# Detailed Description of Logos Model Modules

- **Format -** This module extracts all mark-up commands from input text (HTML, Interleaf, etc) for subsequent re-application to target output. In some cases, mark-up information is made use of during analysis, e.g., bold facing in a string such as '***Insert*** *key*' helps analysis see ***Insert*** as a label rather than as an in-line verb. This module also segments text into discrete sentences. (There is as yet no extra-sentential handling, i.e., no sentence carry). Sentence length is arbitrarily set at 70 elements (including punctuation). Sentences exceeding this limit are automatically broken up into smaller sentences at semicolons or other potential breakpoints.

- **Lex -** Words of NL sentence are looked up in this module (on longest match principle) and are immediately converted to their corresponding SAL element (or elements, where entry is syntactically ambiguous). Functional characteristics of lexicon are as follows:
  - no practical limit on word length in a lexical entry.
  - no limit on number of words in a lexical entry. However, an arbitrary ten-word matching limit is currently imposed by the **Lex** software module.
  - no limit on number of parts-of-speech associated with a given entry. However, a **Res** limitation (see below) requires that only three parts of speech can be extracted for analysis purposes, a limitation meant to be compensated for by various stratagems involving **Semtab**. (The word *building* concerns seven parts of speech.)
  - no limit on number of meanings associated with a given part-of-speech of a given entry, insofar as such entries are differentiated by subject matter domain or user ID. However, there is only one default entry that will automatically be selected if no other selectional criterion is found to apply. This can be a serious limitation in the case of polysemous common nouns, one that is only very partially overcome by recourse to **Semtab** (See Section 8 and Fig. 18).
  - domain codes are hierarchical, allowing matching logic to favor the more specific codes in a selectional list and to default to the more generic.
  - part-of-speech disambiguation on morphological grounds immediately eliminates certain word classes from further consideration for a given entry.
  - In German source, compound nouns that have no lexical match are decomposed and separate elements are looked up. Nothing comparable is done for composite nouns in English source.

- **Res1, Res2 -** Each **Res** module accomplishes a single pass of the sentence, from left to right, collectively effecting a macro parse. No parse tree is formed (no concatenation) but homographs are resolved and all clausal boundaries and clausal relationships are identified. Information regarding the macro parse is passed on to subsequent **Parse** modules to serve as top-down guide for the progressive, bottom-up micro parse effected by these modules. This top-down picture enables the micro parse, when looking at a noun, for example, to know the structural context of that noun, e.g., what sort of clause the noun is in (e.g., a relative clause inside a dependent clause), whether the noun precedes or follows the verb of the clause, the SAL class of that verb, etc.

  For English source analysis, the principal work of the **Res** modules is homograph resolution. The SAL information coming out of **Lex** shows all possible parts-of-speech that the original NL word was found to have and that remained unresolved after morphological analysis. It is the task of **Res1** and **Res2** to resolve such ambiguous forms to a single part-of-speech. The following are typical examples of the work of these **Res** modules. Two sentences of similar construction are shown. Note that both have '**ADV ING N'** strings at the end which have to be resolved quite differently. We show the effect of this resolution on unedited machine output for French, other translation flaws notwithstanding.

(10a) *We spent time doing tasks such as systematically classifying documents.*

(10b) *We spent money eating things such as really satisfying pastry.*

(10a') *Nous avons passé le temps à accomplir des tâches telles* que *la classification systématique des documents.*

(10b') *Nous avons dépensé de l'argent en mangeant des choses tels que la pâtisserie vraiment satisfaisante.*

In (10a) the **ING** form *classifying* is to be resolved to a verb, in (10b) *satisfying* must be resolved to an adjective. Both **ING**'s came into the **Res** modules as two parts of speech (second infinitive and verbal adjective). In (10a), the key to resolution was provided by *tasks* SAL-coded as a member of a 'verbal abstract' noun grouping that anticipates verbal complementation. When a **Res2** rule encounters this Type of noun in the input stream, a top-down verb-expectation bias is established which allows a subsequent rule to interpret the **ING** form as a verb. In (10b), the absence of any such bias (or other determinant) allows the default resolution to obtain, namely to the adjectival form of the **ING**.

**Res2** affords limited look-ahead capability for avoiding garden path situations. In the case of the classic garden path construction in (11b), below, a rule attempting to resolve the syntactically ambiguous form, *run*, to the main verb of the sentence must first invoke a look-ahead function to insure that no unambiguous main verb is to be found to the right. Such search-ahead logic is also pattern-rule based, and while generally effective in shorter sentences, may run into difficulties in longer constructions. Notice, in the raw French output below, how *run* has been resolved to an intransitive verb in (11a) and to a transitive verb in (11b).

(11a) *The horses run by the barn.*

(11b) *The horses run by the barn are tired.*

(11a') *Les chevaux courent par la grange.*

(11b') *Les chevaux dirigés par la grange sont fatigués.*

**Res** modules share a dedicated **Semantic Table** which supports a variety of resolution strategies. For example, a selectional restriction rule in this table helps resolve the noun/verb homograph *loan* to a noun in the context of words like *office, form, value, officer*, etc, as may be seen in (12a) and (12a'), below.

(12a) *We sold everything from desks to loan office furniture.*

12a') *Wir verkauften alles von Schreibtischen zu Darlehensbüro-Möbeln.*

While the number of rules in this table is in the thousands, relatively few actually deal with selectional restrictions, this stratagem never having been found to be particularly efficient or effective in our experience. Selectional restriction rules generally have weak authority, as may be seen in (12b) and (12b'), below.

(12b) *We do not allow our employees to loan office furniture.*

(12b') *Wir ermöglichen unseren Angestellten nicht, Büromöbel zu leihen.*

In (12b), the **Res** rule that registers *allow* as 'pre-verbal' causes a subsequent **Res** rule to outbid the **Semantic Table's** selectional restriction rule otherwise applying, causing *loan* to be resolved to a verb, as the *leihen* in (12b') shows.

In German source analysis, the contribution of the **Res** modules to the parse is more restricted, addressing primarily (i) the ambiguity of noun case markings; (ii) *die, der, das* part-of-speech resolution (to the extent possible). Given the nature of German morphology, it may take the entire pipeline analysis to resolve certain ambiguities. For example, **Res** can readily resolve the part of speech and case of *der* in *der Mann*, but must defer final resolution of *das* in *das Bier* to a subsequent pipeline module. For example, in *das Mädchen, das Bier bringt*, resolution is not effected until the relative clause handling done in **Parse2**.

A graphic illustration of the **Res** macro parse may be seen in Fig. 11.

- **Parse1 -** The four **Parse** modules effect a micro parse of the SAL input stream, building on the output of **Res** and producing a final, single, bottom-up parse tree. Although **Parse** software modules are almost identical programmatically speaking, the compositional approach implied in pipeline architecture presupposes that each **Parse** module will perform a specific range of bottom-up parse functions. The output of each **Parse** module serves as input to the next, affording a progressively more abstract analysis of the sentence.

  Typical operations that **Parse1** accomplishes (not always successfully):

  - Simple **NP** formation (excluding noun series, **REL** and **PP** attachments). Parse tree nodes for **NP** are annotated for **NP** properties (definite, indefinite, SAL Type of adjective modifier if any, etc.).

  - Scoping of adjectives. Scoping of **AJ** in **AJ N N,** e.g., is achieved by sending the adjective and each noun in turn to the **Semantic Table.** In the noun phrases below, with unedited translations, note that the adjective in (13a) applies to the *modifier* noun, *boys*/*garçons.* In (13b) and (13c), the adjective applies to the *head* noun. Default scoping is to head noun. Adjective scoping remains a relatively weak area in **Parse1** analysis.

    (13a) *Smart boys school*
    (13a') *École de garçons intelligents*
    (13b) *Large boys school*
    (13b') *Grande école de garcons*
    (13c) *Smart language students*
    (13c') *Élèves de langue intelligents*

  - Auxiliary verb phrase analysis, concatenation and labeling. For example: *ought to have* [+ verb, past t.] → **AUX(MOD;psma)** [Form = past subjunctive modal active].

    (14) *He ought to have gone home.*
    (14') *Er hätte nach Hause gehen sollen.*

  - Adverbial phrase recognition and concatenation. For example:
    - *in general* → **AV(SENT)** [sentential adverb] (Note that *in general* is not lexicalizable. Cp., *in general terms, in general quarters.*)
    - *all morning/day//year/etc. long* → **AV(TIME)** → *toute la journée/etc.*

  - Resolution of **ING** forms. **Res** determines when an **ING** form is to be seen as nominal, but leaves it to **Parse1** to decide whether, in the case of some forms, the **ING** is a concrete noun or gerund. Source analysis and target transfer of **ING** forms are effected by close interaction between **Parse1** and **Semtab**. This remains a still largely unexploited area of **Parse1/Semantic Table** interaction but the examples below, with unedited output, at least show the current possibilities:

    (15a) *He saw a building.*
    (15a') *Il a vu un bâtiment.*
    (15b) *He witnessed the building of the dam.*
    (15b') *Il a vu la construction du barrage.*

(15c) *The device has a variable speed* *__setting__*.
(15c') *L'appareil a un* *__réglage__* *de vitesse variable*.
(15d) *A new diamond* *__setting__* *technique has been developed.*
(15d') *Une nouvelle technique de* *__positionnement__* *de diamant a été développée.*

- __Re-labeling of *should, provided*, etc. at beginning of declarative sentences__. For example:
  (16) *__Should__ the situation call for such action, we are prepared to act*.
  (16') *__Si__ la situation réclame telle action, nous sommes prêts à agir*.

- __Analysis of *as*.__ The form *as* is in the dictionary only as a **CJ(SUB)**, it being left to source analysis to determine its exact grammatical function, especially when used to introduce a non-lexicalizable phrase. A great many such phrases are handled by **Parse1**, e.g., *as a whole*; *as* **AV** *as***;** *as* **AJ** *as*, etc**.** **Parse1** analysis concatenates these phrases, labels them as appropriate (**ADV, PP, DET, CJ(SUB))**, and affords targets opportunity to override the default lexical transfer. For example:
  (17a) *as of April* → *ab April*
  (17b) *as few as five* → *nur fünf*
  (17c) *as many as five* → *bis zu fünf*
  (17d) *as to …* → *Im Bezug auf …*
  (17e) *as long as possible* → *so lange wie möglich*
  (17f) *as long as you insist, I will come* → *solange Sie bestehen, komme ich.*
  (17g) *four times as fast* → *viermal so schnell*

- __Analysis of *any*__. Translation for *any* commonly depends on analysis of the entire clause, as illustrated in the following:
  (18a) *I do not have any book. Any book will suffice.*
  (18a') *Je n'ai pas de livre. Tout livre suffira.*
  (18a'') *Ich habe kein Buch. Jedes Buch genügt.*

  The context available to **Parse1** is limited and **Parse1** translation for *any* can often be wrong, despite the top-down picture afforded by the macro parse. In (18b), we see that the **Parse1** transfer for *any* has necessarily been overridden by subsequent pipeline analysis.
  (18b) *If you do not wish to have* *__any__* *supper tonight, please tell the cook.*
  (18b') *Si vous ne souhaitez pas dîner ce soir, veuillez dire au cuisinier.*

  Deficiency in target work can also be a source of error, as the French in (18c'). Note that the German in (18c'') treats *any* more correctly.
  (18c) *Do you want* *__any__* *supper tonight?*
  (18c') **Voulez-vous* *__tout__* *dîner ce soir?*
  (18c'') *Wollten Sie* *__kein__* *Abendessen heute abend?*

- __Reversing a parsing decision made by **Res**__. Though infrequently invoked, **Parse1** possesses the means to reverse **Res**, usually regarding **N/V** homograph resolution. To this limited extent, the pipeline parse is not purely deterministic.

See Fig. 13 for a graphic illustration of **Parse1** parsing.

- **Parse2 -** The second **Parse** module builds upon the output of **Parse1**. Among its principal tasks are:
  - Detecting, and effecting connominal **PP** attachments to **NP.** Here the **Parse2** rule generally has the form shown below, where **x** is any preposition or class of prepositions.

    SAL Pattern: **NP(GOVx) PRP(x) NP**

    Constraints: (i) The verb of the clause, if one has occurred, has not made a strong prior governance claim on **PRP**; (ii) the **NP** governs the **PRP**.

    Actions: (i) Preposition is labeled *connominal*.
    (ii) prepositional phrase is attached to **NP** at left.

    Targets: Targets typically alter lexical transfer for the **PRP** at this point, as in (19b'), below.

    In (19a'), below, notice that the gender of the German relative pronoun *die* refers to *revulsion* rather than *money*, reflecting a probabilistic assumption regarding which noun to attach it to.

    (19a) *He has a revulsion to money that cannot be overcome.*
    (19a') *Er hat eine Revulsion zu Geld, die nicht überwunden werden kann.*
    (19b) *He wrote a book about the room.*
    (19b') *Er schrieb ein Buch über den Raum.*
    (19c) *He placed some flowers about the room.*
    (19c') *Er stellte einige Blumen um den Raum.*

    In (19c), above, the prepositional phrase *about the room* is *not* seen as connominal (to *flowers*) in **Parse2.** Rather, the converbal attachment of the **PP** and the target transfer *um* are effected in **Parse3**, as seen in (19c'),

  - Extraction of relative clauses, embedded absolute constructions, parenthetical material, etc. In all cases, these are removed from the clause in which they appear, leaving behind a trace marker. Extracted materials are processed through **Parse2-Parse4** as separate sentences. (See Fig. 14 for graphic illustration of clause extraction.)

  - Discrimination between relative clauses and th-clauses. For example:

    (20a) *The hope that he had was very strong.*
    (20a') *Die Hoffnung, die er hatte, war sehr stark.*
    (20b) *The hope that he would win the race was very strong.*
    (20b') *Die Hoffnung, dass er das Rennen gewinnen würde, war sehr stark.*

  See Fig. 14 for a graphic illustration of **Parse2** parsing.

- **Parse3 -** The third **Parse** module builds upon the output of **Parse2**. It will be noted in what follows that much of the work in **Parse3** is accomplished via interaction with **Semtab**. Among phenomena dealt with in **Parse3** are:

  - Verbs and non-contiguous verb particles. The rule abstracted below is typical: a SAL pattern consisting of an undifferentiated verb (**V**) stretching (**∗**) over any number of non-verbal clause elements to a particle (**PART**) which is following immediately by clausal punctuation (**PUNC(CB)**). Note that up to this point analysis has not yet connected the verb and particle.

SAL Pattern: **V(u;u) ∗ PART PUNC(CB)**
Constraints: None.
Action: (i) Send Pattern to **Semantic Table** for match;
(ii) re-label **PART**, and backspace all the way for next match (re-labeling forestalls rematching).
**Semtab** Action: (i) If match on **V PART** is found, verb and particle are re-labeled with appropriate new SAL code.
Target Action: Target assigns new transfer to re-labeled verb (this is also done in action portion of **Semtab** rule).

In (21) and (22), below, we see the discriminating power of the **Parse3/Semantic Table** interaction, in particular, and of the pipeline parsing strategy in general. Sentence (21) is artificial but serves to illustrate **Parse3** analysis.

(21) *Please take the cover of the box, put on by my brother, off.*
(21’) *Veuillez retirer la couverture de la boîte, mise par mon frère.*
(21”) *Nehmen Sie bitte die Haube der Kiste, die von meinem Bruder angeschaltet wird, ab.*

Below are further examples of **Parse3/Semtab** interaction, all initiated by the single **Parse3** rule abstracted above:

(22) *Take the unit down, take the unit out, take the unit away, take the unit apart, take the work on, take the items back.*
(22’) *Baissez l'unité, sortez l'unité, retirez l'unité, démantelez l'unité, entreprenez le travail, reprenez les articles.*

- Verb argument structure analysis and labeling. The **Parse3** rule effecting the analysis and transformations in the examples below is even simpler:

SAL Pattern: **V(u;u) ∗ PUNC(CB)**
Constraints: None
Action: (i) Send Pattern, including all elements covered by Kleene star, to **Semtab** for match; (ii) inhibit this rule and backspace all the way for next match.
**Semtab** Action: (i) If match on **V** + verb arguments occurs, constituents are re-labeled where appropriate.
Target Action: Target may assign new transfer to verb, converbal preposition, object of verb, subject, etc. (also done in action portion of **Semtab** rule).

The following illustrate typical **Parse3/Semtab** interactions invoked by the above **Parse3** rule and the sort of target work that **Semtab** can effect.

(23a) *Please let me know the result.*
(23a’) *Lassen Sie mich bitte das Ergebnis wissen.*
(23b) *Please let me have the book.*
(23b’) *Erlauben Sie mir bitte, das Buch zu haben.*

- **NP** series concatenation. Rule constraints in series rules allow match only where commas and coordinating conjunctions are *non*-clausal, as established by the earlier **Res** macro parse. For example:

SAL Pattern**:** **NP(u;u) PUNC(COM) NP(u;u) CJ(CRD) NP(u;u)**
Constraints: **PUNC** and **CJ** (conjunction) are not clause boundaries.
Actions: (i) Concatenate as **NP,** using SAL code of **NP1;** (ii) re-label last **NP** with end of series marker; (iii) backspace 1 element.
Targets: (i) Chain articles and preposition series (for French);

(ii) Make case assignments (for German), etc.

The unedited French translation (24') of sentence (24) well illustrates the effect of this **Parse3** rule, and a shorter **Parse3** rule just like it:

(24) *My father gave his house, car and boat jointly to his sons and daughters.*

(24') *Mon père a donné sa maison, sa voiture et son bateau en commun à ses fils et à ses filles.*

- Analysis and concatenation of hitherto unattached prepositional phrases as adverbial **PP**. Adverbial **PP's** are labeled **PP(SENT)**, **PP(LOC), PP(MANNER)**, etc., according to **PRP** and **NP** Type combination.

- Interclausal chaining. The following depicts a **Parse3** rule designed to handle **ING** series in dependent clause, whether chained by comma or coordinating conjunction. Labeling of the **ING** form in the first dependent clause is communicated to **ING** forms of the subsequent clauses. Note that the one rule handles **ING** series of any length, as illustrated below.

SAL Pattern: **V(u;ING) ∗ PUNC(COM) /or/ CJ(CRD) V(u;ING)**

Constraints: **PUNC** or **CJ** must be clause boundaries.

Actions: (i) Previous re-labeling of 1st **ING**, denoting presence of a subordinate clause, is communicated to 2nd**;** (ii) backspace all the way.

Targets: Effect series chaining for **ING** verb forms.

Comments: (i) Rule applies to series chained either by commas or coordinating conjunctions; (ii) re-labeling and backspacing all the way is typical of **Parse3** rules. Backspacing allows for multiple looks at a given pattern, each for a different purpose (as in (26)).

The unedited French (25') illustrates the effect of the above **Parse3** rule:

*(25) When operating the unit, debugging the unit or performing maintenance tests, be sure to check power levels.*

(25') *En commandant l'unité, en mettant au point l'unité ou en effectuant les essais d'entretien, soyez sûr de vérifier les niveaux de pouvoir.*

In (26'), below, we see the joint effect of the two very different **Parse3** rules associated with (22) and (25) respectively, executed sequentially over the same pattern. (Note gender error in pronominal reference):

(26) *When putting the cover on, taking it off, or adjusting it in any way, be sure the power is off.*

(26') *En mettant la couverture, en le retirant ou en l'ajustant de toute façon, assurez-vous que le pouvoir est coupé.*

See Fig. 16 for a graphic illustration of **Parse3** parsing.

- **Parse4 -** The final **Parse** module builds upon the output of **Parse3.** All sentence elements have now been reduced to just five SAL entities: **NP, PP, AUX, V, PUNC,** plus place markers for materials extracted earlier in **Parse2**. Earlier modules have already analyzed and labeled constituents for *intraclausal* grammatical function (e.g., re-labeling **NP** for subject, object, indirect object, etc.). Now, a principal work of **Parse4** is verb tense analysis, both intra- and interclausal.

Functions include:

- Intraclausal tense assignments: Punctuation (**PUNC**) and subordinating conjunctions (**CJ(SUB)**) have been concatenated and re-labeled in **Parse3**. These re-labeled elements now trigger tense assignments in **Parse4**.
  - *(27) Unless he receives instructions to the contrary, he is going to go home.*
  - (27') *À moins qu'il ne reçoive les instructions au contraire, il va aller à la maison.*
- Interclausal verb tense/mood coordination. In the following, the treatment of the complementary infinitive clause is a function of verb SAL Type and verb tense in the principal clause.
  - (28a) *They want John to do it*
  - (28a'*) Ils veulent que John le fasse.*
  - *(28b) They do not want him to succeed.*
  - *(28b') Sie wollen nicht, dass er Erfolg hat.*
  - *(28c) They did not want him to succeed.*
  - *(28c') Sie wollten nicht, dass er Erfolg hatte.*

- Pronoun resolution. Without extra-sentential processing, pronoun resolution obviously remains a weak area. Handling is somewhat better when the antecedent is intrasentential, as in (29):
  - (29) *We buy a house that catches our fancy and then resell it the next year.*
  - (29') *Nous achetons une maison qui attrape notre fantaisie et ensuite la revendons l'année prochaine.*

  But, even here, overly complex structures typically cause the pronoun/antecedent relationship to be lost, as in (30):
  - (30) *We buy a house that catches our fancy and live in it until we are tired of it, and then we sell it again, usually within a few years.*
  - (30') *Nous achetons une maison qui attrape notre fantaisie et vivons dans elle jusqu'à ce que nous soyons fatigués de lui, et ensuite nous le vendons de nouveau, d'habitude dans quelques années.*

The above functions notwithstanding, the principal work of **Parse4** is directed toward support of targets. It accomplishes this by presenting these abstract source constituents (**NP, VP, AUX**, **PUNC**), one by one, to linked target rules. Target rules load these constituents in the appropriate slots of a final, high-level target template for each clause and finally for the sentence as a whole. At this point, everything that must be known about the source string is now available to the target, thus allowing targets to place the by now virtually meta-linguistic constituents into the desired target order, effecting such stylistic transformations as is deemed appropriate and possible. At the end of **Parse4** source analysis, target template slots are unloaded and the resultant target parse (bracketed target string) is input to **Tgt Gen**, for generation of literal target output.

See Fig. 17 for a graphic illustration of **Parse4** parsing

### 5.0 Target Transfer and Generation

In this paper we have focused almost exclusively on source analysis, reflecting a belief that the power to *decode* source is the more fundamental and more difficult aspect of MT. Target work is far from trivial, however, and obviously poses its own set of challenges and difficulties, treatment of which requires appropriate skill. But whenever Logos developers responsible for target work are asked what is most needed to improve translation quality, invariably the answer has to do with improving source analysis.

### 5.1 TARGET RULES

Sets of target rules make up the **Tran** modules of the pipeline. Target rules presuppose the source rules to which they are linked and therefore do not have a SAL pattern component of their own. In other respects they resemble source rules: i.e., they have their own variety of constraint conditions and, most fundamentally, an action component. A target rule can perform additional source analysis of a nested kind when needed by a particular target language, but the principal actions concern contrastive morphological, syntactic and semantic transfer of the elements comprising the source-rule SAL pattern. All target actions in the **Parse/Trans** pipeline take the form of symbolic target parse-tree notations. These notations subsequently drive actions on literal strings in the final generation phrase of machine translation, performed by the **Tgt Gen** module.

### 5.2 TARGET TRANSFER

As evident in Fig. 2, transfer is effected incrementally, in compositional fashion, at the point where each source constituent is considered to have been analyzed in full (consistent with the system's capabilities). Thus, for example, when a descriptive adjective in **Parse1** has been analyzed in relationship to the noun it modifies, each target will make decisions regarding (a) its transfer (e.g. whether to use the default lexical transfer or to overlay it with a transfer derived from **Semtab**), and (b) its syntactic placement in the target language. Target rules in **Parse4** will decide how and where each constituent of the source clause is to be transferred in the target (e.g., where to position a prepositional phrase analyzed as an adverb of time). Template slots in a clause-level target template are thus gradually filled and ultimately arranged in target sentence order, the target equivalent (*mutatis mutandi*) to constituents directly below **S** in the source parse. At the end of the analysis pipeline, all slots are then unloaded and the output (now essentially an ordered string of pointers to target words in the lexicon, with annotation for morphology) is passed on to the **Tgt Gen** module which, as we have said, then takes this data and synthesizes the literal target sentence.

### 5.3 STYLISTIC TRANSFORMATIONS

Things expressed one way in the source are often not expressible that way in the target, calling for syntactic transformations of various kinds. To a limited but not insignificant degree, the Logos Model supports the requirements of proper target style, as the examples in (31), (32), and (33) illustrate (showing unedited output). Transformations are the result of rule interaction throughout the pipeline, but chiefly to source/target rule interaction in **Parse4/Tran4**.

(31) *The situation was alluded to by my friend in his letter.*
(31') *Mon ami a fait allusion à la situation dans sa lettre.*
(32) *The situation was alluded to in their letter.*
(32') *On a fait allusion à la situation dans leur lettre.*

In (33), below, we see French and German output with radically different stylistic treatment, both from each other and from the English original. This shows the limited but real extent to which target linguists working with this Model have been able to overcome the so-called "structure preserving" tendencies said to be inherent in MT (Somers, 1992/3). In particular, note how the English ellipsis in (33) "*… and their information input directly…*" is handled by each of the targets. Note also the German 'subjectless clause' treatment of the main clause in (33''). All output is unedited.

(33) *Other forms of storage media, such as magnetic cards and computer tape, can also be accessed through optional devices, <u>and their information input directly</u> to the system.*

(33') *On peut également accéder à d'autres formes du support d'information, comme les cartes magnétiques et la bande pour ordinateur, par des appareils facultatifs <u>et on peut introduire leur information directement</u> dans le système.*

(33'') *Auf andere Speichermedienarten, wie magnetische Karten und Magnetband kann auch durch beliebige Geräte zugegriffen werden und <u>ihre Informationen können direkt in das System eingegeben werden.</u>*

**6.0 How Do You Deal with Complexity Issues?**

We now need to review our original question regarding complexity. Complexity effects in MT concern two issues: system performance and system improvability (a function, we argue, of system maintainability). While performance will always remain an issue, steady increases in raw computer power tend to make system performance a secondary matter. Far more critical is the system improvability and maintainability issue. How strong can an MT system become? Can a system, for example, handle the many thousands of content-sensitive verb transfers to be found in any good, bi-lingual desk-top dictionary? If not, is it because of complexity effects posed when trying to make effective use of such quantities of data? The requisite linguistic knowledge is certainly available: what remains at issue is a computational approach able to deal with it.

To illustrate this complexity problem more concretely, consider what is involved in correctly translating the English word *as.* In (34), below, we see five different senses of this word, each triggered by a variety of contextual clues, and each with its own implication for German. How is such context to be specified, where are these specifications to be stored, and how are they to be applied? Can one burden a lexical entry for *as* with logic of the complexity needed in order to handle examples such as these? Can a single rule be written to cope with such phenomena? At what point in the parsing process would this lengthy rule be applied? How efficient and effective would it be? How maintainable? More than likely, because of the difficulties involved, such phenomena will simply not be dealt with beyond a certain point. This barrier typifies what we mean by complexity and how complexity limits machine translation.

German translations below are unedited output of the current commercial E-G system:

(34a) ***As*** *you can see, he is sick.*
(34a') ***Wie*** *Sie sehen können, ist er krank.* .
(34b) ***As*** *he is sick, we cannot ask him to work.*
(34b') ***Weil*** *er krank ist, können wir ihn nicht bitten, zu arbeiten.*
(34c) ***As*** *he was being given his medicine, he began to choke.*
(34c') ***Während*** *ihm seine Medizin gegeben wurde, fing er an, zu ersticken*
(34d) ***As*** *he began to recover his health, he realized that his wife had stood by him through difficult times.*
(34d') ***Als*** *er anfing, seine Gesundheit zurückzubekommen, erkannte er, dass seine Frau ihm durch schwere Zeiten beigestanden hatte.*
(34e) ***As*** *a patient, he was very cooperative.*
(34e') ***Als*** *Patient war er sehr kooperativ.*

There are 80 patterns (rules) indexed on *as* in **Parse1**, 52 in **Parse2**, 5 in **Parse3**, and 11 in **Parse4**. All of the sentences above were dealt with at various times along the pipeline by one or more of these rules (See additional treatment of *as* in discussion of **Parse1** in 4.3.2). The examples above were chosen because they are handled relatively successfully. It is quite easy to find other *as* sentences that translate poorly, and that would require additional rules somewhere in the pipeline.

We have argued in this paper that, for all the obvious importance of linguistics, it is the computational approach that will ultimately determine how good an MT system will be--the computational approach regarding representation, storage, and rule application. We have described an approach which we feel copes optimally with these three fundamentals. In what follows, we focus more narrowly on the Model's computational methodology relating specifically to the question of rule application, viz., how an exceedingly rich knowledge store is to be applied, effectively and efficiently, to an unconstrained input stream without giving rise to complexity effects. It is here, perhaps, that the Model we are discussing becomes most novel.