted however, that limited semantic capabilities (via lexical subcategorization and extensive lexicon-contained collocation and valency information, presumably on a preference semantics basis) remain essential to alleviate the interaction load and keep the dialogue within manageable bounds. As mentioned above [see also fig. III-10], an automatic disambiguation attempt will usually precede a request for human interaction.

The required SL-parser can therefore be characterized as 'a syntactical parser with limited semantic capabilities'. Further details follow later in this section.

Readers are reminded not to confuse the parsing of SL, the subject of this section, with the parsing of IL (Step 5 in the overall translation sequence), which is treated in section IV.3. Though some of the design criteria coincide, different constraints play a role in selecting an IL-parsing technique.

Design criterion 6, the IL-directedness, concerns the required output data structure of Step 1 [see 4.1 and fig. III-9], which is an (unordered) IL-tree. This criterion has an overwhelming influence on the SL-analysis, in particular on the lexicon use [see 2.3] and organization, but also on the choice of SL-grammar model, method of disambiguation [as explained in 3.3.3] and data structure built during parsing.

4.2.3.2. IL-directedness.

In fact, what we call 'SL-analysis' is a complete translation process of its own, with the IL as target. To be more precise: Step 1's target (the interface with Step 2) will be one tree, with IL-words at its leaves and IL syntactical categories at its non-terminal nodes. The ordering of this tree will still reflect the order of the original SL-constituents, but apart from this the tree is free of SL-peculiarities and can be further processed by Steps 2 through 5 (the extended IL-kernel) without specific SL-knowledge. Therefore we can call it an 'IL-tree'.
Figs. III-11 and III-12 give examples of such IL-trees, in which - for explanatory reasons - the terminal nodes have been labelled with the original words from the SL-sentence, showing the IL-equivalents only there where the IL is unable to follow syntactic SL-constructions and therefore develops a different structure, such as an IL adverb instead of an SL modal verb. The syntactical category labels throughout these trees are IL-specific, and will be summed up systematically in section IV.1.4. By virtue of the very aim and 'central position' of the IL, these categories largely coincide with most of the syntactic and syntagmatic categories used in general linguistics and notably for languages like English, French etc. This

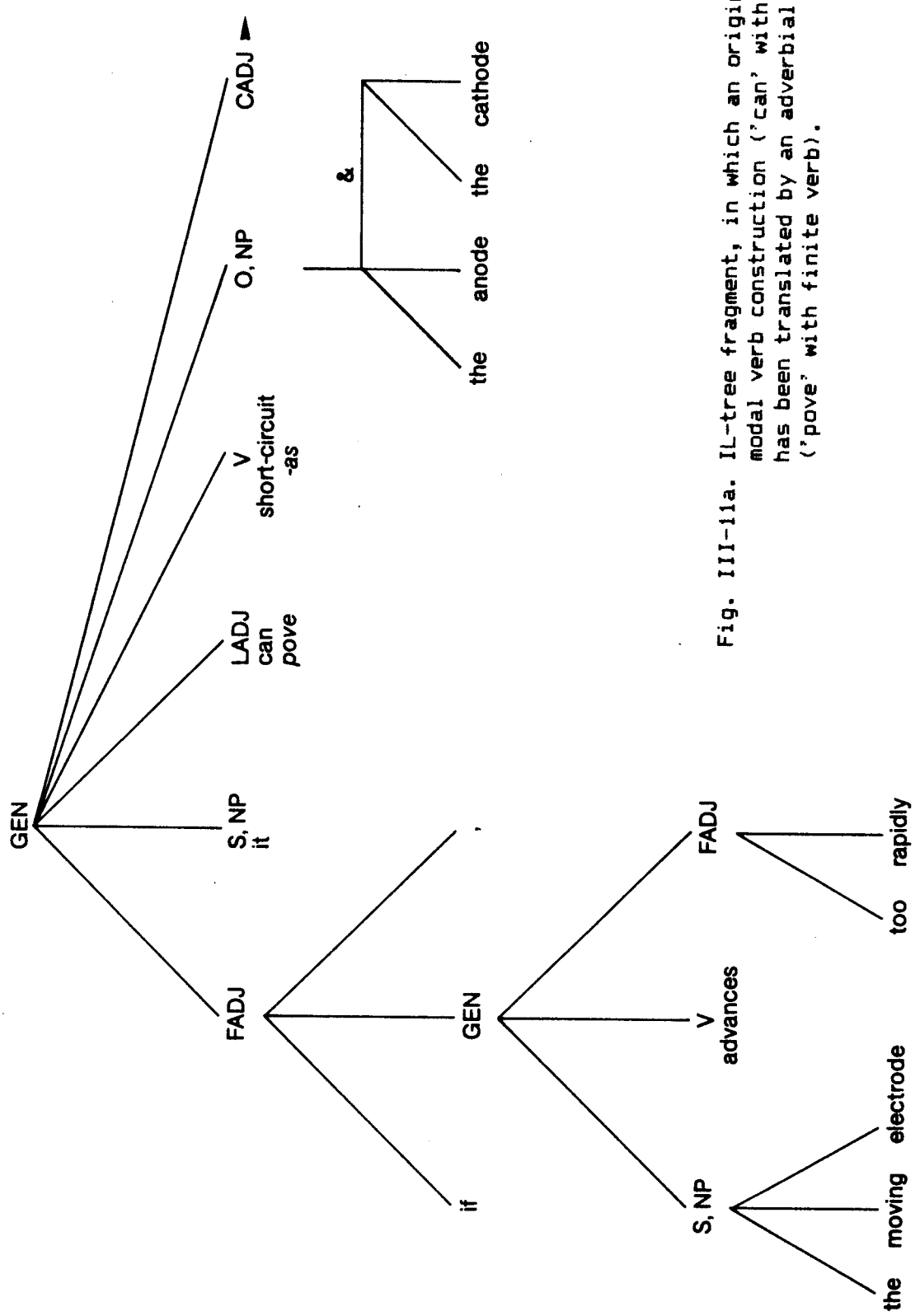


Fig. III-11a. IL-tree fragment, in which an original (SL) modal verb construction ('can' with infinitive) has been translated by an adverbial IL construct ('pove' with finite verb).

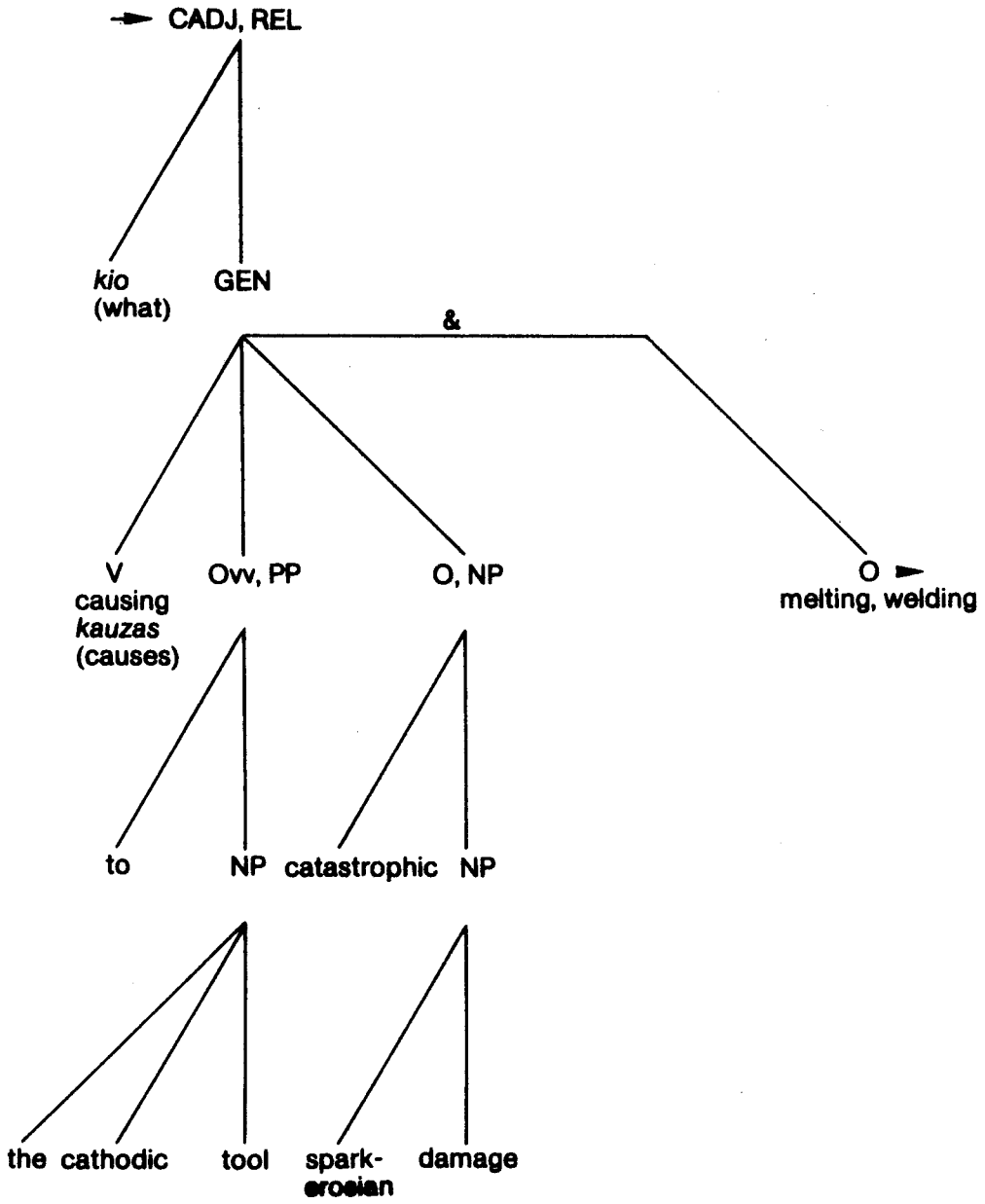


Fig. III-11b. IL-tree fragment, in which an English (SL) '-ing' form has been translated by a pronoun ('kio') and a finite verb ('kauzas'), to form a sentential relative clause (with 11a as antecedent).

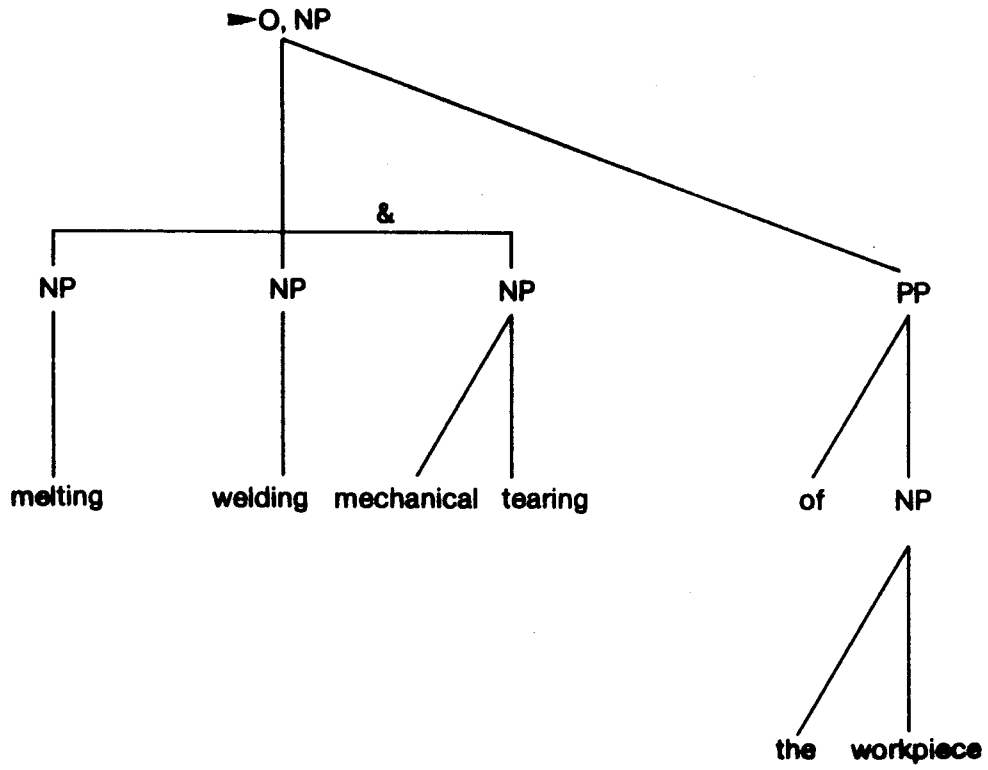


Fig. III-11c. IL-tree fragment (detailing an object of 11b).

Fig. III-11. Impression of IL-tree (labelled with original SL-words) of the SL-sentence: "If the moving electrode advances too rapidly, it can short-circuit the anode and the cathode, causing catastrophic spark-erosion damage to the cathodic tool and melting, welding and mechanical tearing of the workpiece."
The upper case symbols in the tree refer to IL syntactical categories [see IV.1.4]. For clearity, the IL-translation (in italics) is only shown for elements that represent structural differences with the SL.

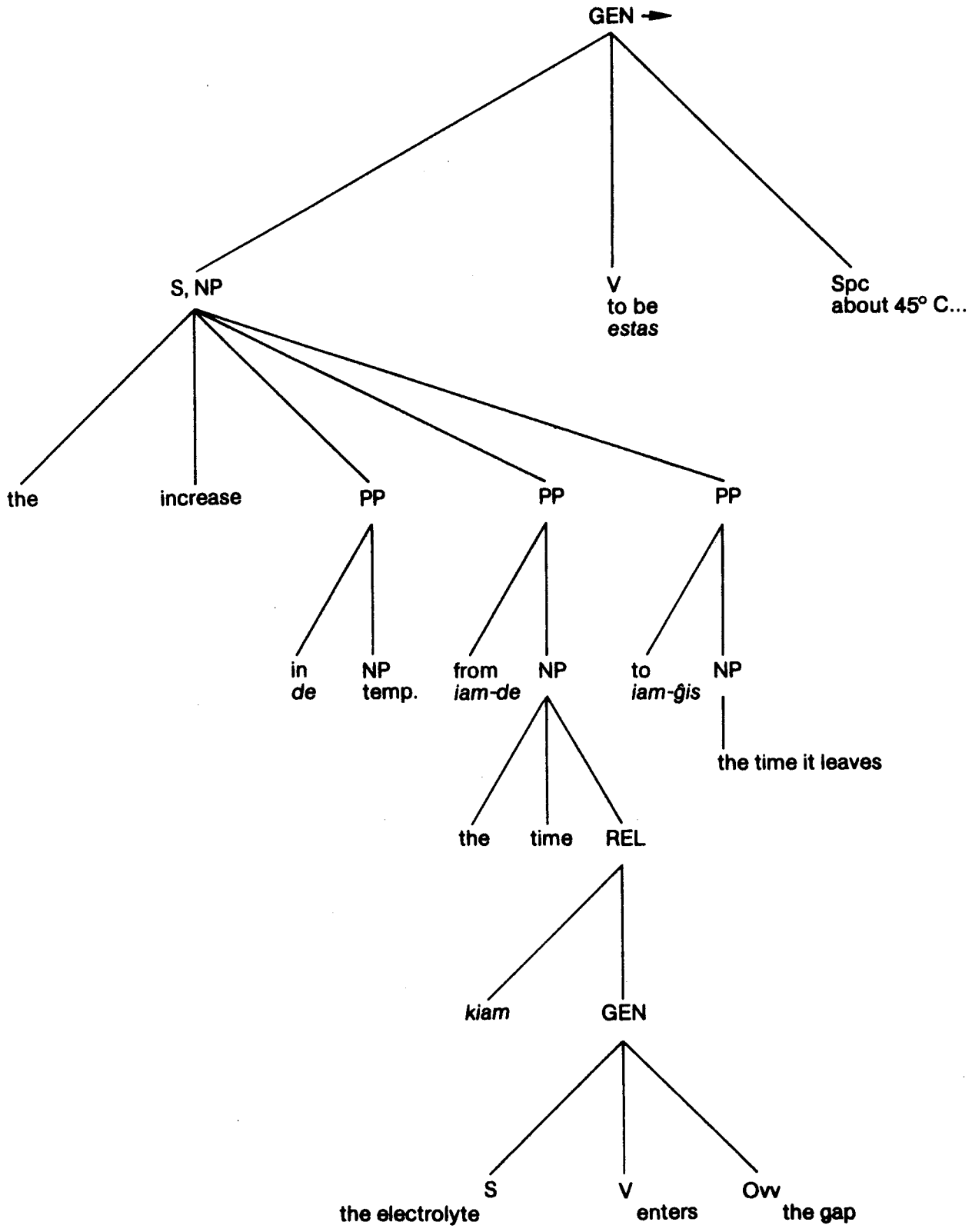


Fig. III-12a. IL-tree fragment, corresponding with an IL general clause. Among other things, it indicates a special (disambiguous) IL-translation of time-related prepositions ('iam-de', 'ĝis-de') and the insertion of the time-related IL-correlative 'kiam' ('the time when').

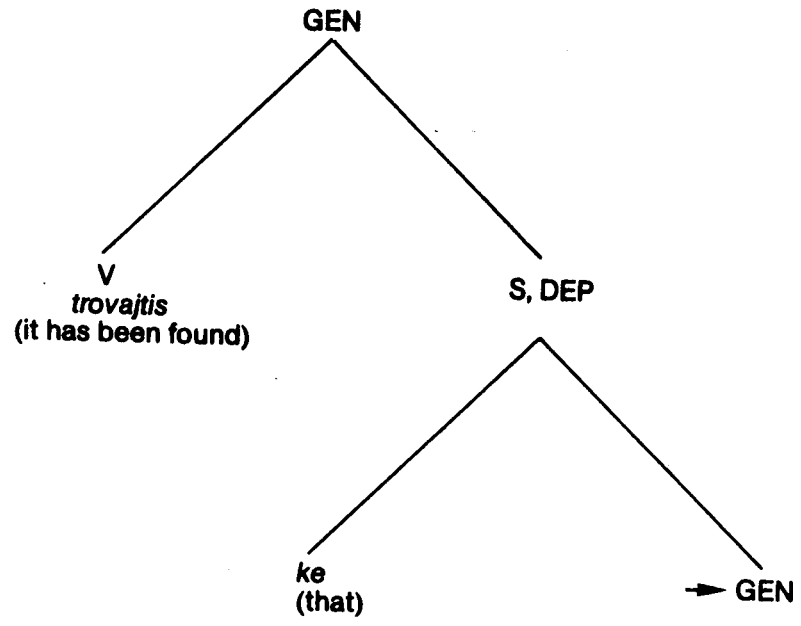


Fig. III-12b. IL-tree fragment, corresponding with the superordinate clause of 12a. Notice the compact synthetic passive form 'trovajtis', which includes the SL anticipatory subject 'it'.

Fig. III-12. Impression of IL-tree (labelled with original SL words) of the SL-sentence: "The increase in temperature from the time the electrolyte enters the gap to the time it leaves has been found to be about 45 degrees Celsius...".
The upper case symbols in the tree refer to IL syntactical categories [see IV.1.4]. For clearity, the IL-translation (in italics) is only shown for elements that represent structural or remarkable differences with the SL.
The apparent structural conversion here is the replacement of the complex predicate '...has found to be...' by an introductory superordinate clause ('trovajtis ke') and a simple predicate ('...estas...'). During the parsing of the original SL-sentence, 12a will have largely been built when 12b is superimposed on the top of it.

indicates a relative advantage for these languages, compared with more 'exotic' SL's, which may require a larger number of structural conversions from unknown to known IL constructions (e.g. Turkish, with asyndetic coordination and subordination).

Having seen Step 1's target, how do we arrive there? Of course, the SL-analysis (which we called Step 1 from a more global point of view) is a sophisticated and multi-step process, the description of which will make up the body of this report section.

First of all, one fallacy must be warned for: it is NOT the intention to split Step 1 in such a way that an IL-independent SL-analysis part is followed by a SL-independent IL-synthesis part (whether or not these parts would be connected by SL-IL 'transfer' in the middle). It is NOT the intention to embed another 2-stage interlingual or even a 3-stage transfer system within a DLT analysis module.
It is true that the internal working of DLT linguistically relies on double translation [1.4], but from a systems architectural point of view the process covered by the SL-module is only a process-half and has not the status of SL-TL translation in other MT systems. The difference is in modularity and development, which justify transfer and interlingual overall architectures for systems with many SL-TL pairs, but not for one single SL-TL pair in isolation. In case of the latter, the so-called direct translation architecture (taking advantage of any coincidential similarity between the languages) becomes attractive.
In DLT, each SL requires only one SL-IL pair, so there is clearly no economy-of-scale motive for further separation of the SL-dependent part. One could think that, on the other hand, separation of an 'IL-synthesis' part, common to all SL-IL modules, would pay. To a certain degree this is the case and has been accounted for in the design, by placing certain tasks (linearly ordering of constituents, effectuation of some agreements) outside Step 1. But one should also consider that the IL lacks the morphological ballast and idiosyncrasies of real TL's, and that it has for instance no complex tense system [see IV.2.2.1]. These factors certainly contribute to keep the IL-synthesis effort small, too small in fact to justify an additional interface stage for it somewhere inside the Step-1 process.
The near-absence of separate IL-synthesis work in the translation process well reflects the role of the IL as an instrument, which is not supposed to introduce a lot of extra work.

So in designing the SL-IL translation process of DLT, no attempt is made to separate an SL-stage from a subsequent IL-stage. Instead, this process is entirely organized as a 'direct translation', in which the SL- and IL-specific elements are interleaved.

4.2.3.3. Example of a parse.

Let us illustrate the SL-IL translation with the processing of an English sentence, appearing intervalwise, one-word-at-a-time, to the syntactical parser (we will number the words of the input):

1. share

The parser will inspect this word's lexicon entry (retrieved by the lexical parse level), and derive the following conclusions from it:

i) it can be a finite form (imperative) of a transitive verb which also allows a prepositional object with the preposition 'with';
the equivalent IL-verb is 'dividi', correspondingly transitive and with the preposition 'kun';

ii) it can be a noun, serving as premodifier in a compound noun string; the lexical entry contains explicit IL-translations for the following technical terms:

share option	= premiuma negoco
share premium account	= agiorezerva konto
share prices	= akciaj prezoj
share register	= akciula registro

moreover, the parser disposes of a rule, how to translate compound strings which are not in the lexicon;
(the parser excludes the possibility of a head-noun, because the word is marked 'countable' and can therefore not occur without determiner).

The parser will start developing two separate data structures, corresponding to the above two possibilities. For the moment, we will refer to such alternative pieces of parse output as 'trails'. The idea is that they are being extended further at each interval, until they prove to be a dead end. If more than one trail survives the entire sentence parse, we obviously have a (total) ambiguity.
Of course, this is nothing else than the prevention of backtracking by parallelism.
But let us continue our parse with the next input word:

2. prices

The parser does not depart from square one now: 2 trails have already been started (according to 'i' and 'ii' above), and the new input may serve both as information to continue or to

reject any of these trails. If the new input presents new alternatives, both compatible with an already existing trail, this trail will fork into alternative trails.

trail i: the imperative finite verb form of interval 1 expects a direct object, for which a noun phrase is a likely constituent; at any rate, it cannot be followed by another finite verb form;
therefore, from the possibilities offered by the lexical entry 'price':

- head-noun = prezo

- premodifier for technical terms:
  price controls = prezregado
  price ring = prezokartelo
  price stop = prezofrostigo

- transitive verb = meti prezon por

the parser selects the (plural) head-noun interpretation (the premodifier option is discarded because of the plural ending);
we could symbolically indicate the current trail contents now by ==V====O,NP== (finite Verb, followed by direct Object, which is a Noun Phrase); the IL-translation fragment developed until this point reads: 'dividu prezojn' (the '-u' of 'dividu' is an imperative verb ending, the '-jn' suffix after the noun 'prezo' denotes the accusative plural);

trail ii: the premodifier interpretation of interval 1 expects a noun to form a compound noun string with; the second input word 'prices' exactly fits one of the technical terms in the first word's lexicon entry;
if the second input word would have been 'figures' (or any other word not cited in the lexicon entry of 'share'), this word's lexicon entry would have been inspected to check whether it could occur as a noun, and thereby produce a compound noun string with its predecessor;
if, for instance, the second input word would have been 'the', the compound noun string possibility and with it trail ii would have been abandoned;
trail ii now contains ==S,NP== (a presumed Subject, which is a Noun Phrase); the IL-translation up to this point reads: 'akciaj

prezoj' (the '-j' is a plural ending).

After this parse step, both trails come out on an NP of which the head-noun has been processed. However, the continuation of an NP after the head-noun (prepositional phrase, relative clause, participle etc. as postmodifiers) is by all means possible in English. This has to be taken into account when the next input word appears:

3. dropped

From the lexicon entry 'drop', the parser derives that 'dropped' is the past tense or the past participle of either a transitive (IL: 'faligi') or intransitive (IL: 'fali') verb.

trail i: the present trail fragment ==V====0,NP== inhibits another finite verb at the main clause level, so the past tense interpretation is excluded;
the participle interpretation is feasible for the transitive sense of 'drop' (the fragment 'share prices dropped' should then be read like the command 'retrieve birds shot'); the IL-translation will now be: 'dividu prezojn faligitajn';

trail ii: the present trail fragment ==S,NP== certainly permits the occurence of a finite verb as main clause constituent; the transitive interpretation of 'drop' is rejected here because the lexicon entry requires it to have a subject (agent) from the category "Animate" or "Intellectual" (human institutions etc.); so only the intransitive verb is possible, yielding: ==S,NP====V==, with the IL-fragment: 'akciaj prezoj falis' (past tense);

trail iii: but also in case of trail ii, NP continuation with a participial postmodifier is possible; this causes a fork and the start of a third trail, which reads: ==S,NP== and 'akciaj prezoj faligitaj'.

The verbal lexicon entry for 'drop' and its IL-equivalents also gives clues as to what may follow. In case of trail ii ('falis'), a direct object is impossible. The adjectival (postmodifying) participles of trail i and iii ('faligitaj[n]') may be followed by an agent and free adjuncts, as parts of a restrictive participial clause (e.g. 'share prices dropped by the institutional investors at the beginning of this year...'), functioning as a postmodifier within the current NP.
The next input word:

4. last

is a nice example of part-of-speech ambiguity. Its lexicon entry shows four sub-entries: adverb, adjective, verb and noun. The adjectival sub-entry shows the following collocations, marked as 'free adjuncts of (past) time':

last time	lastafoje
last night	lastavespere
last week	lastasemajne
last month	lastamonate
last year	lastajare

(the IL-equivalents have adverbial endings). Further, a rule is given how to form the equivalents of similar expressions that have not separately been listed ('last friday', 'last season' etc.). The adjectival sub-entry also states how the above collocations depend upon the absence of a preceding determiner, e.g. in 'his last year in office' the basic adjective translation ('hia lasta jaro en ofico') must be used. The verbal sub-entry indicates an intransitive verb (IL-equivalent 'dauri') with an optional prepositional object (preposition 'for', IL-equivalent 'dum'). The noun-subentry (a shoemaker's instrument, 'ŝuformilo') will have the indication 'countable' and probably a low-frequency mark.

trail i: a verb form, whether finite or infinite, and a noun (NP), whether in a subject or in direct object role, are both syntactically impossible additions to the already existing trail, on the partipial clause as well as on the main clause level;
an adjective (introducing free adjunct) seems, at first sight, a syntactically possible addition on both levels; however, the basic semantics capacity with which the parser is equipped will cause the rejection of a command directed to the past ('*come home yesterday!'); therefore, the only adjective interpretation with which trail i can be continued is a free adjunct as part of a participial clause (cf. 'retrieve birds shot yesterday!'), inside the NP; the main clause still reads: ==V====O,NP==, and the extension of the IL-fragment is postponed in anticipation of a collocation;
as a simple adverb, 'last' can be added onto the main as well as the participial clause level, causing initiation of the following new trails (note that the simple adverb too functions as a free adjunct):

trail ia: ==V====O,NP====FADJ== (cf. 'remove clamps first!'), with the IL-fragment: 'laste dividu prezojn faligitajn';

trail ib: ==V====O,NP== (cf. 'answer calls received first'), with the distinctive IL-fragment: 'dividu prezojn faligitajn laste';

trail ii: again, verb and noun are excluded on syntactical grounds (we are in a main clause, which already has a main verb that does not allow a direct object);
an adjective as part of a free adjunct is possible (and its time reference is not in conflict with the verb's tense), producing: ==S,NP====V====FADJ==, the extension of the IL-fragment awaiting the exact collocation;
in addition, a simple adverb is possible:

trail iia: ==S,NP====V====FADJ== (cf. 'the Americans came first'), with the IL-fragment: 'laste akciaj prezoj falis';

trail iii: a noun cannot be added here either, but a finite verb is possible ('problems postponed last forever'), and its form agrees with the (plural) subject; trail ii then becomes: ==S,NP====V==, with the IL-fragment: 'akciaj prezoj faligitaj dauras';
two other continuations of trail iii are possible, causing a fork and the start of trail iv and trail iva:

trail iv: the addition of an adjective (introducing a free adjunct) as part of a participial clause inside the NP (cf. trail i), leaving the main clause: ==S,NP== ;

trail iva: the addition of a simple adverb, also as part of a participial clause inside the NP, leaving: ==S,NP== with the IL-translation: 'akciaj prezoj faligitaj laste';

assuming here that no free adjuncts of time can occur immediately after the subject in English, this ends the number of alternative parses at the current word-interval.

At the arrival of the next input word:

5. season.

the end-of-sentence appears (because the occurrence of a space or punctuation itself triggers the shift of control from the lexical to the syntactical parser, a full stop is noticed by the latter simultaneously with the reception of the last word). The dictionary entry of this last input word has three sub-entries: a noun (denoting a period of time), a transitive verb and an intransitive verb.

trail i: according to this trail and the dictionary entry inspected at the previous word-interval, 'season' does not fit one of the explicitly listed collocations there, but it fits the rule for producing free adjuncts of time with 'last'; this requires inspection of the dictionary entry of 'season' itself, in order to check that it can be a noun with the semantic category TIME, which appears to be the case;
the resulting imperative sentence has the pattern ==V=====O,NP== and reads 'dividu prezojn faligitajn iam-en la lasta sezono.';

trail ia, trail ib: neither a noun, nor a verb can be added to any of these trails syntactically, so they both are rejected now;

trail ii: as in trail i, a free adjunct of time can be formed here, yielding ==S,NP====V====FADJ== and 'akciaj prezoj falis iam-en la lasta sezono.' ;

trail iia: no noun or verb can be added; trail rejected;

trail iii: no anticipation of a collocation stands out, so the lexicon entry of 'season' is inspected; the main clause developed at this trail leaves no place for an NP, neither as subject, nor as object; an NP functioning as free adjunct would be possible for this word ('problems unsolved last a week'), but only with certain determiners or numerals; this rejects the addition of a noun;
the addition of 'season' as a verb is impossible too, because there is already a verb in the main clause;
this trail is therefore rejected;

trail iv: here, as in trail i and ii, the information in the dictionary entries of 'last' and 'season' allow the formation of a free adjunct of time; this satisfies the structural possibili-

ties of the participial clause within the NP;
however, the structural demands of the main clause are not met: considering that this is the end of the sentence, the main clause remains incomplete (it has no predicate), and thereby ungrammatical;
this trail too must therefore be rejected.

trail iva: a noun cannot be added, but the intransitive verb interpretation of 'season' is possible, and forms a predicate at the main clause level;
the resulting pattern is: ==S,NP====V==, and the IL-translation reads: 'akciaj prezoj faligitaj laste, maturiĝas'.

The English input sentence 'Share prices dropped last season' has thus resulted in 3 alternative parses (trail i, ii and iva), all three recorded with the use of IL. With the completion of the parsing process, we have arrived at the next stage of SL-analysis [see fig. III-10]: an automatic disambiguation attempt, based on advanced semantics, and followed - if necessary - by an interactive dialogue.
In this example, the automatic disambiguator could for instance reject the 'imperative' interpretation (trail i) because a command would not fit in the current context. This still leaves two interpretations of the SL-sentence, which will be submitted for human decision in an interactive dialogue.
After this, only one parse-result (presumably trail ii) remains, which Step 1 hands over to Step 2 in the form of an IL-tree.
Note that we have not considered the possibility of input errors in this example. Also, a certain refinement relating to this parsing example will be mentioned below [in 4.2.3.4c].

4.2.3.4. Process characteristics.

As already indicated, the syntactical parsing process proceeds in discrete steps, corresponding with the appearance of completed orthographic input words (and not to be confused with the major Steps defined in 4.1, of which Step 1 covers the entire SL-analysis). We refer to these steps as 'intervals'.
Beside this, the SL-parsing, as just illustrated in the worked-out example above, shows more important features, which are characteristic for the SL-IL translation process (Step 1):

4.2.3.4a. Immediate insertion of IL-words.

The lexicon linked with the parser is bilingual, SL-IL. Upon retrieval and inspection of a lexicon entry at a given interval, a word's IL-equivalent may immediately be added to the developing parse output. There, the IL-word will serve as a record of the parser's interpretation of the SL-word.
As in the example, if the parser interprets the English word 'share' as an imperative verb form, this is recorded as the IL-word 'dividu' ('-u' being an imperative ending). No separate abstract grammatical formative will need to be inserted into the parse data structure to document this!
Likewise, if the parser in some reading would regard 'last' as a noun, this would be recorded by insertion of the IL-word 'ĝuformilo' (the '-o' ending denoting a noun, the '-il' suffix indicating the semantic category 'instrument'), and not by providing the SL-word with an index.
Whether the English word 'dropped' is understood as a participle or as a past tense, and whether transitive or intransitive, is similarly recorded in IL: 'faligita', 'falis' etc. (with distinctive IL endings and suffixes for adjectivally used participles, perfect tense, past tense, transitive and intransitive verb meaning).

So because of their grammatical transparency and their self-documentary properties (apparent in a perfectly systematic morphology, a powerful mechanism for word meaning differentiation, a particularly neat system of prepositions and syntactic cases, etc. [see Chapter IV]), IL-elements are used in the partial parse output at an early stage of the SL-analysis, where they facilitate the parsing process itself (at subsequent intervals), and where they already form the basis of a target representation, which will (after Step 2 and 3) exclusively depend on IL lexical formatives.
Also, the number of IL-equivalents found in an SL-IL lexicon entry, i.e. the lexical divergence [cf. sections 3.2 and 3.3.3], determines the presence and degree of ambiguity involved with a given SL word. An English word is never considered ambiguous per se, but only because of alternative IL-representations. This is a strong and clear operational criterion: in the example [4.2.3.3], the lexicon entry for 'drop' only contained the transitive/intransitive pair 'fali'/'faligi' as equivalents. If it would contain 'guti'/'gutigi' as well, the question whether the raindrops sense applies would have to be answered. The range of IL-equivalents thus influences the necessary disambiguation procedures, be it SL-directed microcontext-based automatic routines, be it an interactive dialogue which presents the meaning distinctions on the screen in SL.

Two things must be noted about the immediate use of IL-words: Firstly, in the developing parse data structure, the original SL words are retained (i.e. insertion instead of substitution

with IL-words takes place). This is because lexicon entries opened up in later parse intervals may contain the co-occurrence (in the same sentence) of an SL word as a condition, e.g. in:

(16) 'On the building of the nuclear power plant surely
1 2 3 4 5 6 7 8 9

depends the economic future of this country.'
10 11 12 13 14 15 16

the English preposition 'on' will be looked for during parse interval 10, enabling a verb valency SL-IL translation of 'depend on' by 'dependi de'.
Secondly, no claim is made that an equivalent IL-word can be selected and inserted at <u>each</u> parse interval. For some SL-words, the equivalent can only be established at the next or even at a much later parse-interval. In the 'share prices' example, no IL-word for 'share' is inserted into trail ii until the next word, 'prices', has been received. But not only anticipation of collocations holds up IL-word selection, also the occurrence of ambiguous SL function-words will do this. In

(17) 'By the end of this long conference hall, ...'
1 2 3 4 5 6 7 8

the preposition 'by' will not be translated before interval 8, which finally resolves the place/time ambiguity (the passive-agent possibility is already ruled out at interval 3).
Also, the parse strategy will allow that an already inserted IL-word is replaced during a later interval.
The principles underlying these two notes will be further discussed in separate paragraphs below.

Summarizing the issue: IL-fragments are inserted into the partial parse output as the parse proceeds through the SL-sentence. At some parse-intervals this will happen immediately, at others it may be postponed till later intervals. The inserted fragments correspond with SL-IL translation of words, valencies, collocations etc. Divergent translations can be met by alternative parse trails, and once-inserted fragments may sometimes be overwritten again.

4.2.3.4b. Interleaved appliance of selectional restrictions.

All along the parsing of an SL-sentence, beside obvious syntactic restrictions (agreements, valency), semantic selectional restrictions may test the addition of a new input word to an existing parse trail. Depending upon the outcome of such a test, a parse trail may be continued, branched (if multiple

interpretations pass the test) or abandoned as a dead-end.
The semantics involved is basic or 'shallow' semantics, based on selectional subclasses such as: ABSTRACT, HUMAN, INSTITUTION, MATERIAL, MENTAL ACTIVITY, TIME REFERENCE, etc. These subclasses have to be included as word labels throughout the lexicon and must therefore be of fairly general application, covering the wide range of 'formal and informative texts' aimed at by DLT [see Chapter V].
Apart from the need for subclassification of nouns, verbs, adjectives etc., there is no operational difference or sharp division between syntactic and basic semantic restrictions. This justifies to name the parser 'syntactical'. The basic semantics is interleaved with the syntactic parsing, as the selectional restrictions are applied - in principle - at each interval, on partial parse results. In this way, the big value of the semantic tests is that they prevent a parse explosion, i.e. an unnecessary and uncontrolled proliferation of alternative parse trails.
In fact, the syntactic and interleaved semantic restrictions perform a continuous disambiguation during the parsing process. This disambiguation 'on the fly' should be distinguished from the separate disambiguation attempt that may follow the syntactic parsing [see fig. III-10] and which will be discussed in a separate paragraph [4.2.3.4f].
An example of the usefulness of semantic selectional restrictions is the SL-fragment:

'... convince the chairman of the board of the necessity ...'

The lexicon entry of 'convince' will signal the possible presence of a prepositional object with 'of'. Additional indication of the subclass ABSTRACT or 'sentential noun' (nouns that can have an infinitival clause supplement) as preferred valency object will prevent the wrong interpretation:

* (convince) (him) (of the board of the necessity).

On the other hand, the clause:

'... convince the chairman of the board of directors'

will be interpreted correctly by the same mechanism (apart from the possible inclusion of 'board of directors' in the lexicon, as a collocation).

4.2.3.4c. Moderate use of parallelism.

The relative abundance of computing time, distributed over as many parse-intervals as there are words in the SL input sentence, is exploited by developing alternative parse trails in parallel, according to a certain strategy.

This strategy strikes the balance between two extremes: ignoring or postponing all parse alternatives over many intervals, risking a large amount of backtracking at the end of the sentence, the one extreme, and a rapid growth of the number of (parallel) parse trails, the handling (retrieval, testing, continuation) of which again uses up all available processing time (and a lot of memory), the other extreme.
Of course, whatever the strategy, the number of alternative structures produced after completion of the parsing should stay the same, and this is usually a small number. In our example 'share prices dropped last season', 4 out of 5 words show part-of-speech ambiguity: yet only 3 parses remain at the end of the sentence.
What does vary with the strategy is the number of trails when we take a snapshot about half-way the sentence, i.e. not the number of final but the maximum number of temporary, partial parse alternatives recorded. In our 'share prices' example, this number was 8 (for a 5-word sentence). One should consider, however, that English is notorious for part-of-speech ambiguities, and the particular example chosen may even rate above average in this respect (normally, with increase of the number of words in the sentence, articles and other constituents tend to reduce ambiguity).

The 'moderate' parallelism that will be aimed at in DLT's SL-analysis exists in keeping the number of partial parse trails, stored at any time during the processing, below or within the order of n (n being number of words in the sentence). This can be achieved by:

- Exerting selectional restrictions at the parse-intervals, instead of at the end of the sentence. This will not only prevent an undue branching and initiation of new trails, but will also cause intermittent checking and removal of existing parse trails. It acts as a stabilizing factor on the varying number of partial parses (trails) that must be kept track of at each interval.

- Only those ambiguities that cause structural differences are allowed to open up new trails. In the first place, this is part-of-speech ambiguity ('share prices dropped last season', 'control demands change' etc.), but also some other types of ambiguity may be involved.
  Note that any initiation of a new trail of course underlies syntactic compatibility with previous partial parse results, and also the selectional restrictions mentioned above.

- Postponement of the SL-IL translation of a word, or the ability to override it later, will be preferred over the use of separate trails if no (major) structural differences are involved: anticipation of collocations,

certain multiple meanings of function words (e.g. the TIME/PLACE ambiguity of prepositions), polysemy within the same valency pattern, etc. In 'share prices dropped...', a possible choice between 'faligi' and 'glutigi' (cf. German 'senken' and 'tröpfeln') as IL-equivalents for the transitive interpretation of 'drop' would be an example of this (i.e. if this detail is not covered by a semantic selectional restriction in the lexicon entry of 'drop', and needs to be forwarded to the interactive dialogue, it will not cost an extra trail).

- For rare syntactic structures, the initiation of separate parse trails might be given up in favor of a possible backtracking later. This could improve the average parsing efficiency for all sentences. The decision whether a particular type of alternative structure will be accounted for by parallelism or by backtracking, could be taken dynamically during the parsing, depending upon the number of trails already being processed.
  Such a policy would give preference to the 'nonrare' structural interpretation of an SL-sentence, or only raise a 'rare' interpretation (in the disambiguation dialogue) if there are very few alternatives.

In addition to and in connection with the above, an important arrangement must be separately mentioned. It concerns the notion of Structural Impact of Ambiguity (SIA), which we will first explain.
Consider a main clause with the structure

==FADJ====S,NP====V====O,NP==

(a free adjunct, subject, finite verb and direct object), in which both the subject and the object are noun phrases (NP).
Now suppose that the subject NP contains:

'experienced scientists and engineers' ,

then we have ambiguity of scope (NOT to be confused with SIA!) of the premodifier 'experienced' (we do not know whether it also applies to 'engineers'), and we will regard this - quite right - as a 'local' problem inside the NP: it is extremely improbable that structural alternatives on the higher (main clause) level depend on the outcome of this ambiguity, which we therefore are entitled to call a 'local ambiguity'.
But suppose that instead, the NP would read:

'the agent of X and Y ...' ,

again a case of ambiguity of scope, this time of the conjunction 'and'. Now, however, the outcome of this ambiguity determines whether the subject of the main clause is singular or plural,

a factor that may play a role (agreement restrictions) in the selection of structural main clause alternatives. E.g. in:

(18) 'the agent of X and Y question schedules waiting for approval',

there might be uncertainty about the predicate verb; the choice of 'schedules' as such would then be linked with the smaller scope of 'and' in the NP (making it a singular subject). In such a case, the SIA of a seemingly local ambiguity propagates into the main clause. Another example would be the meaning of the polysemous word 'agent', if one of the main clause alternatives underlies a HUMAN subject selection restriction, as part of the main verb's valency.

Apparently, there are many cases in which main clause alternatives are independent of lower level ones, or even totally absent:

(19) 'today, we discussed the agent of X and Y' .

One case concerns our former example 'share prices dropped last season'. Apart from the already mentioned polysemy ('senken'/'tröpfeln'), the postmodifying participle 'dropped' can also be understood in the intransitive sense ('herabfallen'/'triefen', with the IL-participle 'falitaj[n]' instead of 'faligitaj[n]'), which we ignored in our detailed treatment of the parsing of that sentence. The difference only affects the participial clause (which, for instance, might get a passive agent in the transitive but not in the intransitive case). So apart from 8 temporary and 3 final trails to express the main clause parse alternatives, we have to carry along some isolated 'local' alternatives (with the prospect of having to solve these eventually in the interactive dialogue, if interleaved selectional restrictions fail to do so).
As it happens, ANY main clause interpretation having a participial subclause is rejected in the sentence-final disambiguation, as we have seen. Sorting out the local subclause alternatives is therefore not relevant anymore.

By taking into account the SIA-range and accordingly making a distinction between 'local' (as defined here) and other ambiguities, we can avoid an unnecessary combinatorial explosion, such as would occur if all ambiguities would be interrelated. Instead, a concept of 'subtrails' must be thought of, to record local ambiguities in an economic way.
Apart from a substantial moderation of parallelism, such a measure also contributes to an optimal structuring of the ultimate selection procedure, the interactive disambiguation dialogue.

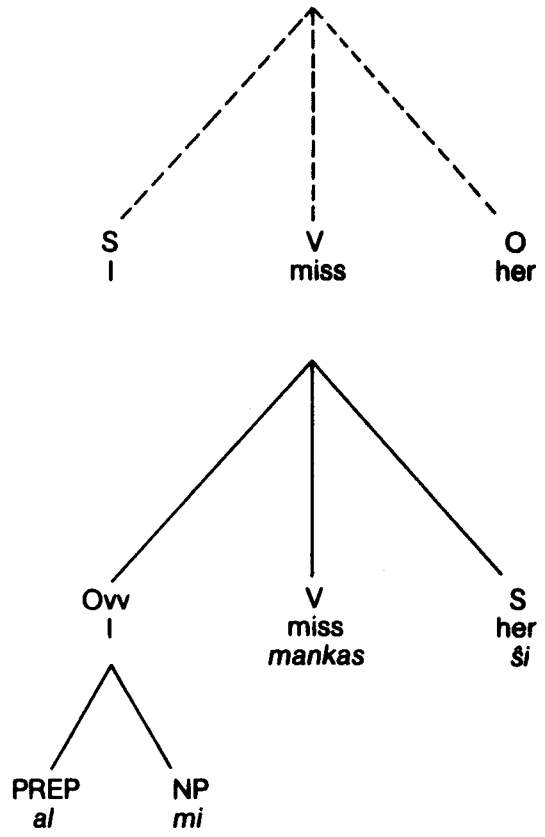
4.2.3.4d. Structural SL-IL transfers.

We have seen, up till now, that the IL-translation develops intermittently, as the SL-sentence is being parsed (from left to right) in a stepwise manner, interval after interval. We have also seen that alternative interpretations may give rise parallel parse trails, and that the insertion of IL-fragments may sometimes be postponed or redone, in other words: a careful balance between moderate parallelity and limited 'backtracking'.
The latter, however, can be more than just a substitution or postponed insertion of an IL lexical formative. It can also change the syntactic function of the involved element, and thereby affect the node labelling or geometry of the developing tree structure (which, essentially, is an IL-tree in state of construction [see also 4.2.3.2]).
In the preceding paragraph on parallelity, we argued that only structural alternatives can justify the initiation of parallel parse trails. This does not mean that all structural alternatives will do so. We can distinguish between 'major' and 'minor' structural changes, the former of which will be handled by parallelity, the latter by backtracking:

Major structural changes:
- a finite verb may turn out to be a noun or adjective, and vice versa ('share prices dropped last', 'government demands change');
- the phrasal vs. clausal antecedent interpretation of certain relative pronouns or gerunds (e.g. 'it can short-circuit the anode and the cathode, causing spark-erosian damage ...' [fig. III-11]);
- the phrasal vs. clausal object meaning of certain verbs (e.g. 'the dog has been found' vs. 'the increase in temperature has been found to be ...' [fig. III-12]).

Minor structural changes:
- a modal verb is replaced by an adverbial (e.g. in 'it can short-circuit' [fig. III-11a]);
- syntactic function exchange among the following clause constituents: subject, direct object, indirect object, subject and object predicative complements and free adjuncts (S, O, Ovv, Spc, Opc, FADJ in terms of IL syntactic categories [IV.1.4]), i.e. all except the main verb); an example is fig. III-13 (to be discussed below);
- addition of certain constituents, e.g. free adjuncts at clause level; we refer here to structural additions with respect to the SL originals, as in fig. III-14.

Of the minor structural changes, some (the syntactic function exchanges between 2 or more constituents) can be characterized as horizontal tree transformations around a stable main verb node (TGG-adepts are reminded that the transformations about



SL-IL lexicon entry:

miss N	fraulino
miss N	manko
miss V	
be ~ ing	
~ school	
~ the train	
	manki →
Ovv = S, S = O	

Fig. III-13. Example of an IL-directed tree transformation, guided by valency information in the SL-IL lexicon (the rule contained in the lower line of the lexicon entry is applied here), and effectuated during SL-analysis.

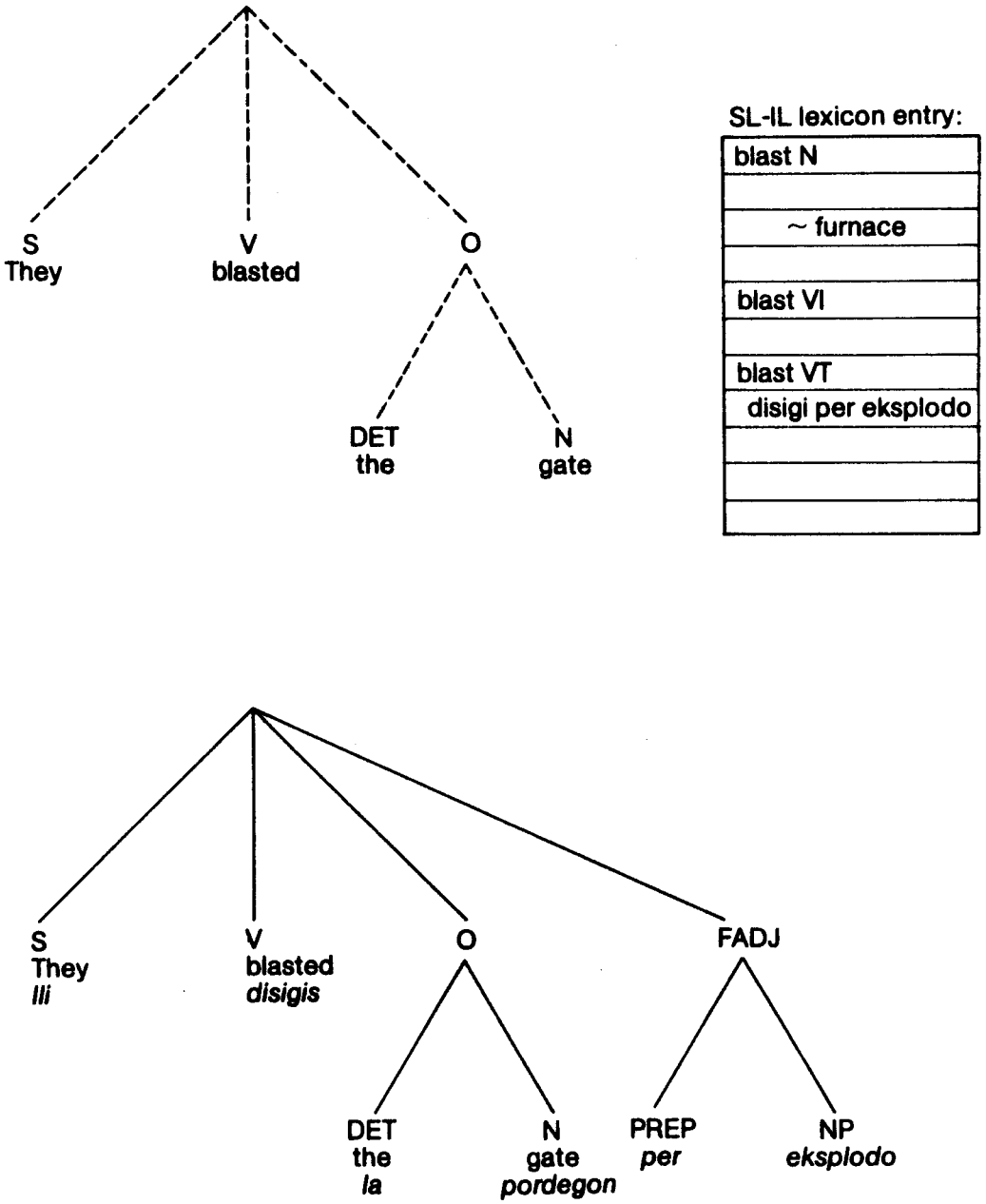


Fig. III-14. Another example of an IL-directed transformation, consisting in a horizontal extension of the tree during SL-analysis. Notice that the tree is still unordered in terms of IL canonical word order [see IV.2.1.2].

which we speak are contrastive SL-IL syntactic conversions!).

Let us look now at fig. III-13, showing the SL input fragment 'I miss her' and the SL-IL lexicon entry of 'miss'. Upon reception of the first input word, a first tree node will be created, marked S (subject), with the terminal 'I' (and its IL-translation 'mi') under it. After the second word of this fragment has been received, the immediate precedence of the pronoun rules out any noun interpretation for it. The verbal translation of 'miss' depends upon its possible use in several collocations, as the lexicon entry suggests, and is therefore postponed. If the next word is 'her', the non-collocational meaning of 'miss' (IL-translation: 'manki') is assumed and the according lexicon-contained transformation procedure is executed: the former subject node S now becomes indirect object node Ovv (verb-valency object), and an IL-preposition is added to the terminal 'mi' there; the new word 'her' (with its IL-translation 'si') will be put under a node marked S.
The resulting tree structure after this input fragment (the solid tree in fig. III-13) emphasizes the IL-directedness of the SL-analysis: the tree developed is an IL-tree, certainly when compared with the structure of the original SL-fragment (the dashed tree in fig. III-13).

We must admit that the example just given has been simplified. In reality, the IL-translation of the pronoun 'her' will be postponed till further input indicates whether it is an autonomous pronoun or a determiner ('I miss her daughter').
A more intricate detail is the following:
The question can be raised what will happen if the next or some later input word would be 'train' ('I miss her train', 'I miss her eldest son's train', etc.). First of all, in the lexicon entry of 'miss', the information on collocations or idioms with the word 'train' must allow such a pattern, i.e. one in which the possessive pronoun does not refer back to the subject: whereas 'she will miss her train' is quite common, 'I will miss her train' might be regarded as rare. Of course, this is up to the opinion of the lexicographer responsible for the particular lexicon entry. Let us suppose that he decides to allow the latter pattern and to mark it as 'rare'. Now if later during the parsing of 'I miss her ...' the word 'train' is received, then the already assigned reading ('manki') must be overridden: a clear example of back-tracking, deliberately chosen for rare cases.
But in order to keep these possibilities open, some mechanism must be available by which the effects of a previous transformation can be turned back. This could either be a record of transformations performed on each trail, or it could be a parallel developed untransformed parse trail: the latter would contribute to the homogeneity of the parse data structure.

Therefore, where a transformation of the SL syntactic structure

is applied, it seems a wise precaution to pursue a 'virginal' trail next to the transformed one. This policy provides backing in case of a necessary rehash, and still it does not require a trail for each contingency separately. The virginal trail can be copied when needed and must be regarded as a general safety provision in a strategy aimed at the best compromise between parallelity and backtracking.
Within a clause, the main verb will function as a transformation's 'center of control' and 'center of rotation'. Though further study will be needed, it is felt that one transformation will suffice to translate a clause into IL (many SL-clauses will require no transformation at all). Also, SL-IL transformations in different clauses are assumed not to be mutually interrelated. Based on these assumptions, it is expected that the number of parallel virginal trails will be in the order of the number of transformations, and that the concept of virginal subtrail (cf. the introduction of the subtrail concept in the previous paragraph, in connection with the SIA-range) will prove useful.

4.2.3.4e. Left- and right-looking conditions.

The SL-IL lexicon entries corresponding to the SL input words will contain information which may amount to an estimated 90% of what the parser needs: word class indications will trigger parse steps dependent on the parse history, noun subclasses will activate rules for SL-IL article mapping, many words will have valency or collocation information with them, often exerting a major influence on the parsing process.
Though a general SL grammar model remains indispensable to the parser, the big influence of lexicon-contained or lexicon-activated rules - together with the intervalwise parse progress - make it fruitful to regard the parsing as 'data-driven', even if it relies on a top-down general grammar model (which would only amount to an estimated 10% of the information used).

Procedures coded within the lexicon thus direct the parsing and the SL-IL translation to a large degree. These procedures essentially are condition-action pairs, of which the conditions refer to the co-occurrence of certain words (named literals, words of a certain class or semantic subclass) in the microcontext (the current clause or sentence).
A variety of procedures can be distinguished with regard to which position of co-occurrence their conditions specify: to the left, to the right or at either side of the current word, immediately next to it or not, etc. So we will have 'left-looking' and 'right-looking' conditions, in addition to universal ones.
Right-looking conditions will imply postponement of minor actions, such as the IL-translation of a word which can form collocations (e.g. the translation of the noun 'share' in

'share prices dropped', the translation of the verb 'miss' in 'I miss her', etc.). If syntactic structure is involved (e.g. the looking for an object with a predicative supplement, as in 'they found the report surprising'), these conditions will cause parallel trail initiation to anticipate a negative as well as a positive outcome of the search. If one outcome is marked 'rare' in the lexicon, then neither postponement nor trail-splitting may be observed, but preference given to later backtracking.
The same applies to universal conditions which have not found a match to the left of the current word.
Left-looking conditions, and universal conditions that find a match to the left, will generally cause some overriding of previously assigned syntactic functions and IL-translations ('on the building of the power plant depends ...', 'on the ship-yard they spoke ...', etc.). These conditions can immediately be applied during the interval in which they are accessed.
Right-looking and unfulfilled universal conditions remain active till a match has been found in later intervals. One should keep in mind that, at each particular parse-interval, not only the lexicon-entry corresponding to that interval's input word, but also all entries of preceding words (of the sentence) can be inspected by the parser.
Also, it should be realized that many conditions are not of an absolute nature with regard to what particular words or precize subclasses they specify: it is often a matter of preference, where more than one possibility presents itself. This implies the need of a continued search for equal or better matches, even after a match has been found; e.g. in:

(20) 'He convinced the proponents of this alternative possibility
1 2 3 4 5 6 7 8

of the extreme financial risks it envolves'
9 10 11 12 13 14 15

a prepositional object match for 'convince' will be established at interval 8 ('sentential noun' subclass, preceded by the preposition 'of'), another match will occur at interval 13 (on the parse trail in which 5-8 depends on 4). The subsequent detection of 14-15, a clausal supplement to 13, could cause the latter alternative to be 'rated higher'.

It is important to notice that the lexicon-contained procedures (at least their condition parts) must be fully SL-oriented. They look for SL-words and have been coded in terms of SL collocations, SL technical terms, SL idioms, SL valencies and SL frequency statistics. Some of the patterns looked for are invariant ('prime contractor', 'blast furnace'), others may vary with regard to the mutual positioning of their members ('it depends on ...' vs. 'on ... it depends'), according to a

largely known frequency distribution (of course, a certain demarcation of text type plays a role here [see Section V]). So one of the members of a collocation or valency may definitely or predominantly be the left member, the other the right member (ignoring, for the moment, the existence of 3-word collocations etc.).
Regarding the lexicon entry in which a collocation or valency should be encoded, the question arises whether it should be in its left or in its right member. One efficiency consideration is to take the less frequent of the two (e.g. 'leave', not 'on', in case of 'on leave'). Often however, there will be two head-words, or the frequency difference will be unsignificant ('blast furnace', 'cotton gin') which leaves room for other criteria.
Now a parsing process which intently follows relatively slow natural language input in one single LR parse, will certainly profit from an early signalling of potential collocations, valencies etc. by their left members (i.e. upon receiving these left members and accessing their lexicon entries). Such an 'early warning arrangement' solely requires lexicon building with a mild additional criterion. In our example 'share prices dropped last season', the inclusion of collocational information under 'last' secured a parse trail for a statistically probable continuation (which otherwise one would have been forced to tackle by backtracking).

In other words: an on-the-fly under-the-keyboard parsing system as DLT will prosper from a lexicon scheme which gives a slight preference to the mentioning of 'venture capital' under 'venture', 'capital punishment' under 'capital', etc. It is not a matter of success or failure, but rather one of the best distribution of parsing efforts and resources over the available intervals, thus contributing to the parallelism/ backtracking balance.
The measure in which such a lexicon scheme should be pursued also depends on the particular SL. As to the valency of verbs for prepositional objects ('it convinces us of .../) or the valency of nouns for infinitival clauses ('the fear to lose'), these are normally indicated in the verb or noun entry respectively. In SVO languages, the verb is most often and the noun is always the left member of these valencies, so everything is fine in that respect! But for SOV languages (Japanese, Turkish) other arrangements should be explored, to avoid an accumulation of postponed parsing work at the last word of the sentence. Suppose one had to cope frequently with English non-SVO patterns like

(21) 'on the building of the power plant, the future of our country depends'

(22) 'on the bus drivers dispute they spoke for two hours'

for which one had to design a verbal-valency early warning provision, then one would probably resort to a system which indicates this valency indirectly in the lexicon entry of 'on' in such a way that this entry refers to a group of verbs, including the subclass MENTAL ACTIVITY. The parse trail initiated for this interpretation of 'on' would then survive after locative ('on the bus') and other trails would have been rejected in favor of the nominalized-verb ('building') or tractative ('about the bus drivers dispute') interpretation. Semantic noun subclasses and numerous lexicon-contained collocations with 'on' would also have to be checked in order to achieve this, but - to be honest - even in the SVO case, problems like

(23) 'he spoke on the plane'

must be defied and will require the support of some semantics (the last resort is always the disambiguation dialogue).

4.2.3.4f. World knowledge.

The semantics used along the parse intervals is limited to selectional restrictions [see section 4.2.3.4b]. These must be regarded as an extension of syntax (and cannot be sharply separated from it). The lexical subclassification scheme on which this semantics is based (ABSTRACT, HUMAN, etc.) could be characterized as the 'common divisor' of such schemes known from literature and practical experience [Kelly, 1975]. In principle, the text range on which the basic semantics procedures operate is the current clause or sentence. Only a limited carry-over of (anaphoric) information from the immediately preceding sentence takes place. This restricted text range is called the 'microcontext'.

Different from this basic semantics, and of a higher degree of difficulty, is what one might call advanced or 'deep' semantics. The terms 'world knowledge', 'knowledge-of-the-world' and 'macrocontext' indicate well what is meant:

- knowledge of the 'world', i.e. of the objects, concepts, disciplines, the natural-language text is about (e.g.: 'a dead man can not drive a car' [Carbonell, 1981], 'a box does not fit into a writing pen' [Bar-Hillel, 1955]);
- knowledge of the situation, in particular the purpose of the natural-language text (report, instruction manual, inquiry form etc.);
- knowledge of the context, notably the whole of preceding sentences of the text (the 'macrocontext').

At the outset of DLT, the place and role of this advanced semantics in the SL-analysis will be modest, but can be demarcated clearly: advanced semantic procedures will be called upon after the end-of-sentence has been reached, in an attempt to avoid or reduce subsequent interactive disambiguation. As such, they form a separate substep in the Step-1 analysis process [fig. III-10], separated from the intervalwise and interleaved syntactic-semantic parsing.

The extent to which one will provide the system with advanced semantic routines is as much a matter of cost-effectiveness as of state-of-the-art: algorithm formulation and additional lexicon encoding must be weighed against the frequency and bore of specific interactive disambiguation requests. A cream-skimming approach will certainly be observed in this respect.
But apart from developmental strategy, there is also an operational issue: advanced semantics will tend to involve extensive searches (knowledge base, macrocontext), demanding relatively much processor time. It is not attractive to apply these searches before less-demanding efforts have been made, and this is one of the reasons why the advanced semantics will not be interleaved with the syntactic-semantic parse at separate intervals.

In our example 'share prices dropped last season', an imperative interpretation presented itself, already upon appearance of the first word. One could have tried to immediately verify the admissibility of the imperative in the existing situation or macrocontext, maybe not exactly a minor routine. Suppose the example would instead have read:

(24) 'share prices fell sharply last week',
1 2 3 4 5 6

then any resource-demanding macrocontextual attempt to check for an imperative interpretation of 'share' would have been in vain, because interval 3 would have ruled out such an interpretation anyway, on simple syntactic grounds.
Similarly, in the following case (an extension of the example given in fig. III-5):

(25) 'they saw the girl with the binoculars, after they had
1 2 3 4 5 6 7 8 9 10

managed to climb into a tree on the top of the hill'
11 12 13 14 15 16 97 18 19 20 21 22

one could imagine the appliance of a sophisticated semantic procedure (dealing with visual perception, optical instruments, cutting instruments etc.) at interval 7, to resolve the

homonymy of 'saw'; but at interval 10, simple syntactic-semantic considerations will reject a present tense (and thereby the homonymy of 'saw') in the main clause anyway.

Summarizing: the SL-parsing process of DLT is primarily syntactic-semantic (syntax extended with basic semantics). As subsequent and separate disambiguation steps after the parsing of the whole sentence, advanced semantics [see also section 6.2] and human interaction [see 4.2.4] will follow.

4.2.3.5. Existing techniques.

In the previous sections, we gave an impression of how an SL-sentence will be analyzed, and we explained the features of the SL-parsing process in DLT.
In this section, we will relate these process characteristics to various techniques, methods and viewpoints, more or less widely known in circles of natural language processing and MT. An up-to-date and instructive overview of this specialty field, to which we will often refer, is found in [King, 1983].

4.2.3.5a. Choice of grammar model.

The strict single-pass LR requirement leaves the following possibilities:

- an ATN parser (with trees or charts as intermediate output data structure);
- a 2-level grammar, consisting of a BNF and a restrictions component, based on linguistic string analysis and on the principle of insertion of adjunct strings [Sager, 1981];
- a deterministic parser [Marcus, 1980; Charniak, 1983].

If one adds the requirement that the technique must have been in use for a number of years and proved its practical feasibility for general purpose natural language parsing, then deterministic parsing might become relevant in (say) five years from now. A parser as Paragram [Charniak, 1983] could certainly be interesting for SL-analysis, both because of its ability to handle ungrammmatical input (spelling and grammatical errors) and because of its inherent parallelism.

The ATN (Augmented Transition Network) parser and the 2-level grammar both have a record of more than a decade now, and can be considered matured techniques. Despite historical and terminology differences, it can probably be shown that the 2-

level grammar could be represented by an ATN as well (its geometry would then correspond with the BNF, its augmented conditions with the restriction component). Though the definition of ATN s is a bit open-ended, ATN's appear to be a more generally known and supported technique than the linguistic string analysis of [Sager, 1981].

Another argument in favor of ATN as SL-grammar implementation is the fact that the SL-analysis is strongly IL-directed, whereas the IL-grammar relies on an ATN-representation itself (for reasons explained in IV.3.1). It will contribute to cost-effective development of SL-modules (or at least the first few of them), if ATN s are adhered to as one uniform grammar implementation model throughout the DLT system.
SL-ATN s for French, English, German etc. can be elaborated from the earlier developed IL-ATN (notably the IL-recognizer, the full version) as much as possible:

- the network geometry of SL-ATN's can be made to coincide largely with the IL-ATN. Only a number of arcs (including those from the initial state) need to be added, to reflect the larger freedom in SL word ordering. Some paths must be added in order to represent typical SL-constructions;

- the network augmentations (conditions and actions on the arcs will show many differences in detail between the SL and IL.

The idea is to retain as much correspondence between SL- and IL-ATN s as possible. This uniformity will positively affect the clearity, testability and maintainability of the system. Also, the correspondence will guide the way in which an SL-grammar can be developed phase by phase: First, one will aim at English (or French, German, etc.) limited to such a syntax model as coincides with (the geometry of) the IL-ATN. Specific SL-constructions (e.g. certain sentential constructions with the English gerund) will not be parsable then (or will be parsed incorrectly, e.g. an NP functioning as free adjunct) and can be added at a later stage.

The uniformity argument and the viewpoint considering SL-ATN s as 'variations' around a common basic IL-ATN backbone well reflect the philosophy behind DLT's IL design: many IL design decisions have been guided by the awareness of common or (in contrastive syntax) dominating patterns in the primary SL-candidates: French, German, English. This can for instance be observed from the prescription of word group order (basically SVO), but also from numerous more detailed issues [see Chapter IV]).

4.2.3.5b. Intermediate output data structure.

Although the final output of Step 1 has been specified as a tree, the formal type of data structure on the intermediate parse output along the SL-analysis has not been specified (the word 'tree' has been avoided as much as possible) in the preceding sections.
An important feature of the SL-parsing in DLT is the development of parallel 'trails' to accomodate ambiguity. A chart type of data structure could prove effective for this, and can be used in conjunction with an ATN-parser [Varile, 1983]. Note that, in such a configuration, the chart graph would have to develop in parallel (without serial iterations), in line with the strictly single-pass LR sequence of parsing.
The intermediate data structure must especially provide for the additional concept of 'subtrails', intended to represent local ambiguities (with limited SIA [see 4.2.3.4c]). Apart from a chart type structure, a technique like WFST (Well Formed Symbol Table), sometimes used in connection with ATN-parsers [Johnson, 1983], could be utilized. Because the sorting out of ambiguities and their mutual interdependencies is a crucial element in DLT's SL-analysis (in particular with regard to an optimally structured interactive disambiguation dialogue), computational schemes directed to ambiguity combinatorics [e.g. Church, 1982] may be helpful. Also interesting would be a device which indicates in a tree structure how the effect of an apparently local ambiguity propagates through the sentence, by including ambiguity-indicators in node labels, according to certain rules. Such a device could be similar to the 'semantic rule trees' described by [Ritchie, 1983]: instead of the semantic structure, the 'ambiguity profile' for each subtree is then recorded as soon as possible during the parse.

4.2.3.5c. Relation between strategy, grammar and lexicon.

In contrast to the following remark made by [Hutchins, 1982]: "As in all modern systems, EUROTRA will maintain strict separation of algorithmic processes and linguistic data...", DLT does NOT observe this strict-separation principle.
Whereas a system like EUROTRA has been characterized as an 'expert system' (viz. data base) on language translation, DLT has been conceived as a system closely integrated with the operational environment of word processors, personal computers, consumer-electronics and networks. A one-pass LR parsing of input text, an interval-wise IL-directed analysis, human interaction as an ultimate disambiguation instrument, these are all deliberate design choices and <u>key features</u> in DLT, not just values of strategy variables.
To put it more bluntly: where EUROTRA promises portable MT software, DLT promises a turn-key MT system of portable

hardware components, a difference one indeed expects between projects originated in a university vs. an industry environment respectively. A certain loss in generality is the obvious price DLT has to pay for the higher degree of concreteness in its design.

Strategy and grammar are therefore not strictly separated in DLT. The ATN-grammar model itself incorporates elements of a parsing strategy. Updating and maintenance of the grammar can however be enhanced by a proper choice of host language for the ATN [see section 5.1], apart from the fact that positive [Bates, 1978] as well as negative [Johnson, 1983] opinions exist on the perspicuity of ATN's themselves. Moreover, DLT too will have a strategy or control mechanism outside the ATN-grammar. This mechanism will shift control between lexicon-contained rules and the general grammar, will decide whether certain alternative parses are handled by parallelity or backtracking, etc.

Strategy and lexicon are well separated in DLT, enabling the use of a lexicon structure that could also be of use for many other systems or purposes. The only exception is the mild preference criterion for 'left-member encoding' [expressed in 4.2.3.4e].

Grammar and lexicon are not separated as to the power of the grammatical rules that they may contain: this principle is accordance with current MT developments like EUROTRA.
During the parsing of a sentence, not only the rules of the general grammar (embodied for instance in the ATN) apply, but also the rules contained in the set of lexicon-entries retrieved for that sentence. These lexicon-based rules can trigger tree transformations (corresponding to structural transfers such as illustrated in figs. III-13 and III-14) as well. In case of 'conflicts', lexicon rules and general rules supplement each other in such a way that the latter apply 'by default' of the former.

4.2.3.5d. Interleaved vs. sentence-final semantics.

Interleaved semantics in the form of selectional restrictions has been defended and amply used in practice by [Sager, 1981]. On this issue, the DLT design has also been strongly inspired by the pioneer work of [Kelly, 1975]. As the selectional restrictions will be used to select (between multiple alternative possibilities) rather than to filter out single interpretations, they are in line with Wilks' preference semantics [Wilks, 1979].

The usefulness of selectional restrictions has been affirmed by [Ritchie, 1983], who at the same time underlines their

limited application. For DLT, this limitation is quite acceptable, if one considers that the selectional 'basic semantics' are supplemented by other disambiguation techniques there [see fig. III-10: advanced semantics and interactive disambiguation]. Further, the possible improvements which Ritchie suggests (reference evaluation, computing the effect of modifiers on the semantic features of the head-word, etc.) could well be pursued by future intra-IL AI-routines [see also 6.2] in DLT.

Important for DLT is also Ritchie's remark [Ritchie, 1983] that the appliance of world-knowledge or advanced semantics would NOT contribute to structural disambiguation during the parsing. This justifies a separate, sentence-final 'advanced semantics' disambiguation step in the DLT design [see fig. III-10]. It also favors modularity: the highly advanced AI-routines [see 6.2] that will be required, can be added in a later DLT-development stage. As a sentence-final disambiguation provision, these advanced semantics routines and the subsequent interactive dialogue will be complementary in DLT.

4.2.4. The disambiguation dialogue.

The interactive disambiguation dialogue is the last substep of the SL-analysis [see fig. III-10]. Its existence is an important feature of the DLT system as a whole: it contributes to the system's 'outside look', and is experienced concretely by the users at the text-generation side [see also Chapter II].

With regard to the linguistic basis of DLT, the recourse to human intelligence helps to overcome the ambiguity barrier without the need to await future breakthroughs in advanced semantics and AI. This opens the way to high-quality translation with mainly basic semantics [4.2.3.4b]. The complementary functioning of man and machine [fig. III-15] is referred to as 'SAHQT' (Semi-Automatic High-Quality Translation).

Of course, the presence of a human being (with knowledge-of-the-world) at the WP-like terminal is a resource which must be utilized with care and reserve. In the computer industry, a lot has been learned on the psychological implications of man-machine dialogue design [e.g.: Martin, 1973]. Certainly, the WP attendant should not be overloaded, either quantitatively or qualitatively.

In conjunction with the syntactic parsing [see for instance the SIA concept in 4.2.3.4c] and the sentence-final semantics [4.2.3.4f], the number of questions to be put to the WP-operator should be minimized, an objective which can be achieved by arranging the questions in the right order (in such a way that

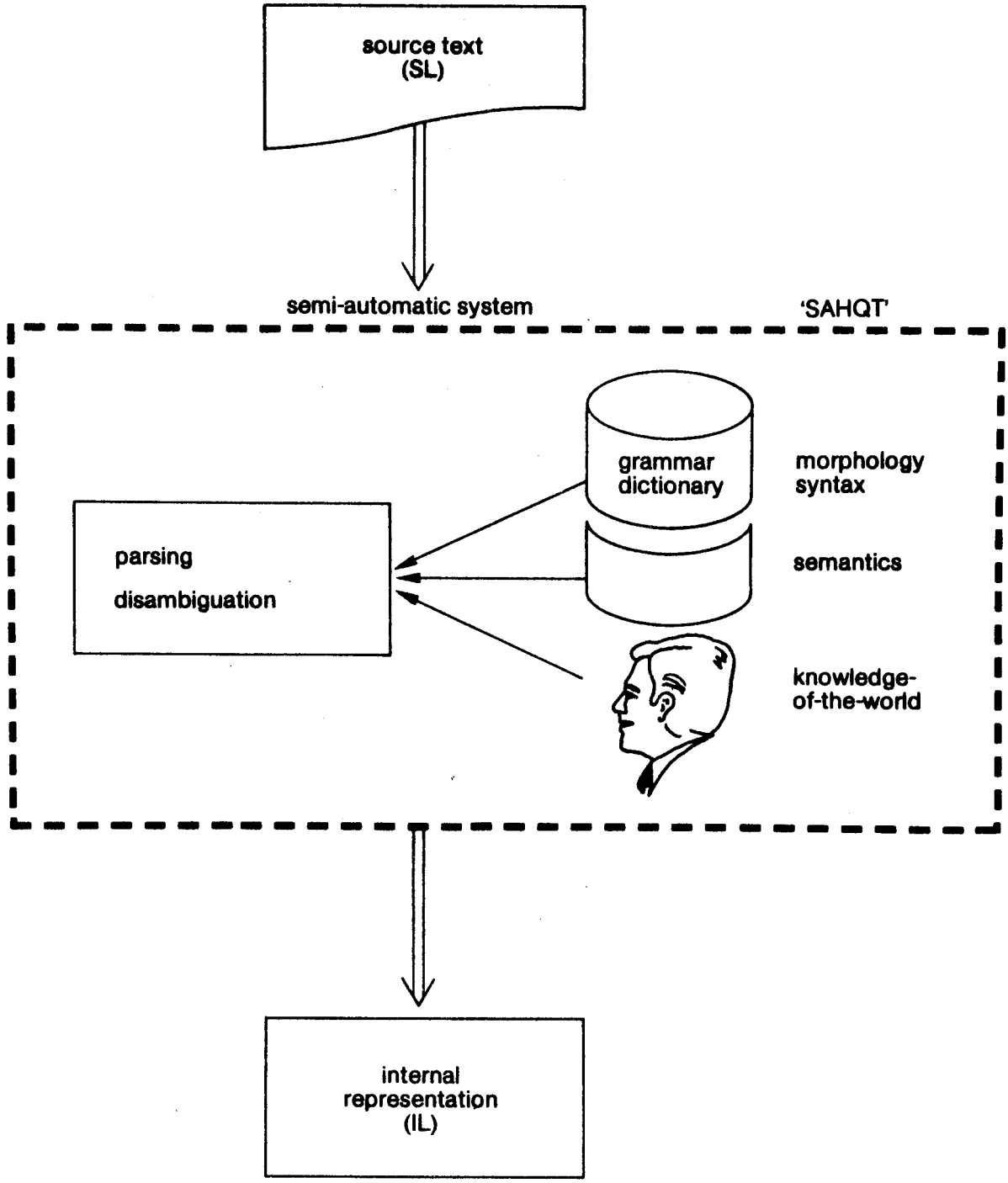


Fig. III-15. Outlook for the next couple of decades: the presence of a human being (at the text entry terminal) will still be necessary.
With his knowledge and intelligence, he will assist the machine via a computer-initiated dialogue.

most of the possible responses will discard the need for further questions).
At the same time, the questions should be kept clear and simple, allowing an educated person to comprehend and answer them quickly. There is no point in trying to restrict the number of questions tenaciously to 1 if one has to sacrify clearity for it (such an elaboration would also be very costly in terms of algorithm development).

The disambiguation by interactive dialogue applies to structural as well as lexical ambiguity. In connection with with translation speed estimates [see also section VI.2], an average of 3 question/response pairs per sentence has been assumed. There will of course be individual sentences without interactive disambiguation at all, an obvious aim for any sentence! Approaching this aim requires more semantics, i.e. more comprehensive lexicon entries, more sophisticated algo-

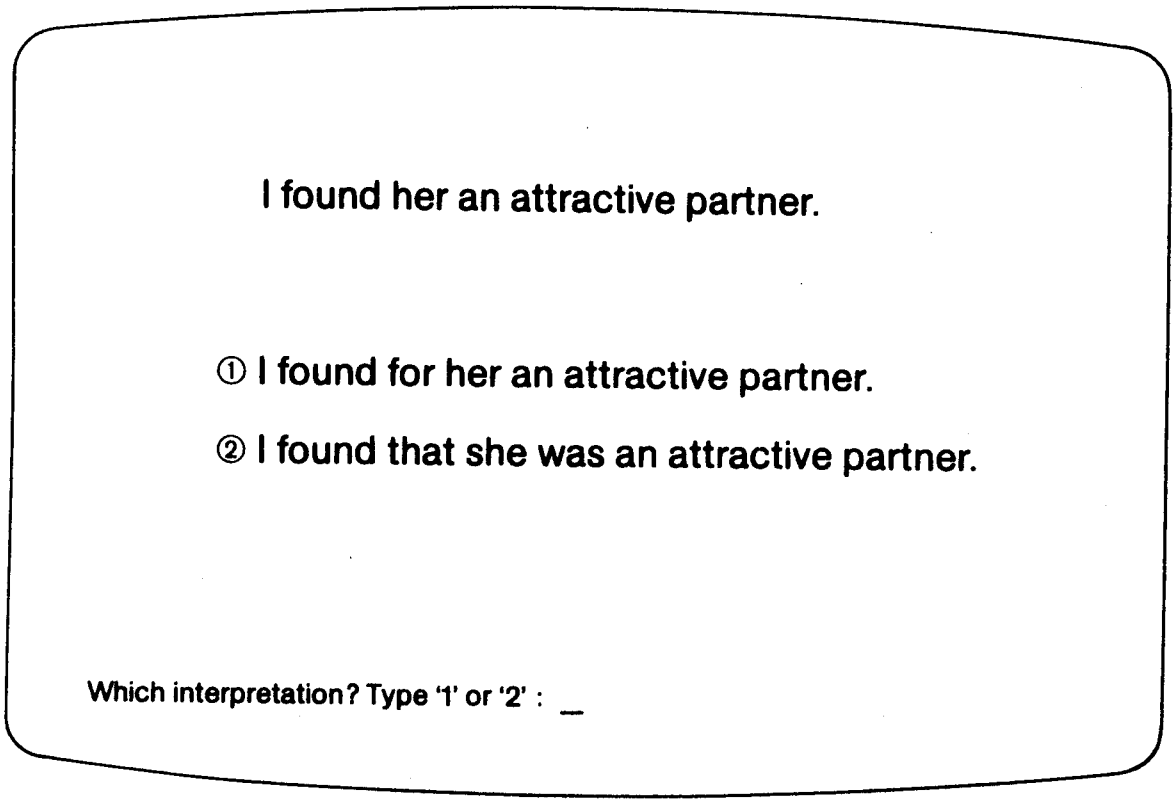


Fig. III-16. Example of a computer-initiated disambiguation dialogue, using menu-technique and automatic paraphrasing of the meanings concealed in the input sentence (the sentence at the top).

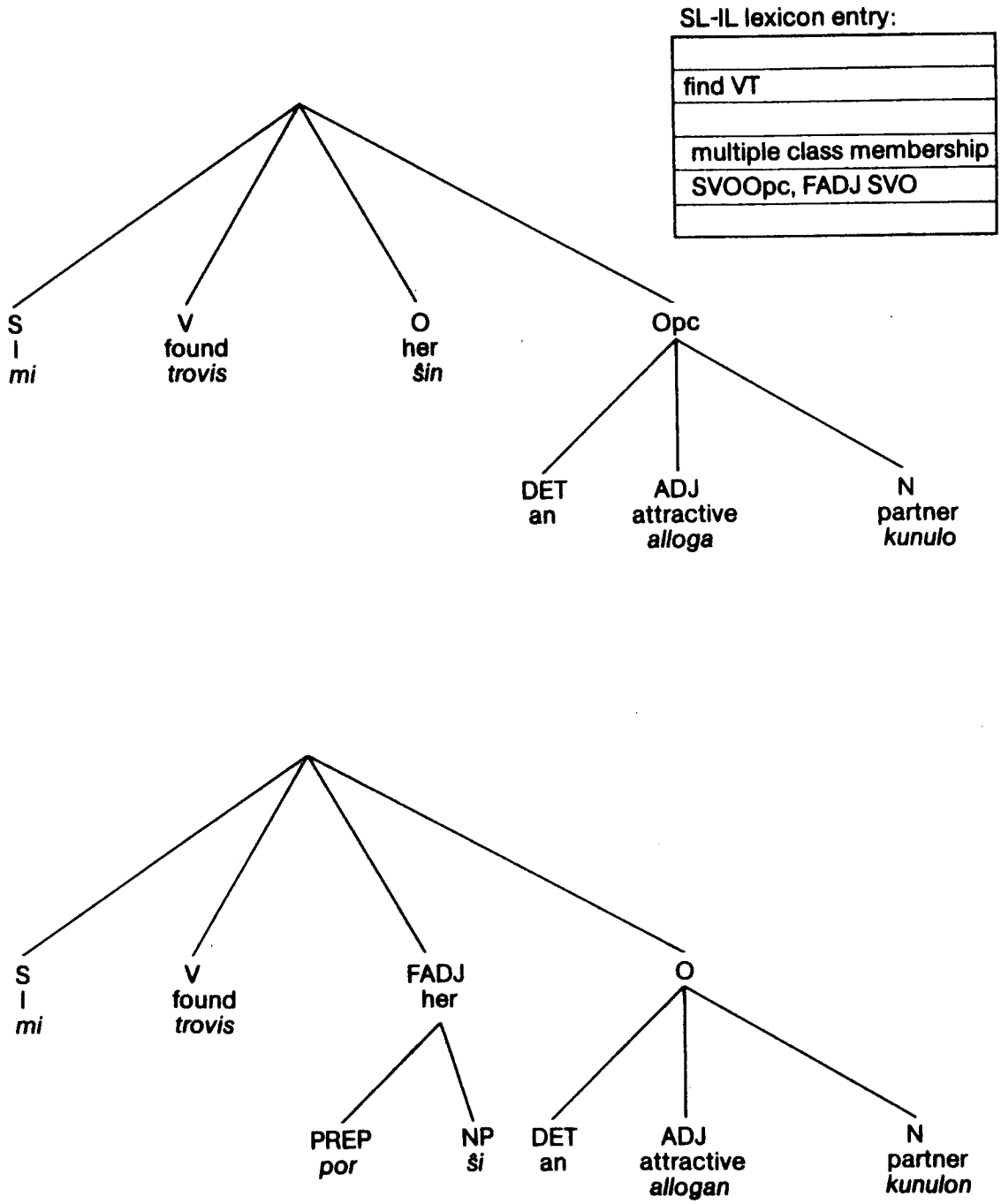


Fig. III-17. Pair of IL-trees resulting from previous SL-analysis, and underlying the interactive disambiguation of fig. III-16. The lexicon entry relates the various English meanings and valencies of 'find' to IL syntactic categories [see IV.1.4]. Notice that not only the IL-trees, but also the linear strings of IL terminals clearly distinguish between the two alternative readings.

rithms etc.: an evident trade-off between the work-load on the WP-operator and system development cost. In the years ahead, as DLT develops, the balance will gradually shift towards more automatic disambiguation [see also section 6.2].

Fig. III-16 illustrates a dialogue for structural disambiguation. The interactive process is set in if more than one parse tree results after the preceding SL-analysis [fig. III-17]. For either of the alternative interpretations of the SL-sentence, the IL-form has already been composed (in course of the IL-directed SL-parsing) before the dialogue starts. In other words: only a selection between two already existing IL-representations has to be made as a consequence of the disambiguation dialogue. The task of the dialogue module in DLT is NOT to generate a syntactic structure after having consulted the WP-operator.

The interactive dialogue requires the following non-trivial linguistic tasks to be performed by the system:

- Planning of the dialogue.
  This is the sequencing of questions in such a way that the first will probably make redundant the second, etc. The SIA concept (as already mentioned above) and probabilistic data in lexicon-entries will play a role here.

- Generation of paraphrases.
  Though the dialogue module has NOT to generate sentence structures as a result of the interaction, it has to do so in order to enable a smooth interaction. The difficulty lies in how to present an ambiguity to somebody (as with optical illusions, some people perceive only one interpretation). This explains the necessity of two paraphrases, as in fig. III-16.
  The automatic generation of such SL-paraphrases is a process comparable to TL-synthesis. In general, it may involve syntactic transformations, valencies, morphology etc. If the ambiguity is lexical, the lexicon-entry has to provide suitable synonyms for use in dialogue displays.

In connection with this linguistic support, as well as with dialogue design and modern display techniques, the following principles will be observed for the interactive disambiguation in DLT:

- Computer-initiated dialogues: the WP-operator has no other responsibility than to react to the issues submitted to him by the DLT system. Apart from intelligently responding to questions, and occasionally overriding 'default' selections or AI-based assumptions displayed to him, the human role is a passive one: the WP-operator cannot instruct the translation system spontaneously.

- Menu technique: the selectable alternatives are presented in such a way, that only a minimum amount of keyboard input by the WP-attendant is required - following his reflection - such as a number (as in fig. III-16) or a YES/NO response. This principle prevents the need for linguistic analysis of lengthy responses, which again could introduce ambiguities or input errors. The absence of textual responses also speeds up the dialogue.

- SL only: everything in the dialogue will be kept in the source language, i.e. the language of the input text to which the dialogue refers. This applies to synonyms and paraphrases (emphasizing the different interpretations of a sentence), but also to any other texts (instructions, reminders etc.) displayed. The person attending the terminal is assumed to be monolingual!

- Avoidance of jargon: it should not require a degree in linguistics to operate a DLT-terminal. Preference will be given to paraphrasing, avoiding any technicalities in the formulation of the problem at all, e.g. in case of the input sentence (an example derived from [Carbonell, 1981]):

  ISRAEL SEIZED LARGE QUANTITIES OF NEW WEAPONS FROM THE USSR

  the disambiguation dialogue could be arranged in a veruy 'natural' by presenting the paraphrases:

  1. WEAPONS FROM THE USSR WERE SEIZED.
  2. THEY WERE SEIZED FROM THE USSR.

  For the same example, a less expensive (in terms of linguistic system development) but still acceptable solution would be the more 'technical' formulation:

  'FROM THE USSR' REFERS TO:

  1. SEIZED.
  2. WEAPONS.

  Another compromise between linguistic (paraphrasing) sophistication and 'natural' problem formulation is highlighting: the ambiguity issue may be brought to the terminal operator's attention by underlining the relevant words, or by using double density or color:

  1. ISRAEL SEIZED LARGE QUANTITIES OF WEAPONS FROM THE USSR
  2. ISRAEL SEIZED LARGE QUANTITIES OF WEAPONS FROM THE USSR

Finally, the use of simple grammatical concepts (verb, noun, subject, object) may be more tolerated in certain cases [such as in the example of fig. II-1a] and in particular during transitional development phases of the DLT system, where it may be more cost-effective.

- 'Default' and 'override' options: in case of a frequent repeat of the same lexical ambiguity within one piece of text, a corresponding repetition of the same dialogue would be annoying. Instead, the WP-operator will have the option to specify a global interpretation 'by default': for instance in case of the word 'bank' in a financial text, he does so at the first disambiguation dialogue devoted to this word. All subsequent occurrences of 'bank' will then cause the system to display a reminder:

  BANK = FINANCIAL INSTITUTION

  to which the user can react either by giving his agreement (hitting the space bar) or by overriding (typing 'N'), after which the original selection menu for the different senses of the word 'bank' will be redisplayed. The need for an override possibility can be explained by the general experience reported in MT literature, that homonyms cannot simply be handled by assignment according to the special text category in which they appear (i.e. in a financial text, one cannot exclude the occurrence of 'bank' in a non-financial sense).

- Limited interruption: the computer-initiated dialogue will NOT interrupt the WP-operator during the typing of a sentence. The dialogue can only be started after the completion of a sentence. This is connected with the internal sequencing of the SL-analysis [fig. III-10], but is also a principle of terminal-user psychology. It permits the WP-operator to enter a sentence without being disturbed by DLT's disambiguation. For users who prefer to type there whole text undisturbed, and work through all the dialogues afterwards, a queued-translation and postponed-dialogue mode [see section VI.2] will be offered in addition.

Though the function within the overall DLT process and the principles of the interactive disambiguation have been laid down, the dialogues should be tried out in practice for further evaluation. Therefore, a simulation of these dialogues as part of a further study of the SL-module is proposed for the pilot project [see VII.7].

4.3. Principles of the TL-synthesis.

The TL-synthesis consists of the DLT process Steps 5 and 6 [see fig. III-9], and is a fully automatic process, NOT involving any human interaction.

In the MT field in general, TL-synthesis is judged to be a less difficult task than SL-analysis, due to the presence of ambiguous input to the latter, and the assumed absence of it to the former.

To a certain extent this is also true for DLT, and can be very loosely phrased as follows: "Automatic translation from Esperanto to English, French, German etc. is possible; in the reverse direction, it is not".

As we have seen in this Chapter, ambiguity is a relative concept in translation, and there is no pretension that DLT's Esperanto-based IL, from which the TL-synthesis departs, is 'ambiguity'-free with respect to the target language.
As illustrated in figs. III-6c, III-7b and III-8b, divergence at the IL-TL interface requires TL-specific word choices in addition to the selection of syntactic constructions, variables such as time and aspect, etc. This 'residual disambiguation' relies on comprehensive information (idioms, collocations, microcontext-based procedures) in the IL-TL lexicon. As stated in 3.2.5, such an approach can lead to stylistically fitting TL word choices and a high overall translation-quality.

The TL-synthesis in DLT is therefore by no means a trivial subprocess. It requires (largely lexicon-based) sophistication from the beginning, and is destined to grow into a powerful language-generation module (including macrocontext-based procedures and AI) at long term [see fig. III-21].
Though it can and probably will make use of components (TL lexicon material, TL-morphology) developed in other MT projects [section VII.7], it shows one particular DLT-specific feature [see also: Witkam, 1983: 203]: macro- and micro-contextual procedures (even if TL-specific) can be formulated (in the IL-column of the IL-TL lexicon) in terms of IL, and executed on the IL-representation of the sentence. Because of the straightforward morphology and the unique, compactly coded morphematic structure of the IL, this guarantees fast and efficient pattern matching operations.

Like the SL-analysis, the TL-synthesis is a bilingual process [fig. III-3]. There is a difference however in the further internal structure of these two major subprocesses. Whereas the SL-analysis is an SL-IL interleaved 'direct translation' [as explained in 4.2.3.2], the TL-synthesis can be split into the following segments:

- an analysis part, DLT process Step 5 [fig. III-9], being the monolingual IL-parsing (string-to-tree conversion);

- a transfer part, which has as its input the canonically ordered IL-tree left by Step 5; this transfer part is the dominating and bilingual stage of Step 6, involving IL-TL structural transfers, TL-specific word choices via IL-based procedures etc.;

- a synthesis part (in the narrow sense): a TL monolingual, mainly morphologic and word-ordering part, which concludes Step 6.

Thus, within the second main-stage of the overall DLT (double translation) process, a transfer system (with a substantial transfer stage) appears to be embedded.

5. Participation of translators and linguists in DLT's development and maintenance.

5.1. Expert-system interface.

Referring back to the chronology presented in fig. I-1, it should be noted that MT systems tend to have a very long life cycle. SYSTRAN for instance, which originated in the early 1960s, is likely to extend its services into the 1990s. Also EUROTRA is presented as a growth system [King,1982], and so is - in fact - DLT.
These systems require a policy and framework for long-term (i.e. decades of) development and maintenance of software, including the dictionaries with their steady updates. These dictionaries, as we have seen [section III.4.2.3.5c] are by no means simple word-lists: numerous lexicon entries contain routines with the power of a grammar.
In addition, the general grammars for the gradually increasing number of supported SL- and TL-modules (as well as the IL-grammar in DLT) will require constant tuning in the first years after their installation.

There seems to be general consensus about the necessity for far-going modularity in natural language processing systems now. In particular, it should be possible to separately specify, change and cancel a grammar or idiomatic rule, without concern about the intricacies of the MT system's internal operation. Such a rule is only a description of the linguistic reality, and should be considered as 'data' as opposed to the systems internal procedures and algorithms that work with it.

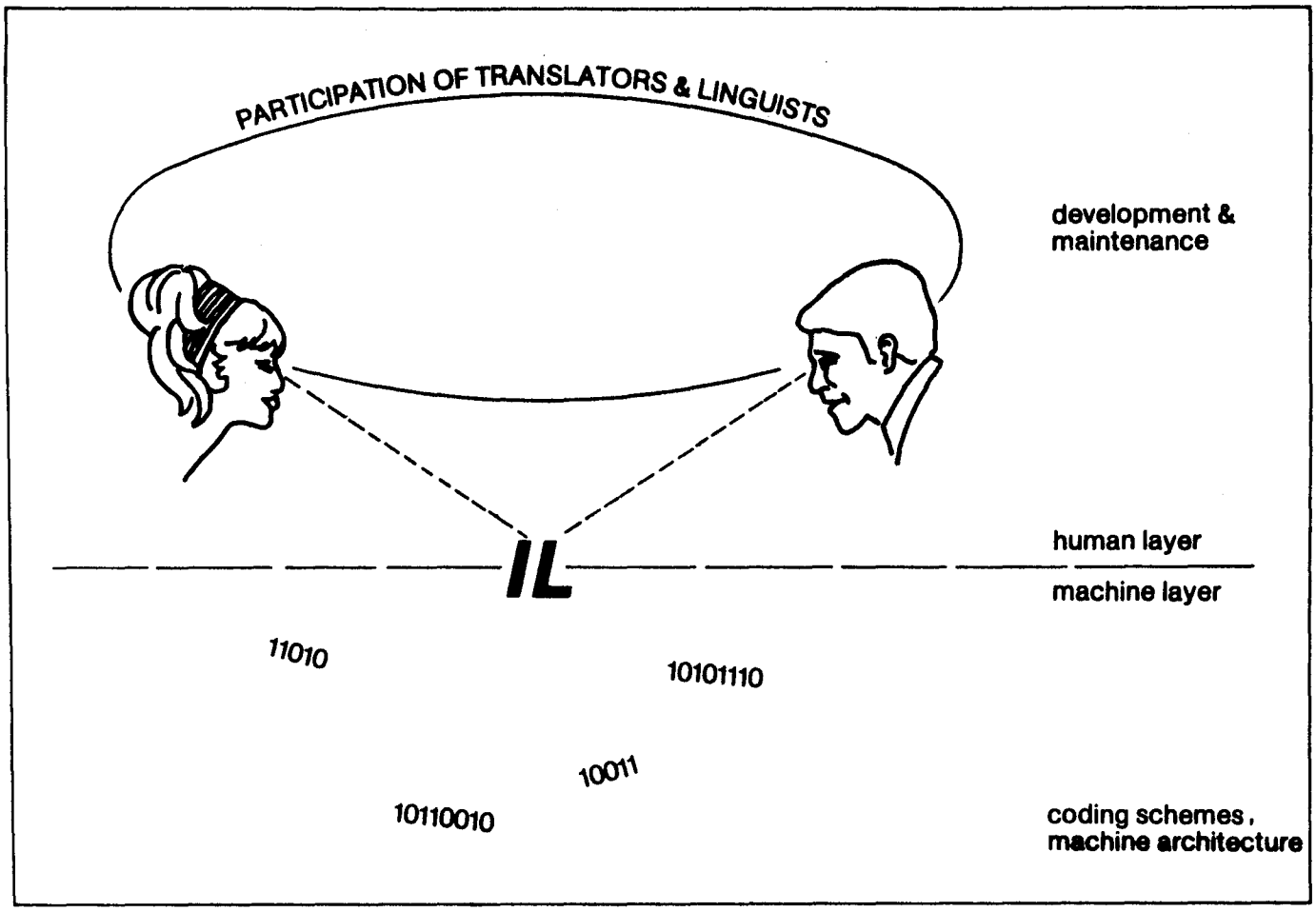


Fig. III-18. The backbone of DLT, the IL (Intermediate Language), has an internal (machine) and an external (human) face. The latter looks like Esperanto and will make the multilingual system's development accessible to a wide circle of translators and linguists.

Whereas the 'procedural' component of an MT system must be trusted to computational linguistics close to the system and part of the development team which gets it off the ground, the 'declarative' component should profit from contributions out of a wider circle of linguists and translation specialists, who are closer to the intricacies of a particular language (SL or TL such as French, German, English etc.).

Also DLT will provide an easy interface for language-oriented developers. This interface will be based on the use of PROLOG [see VI.4.1] as host language for DLT's ATN-grammars. The

latter should be regarded as fast and compact system-oriented versions of the grammars; in DLT, the emphasis is on their performance (in a specific, microprocessor-based hardware design [see VI.3]).
Implicitly, a conversion from the language-oriented declarative form to the system-oriented procedural ATN-form will be needed at each change or addition. The performance (speed, storage consumption) of this conversion mechanism (which will be supported by DLT's development facility [see VI.4.2]) is considered relatively unimportant.

Similarly, lexicologists will be able to enter and update dictionary items at VDU terminals via a self-explaining, error-screening and interactive data-entry program. The convenient external rules and formats will then be converted to compact internal routines and representations automatically.

5.2. Typical DLT aspects.

The outstanding characteristic of DLT is of course its natural-language resembling IL.
The value of this for inspection of the system's main interface, for troubleshooting and maintenance, has already been indicated in Chapter III [see fig. III-9]. It is to be reminded, that the IL plays also an important role in the DLT dictionaries: these are either SL-IL or IL-TL bilingual lexicons, with fully developed IL-columns, requiring IL-familarization of participating lexicologists as well.

The grammatical and lexical properties of the IL, with numerous examples of its appearance, can be found in Chapter IV, which approaches a complete definition of the language. As will be pointed out in IV.1.1, extralingual elements such as labeled brackets have deliberately been kept out of the language, in order not to jeopardize its readability.

The natural form of the IL and its closeness to existing Esperanto will not only improve development and maintenance conditions in the proximity of the system, but will also allow dissemination and discussion of specific translation problems within a wider circle of experts [fig. III-18], similar to the use of ALGOL for algorithm publication.

Of course, the IL has its computer-internal representation: a compact variable-length code scheme, with the morphem (NOT the character) as code unit. We call this code scheme BCE (Binary Coded Esperanto). Each binary code element has a fixed 1-1 relation with an IL morphem, a morphem being a word root, affix, grammatical ending or punctuation mark. The internal-to-external conversion is a simple straightforward process (a homomorphism).

However, the utility program for displaying IL-sentences on a VDU may offer options that facilitate the IL's inspection on specific aspects. One example of a useful option is identation, which would produce:

la atribuado

far la komunumo

de grenojn

al afriko

instead of the standard linear display mode, which makes use of extra spaces or underscores (separators which are essential elements in the IL [see IV.1.1]):

la atribuado far la komunumo _de grenojn __al afriko

(the assignment by the community of cereals to Africa)

[the English translation is not part of the display].
Another example of a possible display option is the hiding of disambiguating prefixes of prepositions, such as 'iam-' and 'ie-' [see section IV.2.5.2]. If the semantics of prepositions is not at stake in an IL inspection, then these prefixes could be obtrusive.

6. DLT's long-term prospects.

6.1. Stylistic improvement and optimization.

During the first years of operation, DLT's TL output (though grammaticaly correct) will more or less reflect the structure and style of the SL from which the text originates [fig. III-19a]. As an example, let us consider the following SL-sentence:

(26a) Ils traversèrent la rivière à la nage.

Human translators would produce the following equivalents:

(26b) They swam across the river.
(26c) Sie überschwammen den Fluss.

However, the Esperanto-based IL is capable of imitating each of the three above structures, and - what is even more important -

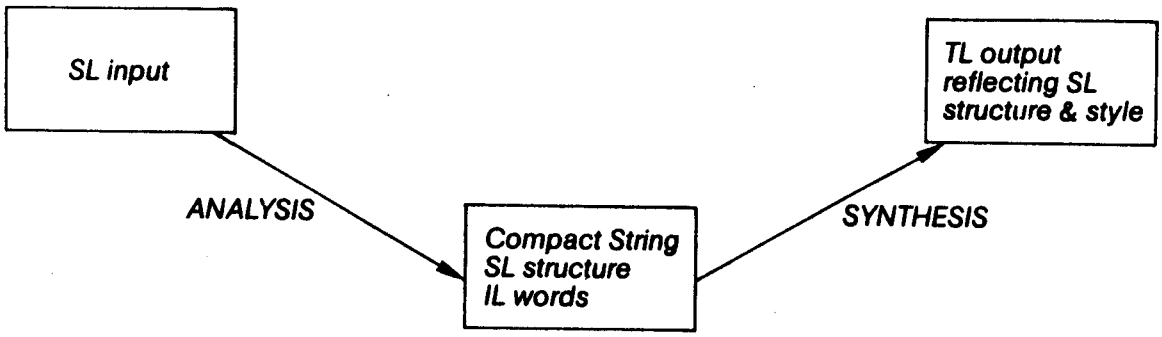


Fig. 19a. Short-term outlook for DLT: the IL will tend to 'pass' the SL structure and style to the TL.

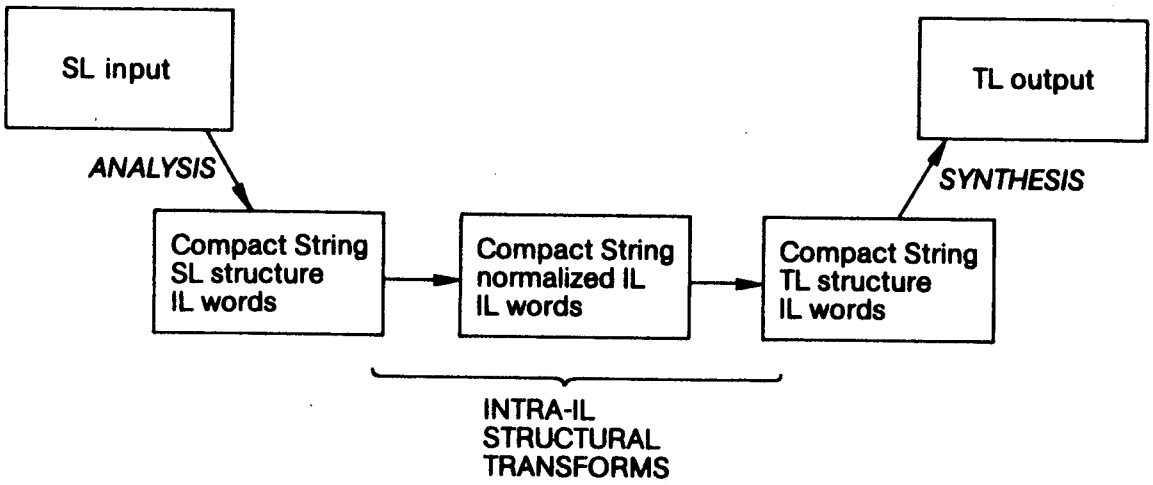


Fig. 19b. Long-term outlook for DLT: a stylistically normalized IL will 'decouple' TLs from SL influence. The intra-IL structural transformations will be arranged as part of the SL- and TL-modules.

shows no 'preference':

(27a) Naĝante ili trairis la riveron.
(27b) Ili naĝis tra la rivero.
(27c) Ili tranaĝis la riveron.

Unless the TL-modules are very sophisticated, this situation will result in TL ouput such as:

(28a) They crossed the river swimming.
(28b) Schwimmend überquerten sie den Fluss.

A future improvement could be based upon intra-IL mappings, in such a way that a 'preference' IL-representation is established (e.g. 27c) and other versions (27a, 27b) are automatically being mapped to it. Dictionary information should play an important role in this.
The result would be a 'normalized' IL and a stylistically more uniform input to the TL-modules. Sophistication of the latter remains a requirement, but will be facilitated by intra-IL (IL-to-IL) mappings preceding their IL-TL operations [see fig. III-19b].

Further, improved TL word choice may be obtained by lexicon extension with macrocontext-oriented procedures, which brings us on the following subject.

6.2. DLT and AI (Artificial Intelligence).

Though we make ourselves no illusions about the necessity of human assistance (via an interactive disambiguation dialogue [see section 4.2.4]) for the next couple of decades, we assert that DLT - with its particular IL - is an excellent platform for AI-enhancements. The same properties (accessability, compactness, pattern matching speed) which make the IL attractive for linguistic purposes, also make it interesting for language-independent or language-neutral world-knowledge implementation [see also Witkam, 1983].

The idea is to steadily develop a world-knowledge bank written in IL, which will be consulted in addition to the lexicon, during TL-synthesis as well as during SL-analysis. This knowledge-bank will be replicated on each DLT-terminal (by means of an optical disc [see VI.1.2]), and will be accessed by macrocontext-oriented algorithms run on a dedicated AI-processor.
This will pave the way for DLT as a 'knowledge-based MT' [Carbonell, 1981] or a 'Type-B' system [Hendrix, 1981]. Fig. III-20 shows the ultimate objective, and fig. III-21 gives another characterization of DLT's translation mechanism, with emphasis on its future evolution.

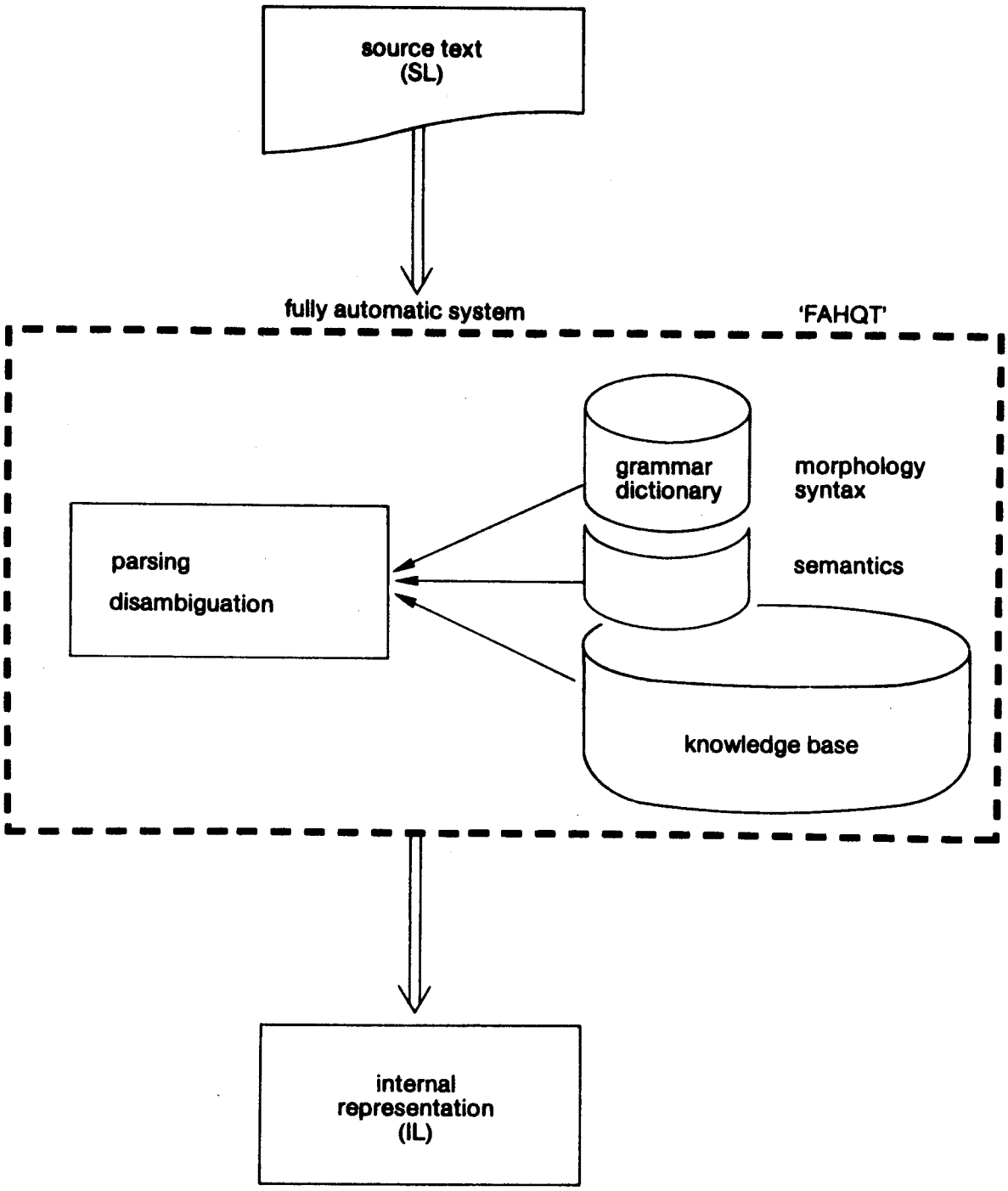


Fig. III-20. Long-term ultimate objective: a fully automatic SL analysis, implying a fully automatic DLT. The knowledge-base will be replicated at each terminal.

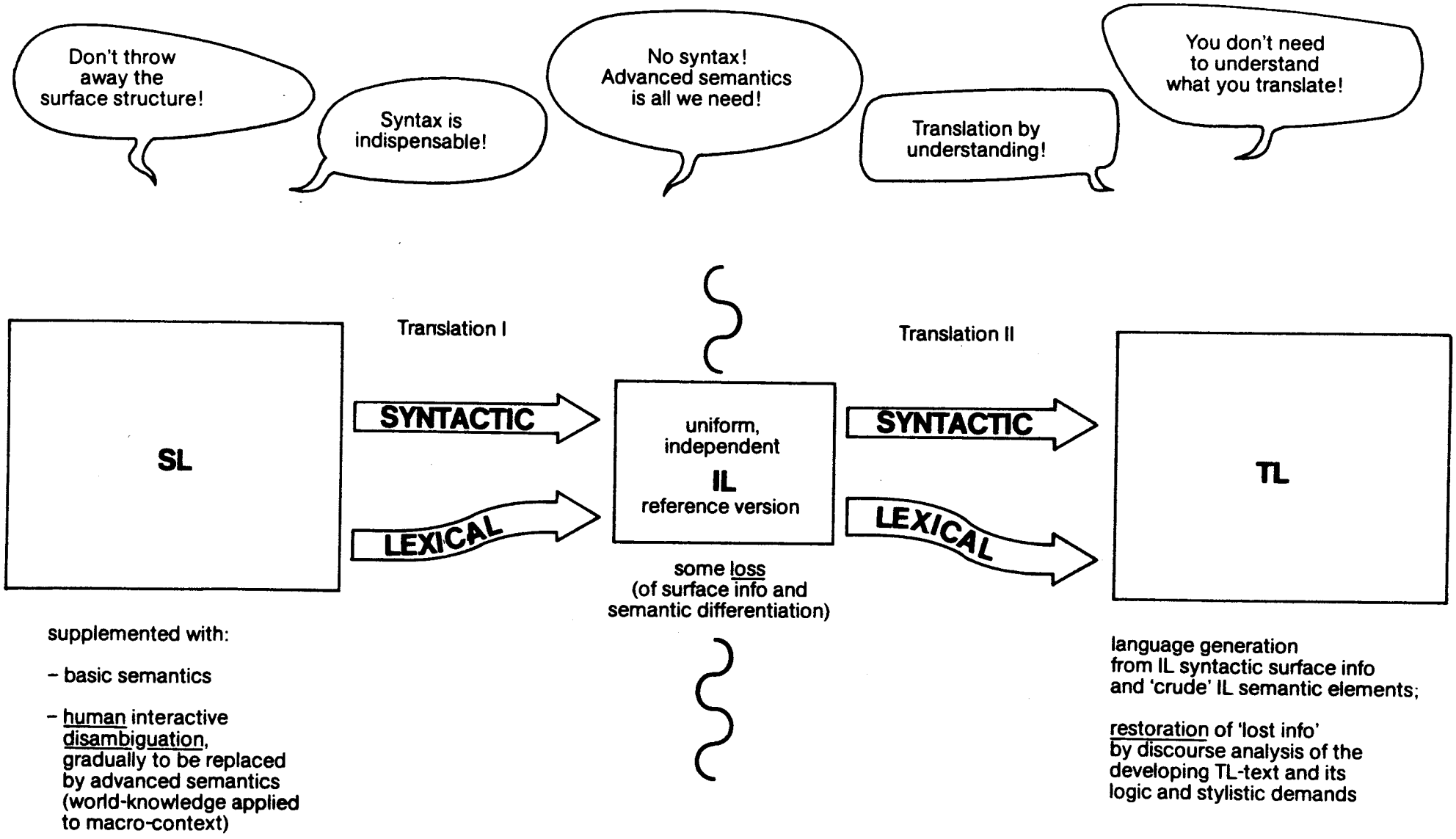


Fig. III-21. Profile and prospect of DLT's translation method against a background of conflicting views in linguistics and AI (Artificial Intelligence).