Meta
Journal des traducteurs
Translators' Journal

Meta

Simplifying the Complexity of Machine Translation

Randall Sharp and Oliver Streiter

Volume 37, Number 4, décembre 1992

Études et recherches en traductique / Studies and Researches in Machine Translation

URI: https://id.erudit.org/iderudit/004127ar
DOI: https://doi.org/10.7202/004127ar

See table of contents

Publisher(s)

Les Presses de l'Université de Montréal

ISSN

0026-0452 (print)
1492-1421 (digital)

Explore this journal

Cite this article

Sharp, R. & Streiter, O. (1992). Simplifying the Complexity of Machine Translation. *Meta*, *37*(4), 681–692. https://doi.org/10.7202/004127ar

Article abstract

A description of the CAT2 machine translation system is presented as an example of a model which stresses simplicity over complexity. The design of the formalism encompasses a minimum of formal constructs, yet is powerful enough to describe complex linguistic and translational phenomena. The paper presents an overview of the formalism, followed by examples of its usage in linguistic applications. A critical evaluation is presented, in which the authors discuss the shortcomings of the system and present the directions that are being taken to achieve a more realistic, and more simplified, model of machine translation.

SIMPLIFYING THE COMPLEXITY OF MACHINE TRANSLATION

RANDALL SHARP, OLIVER STREITER
IAI, Saarbrücken, Germany

Résumé

Le prototype d'un système de traduction automatique CAT2 est présenté comme exemple d'un système qui met l'accent sur la notion de simplicité. Les quelques bases formelles du formalisme sont expliquées et illustrées par des applications linguistiques. Au moyen de quelques exemples, nous montrons comment un tel formalisme permet avec des moyens très simples de réaliser des grammaires aptes à traiter des phénomènes complexes et linguistiquement intéressants. Sont également esquissés et présentés les développements attendus du formalisme actuel.

Abstract

A description of the CAT2 machine translation system is presented as an example of a model which stresses simplicity over complexity. The design of the formalism encompasses a minimum of formal constructs, yet is powerful enough to describe complex linguistic and translational phenomena. The paper presents an overview of the formalism, followed by examples of its usage in linguistic applications. A critical evaluation is presented, in which the authors discuss the shortcomings of the system and present the directions that are being taken to achieve a more realistic, and more simplified, model of machine translation.

In developing any natural language processing (NLP) system, particularly for machine translation (MT) applications, it is crucial that the basic constructs of the system become, and remain, simple. Otherwise the complexity of the system becomes overwhelming (1) for the system developers, who have to maintain the system over its lifetime, (2) for the linguists and translators who maintain the grammars, lexicons, and translation dictionaries, and (3) for the end-users who actually make use of the system in some application. The notion of simplicity is often at odds with notions of power, expressiveness, efficiency, which one also obviously needs but which seem to promote complexity, and often, always unwittingly, metamorphoses into complicatedness, where the internal and external workings of the system are no longer well understood. This tension between simplicity and appropriate complexity underscores the design and development of an experimental MT system known as CAT2.

CAT2, a transfer-based MT system, was first developed in 1987 as a sideline to Eurotra, the massive MT project sponsored by the CEC in Luxembourg encompassing nine languages in twelve countries. In 1985 the CEC sponsored a first design based on the <C,A>,T framework (Arnold *et al.* 1985, 1986), in which a "rapid prototyping" approach was advocated. However, given the scope of the undertaking and the intense and varied use to which the prototype was subjected, its inadequacy as a working system soon became apparent. In 1987, the CEC cancelled this attempt and developed the Engineering Framework (Bech and Nygaard 1988) which has been in development up to the present time. Although some of the shortcomings of the original prototype were overcome, the "E-Framework" also introduced a number of new complexities into the framework (see Sharp 1991 for a review), to the point where the CEC has once again

undertaken a new study to develop a third, even more powerful, prototype (Alshawi *et al.* 1991), scheduled for completion in 1994. In the meantime, a number of sideline alternatives have sprung up in order to continue the research into machine translation, among them the MiMo system (Arnold and Sadler 1990), CLG (Balari *et al.* 1990), and the CAT2 system (Sharp 1988).

The CAT2 system, adhering to the original <C,A>,T methodology and abiding by the maxim of controlled simplicity, has continued to develop into a viable MT prototype in which current theories of computation, linguistics, computational linguistics, and translation are being tested, refined and applied. To date, experimental translation systems have been developed for English, French, German, Spanish, Greek, Portuguese, Russian, Czech, and Japanese.

In this article we will look at the nature of the CAT2 formalism, and try to reflect the notion of simplicity behind its design. The first section briefly describes the formalism, the second illustrates its application, and the third covers some of the necessary extensions to the formalism which will further enhance and underline its inherent simplicity.

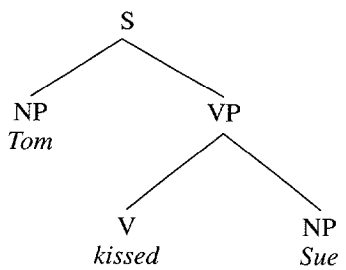
1. OVERVIEW OF THE CAT2 FORMALISM

The CAT2 formalism is used to describe (1) the grammar of a language which defines the set of well-formed linguistic structures that belong to that language, and (2) a mapping relation between the linguistic structures of one language and those of another. To this end, the formalism consists of descriptive mechanisms for generating the linguistic structures and for translating from one structure to another. The former are called Generators, the latter Translators. (This terminology is taken from the original <C,A>,T specifications.)

1.1. GENERATORS

Generators describe linguistic structures in terms of trees. We will assume here an intuitive understanding of the notion of a tree, *i.e.* that of a root (mother) node dominating zero or more subtrees (daughters). Each daughter has a unique mother node, except the topmost node, which has no mother node. Furthermore, we will assume that the daughters under a root are ordered, so we can speak of a left branch, right branch, middle branch, etc. Finally, a root with no daughters is a terminal node, a leaf in the tree.

An example of a typical tree structure is the following:



This tree shows only the syntactic categories, but this is only one piece of linguistic information. Other properties, such as person, number, gender, tense, and many others, would be required to fully describe the actual properties pertaining to a given construction. As is now standard in modern computational linguistic theories these properties are described as **features**, *i.e.* attribute-value pairs; each node in the tree contains a set of

such features, called a **feature bundle**. For example, the subject NP node above might be partially described by the following feature bundle:

```
{ cat = np
  ortho = 'Tom'
  lex = tom
  agr = { per = 3
          num = sing
          gen = masc }
  case = nom }
```

A feature may be **simple**, *i.e.* it has an atomic constant as its value, *e.g.* np, 'Tom', etc., or **complex**, *i.e.* it has another feature bundle as its value, *e.g.* the agr feature above. A third possibility is that the value is a variable, bound to some other variable within the tree. When one variable becomes instantiated to some value via rule application, all other variables bound to it become identically instantiated. We will see the power of this in section 2.

Generators consist of two types of rules: (1) structure-building rules (called "b-rules"), and (2) feature instantiation rules ("f-rules"). The former are used to describe structural relations, *i.e.* trees, and the latter fill out the feature content within the nodes of the tree by a process of *unification* (described below); they essentially are used to embody the well-formedness principles found in linguistic theories.

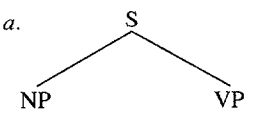
1.1.1. B-RULES

B-rules define partial tree structures, *i.e.* mother nodes and their immediate daughters. A b-rule has the following form:

ROOT.BODY

where ROOT describes the root of the tree, and BODY is a list of the immediate daughters under the root. By restricting the tree description to the *immediate* constituents, we have the equivalent of a context-free grammar. That is, the structure shown in (*a*) corresponds to the context-free rewrite rule in (*b*), which is written in CAT2 notation as (*c*):

a.



b. S —> NP VP *c.* {cat=s}.[{cat=np}, {cat=vp}].

The notation is enriched to allow for specifying optional constituents within the body of the rule, and for alternations of constituents. This makes it possible to collapse many rules into a single rule, with a corresponding gain in efficiency and perspicuity of the grammar.

In the case of a terminal node, BODY is the empty list; such rules, called **atomic b-rules** or **atoms** for short, define extensionally the lexicon of the generator, *i.e.* those constituents which cannot be further decomposed. A sample atom for the verb *kissed* might look as follows:

{cat=v,lex=kiss,ortho=kissed,tense=past,...}.[]

A more complete entry would include argument structure, Aktionsart, aspect, inflection, possibly phonological information if relevant, etc. We will look at some of these in more detail in section 2.

1.1.2. F-RULES

F-rules operate on the partial trees generated by b-rules and are used to enforce the well-formedness of tree and feature structures. They have a form similar to b-rules, except they state feature condition-consequence relations, effectively as *if-then* statements. That is, they describe the condition(s) under which some features are required or prohibited in a given structure. Also, unlike b-rules, they are not restricted to describing just the immediate daughters; they can map onto an arbitrarily deep tree structure.

For example, the following is an f-rule that links the index of an anaphor with that of the subject (assuming the example tree given above):

{}.[{cat=np,index=I}, {}.[*,{cat=np,type=anaphor}>>{index=I},*].

This rule states that, given any structure which contains an NP and an embedded NP whose type is ANAPHOR, the latter's index must equal that of the former. The "*"s indicate that anything (or nothing) may precede or follow the anaphor. All feature bundle descriptions to the right of the ">>" symbol represent the consequent of the rule, and everything else represents the conditions under which the consequent must be valid.

In practice, f-rules are used quite extensively, since the set of phrase structure rules, *i.e.* b-rules, is rather minimal, given the implementation of X-bar theory as outlined in section 2. All well-formedness conditions in a CAT2 grammar, then, are described by a few structural b-rules, and several f-rules; no other rule types exist within the formalism.

1.1.3. RULE APPLICATION

CAT2 belongs to the family of unification-based grammar formalisms such as PATR (Shieber 1984). This means the basic operation is that of unification (Shieber 1986), both of tree and feature structures. As an illustration, assume we have the feature bundle description in (*a*) and this is to be unified with the feature bundle in (*b*):

a. {wh=no,agr={per=3,num=sing}}
b. {cat=n,ortho='Sue',agr={per=3,gen=fem},wh=X}
c. {cat=n,ortho='Sue',agr={per=3,gen=fem,num=sing},wh=no}
d. {cat=n,agr={per=3,num=plu}}

The result of unification is the feature bundle in (*c*), where the variable X has been unified with the value no, and the value of agr is extended to include {num=sing}. (*a*) cannot unify with (*d*) because of conflicting values for num.

The notion of unification has been generalized to that of **constraint satisfaction**, in which feature descriptions state positive, negative, or disjunctive constraints on values. For example, the following illustrate typical usages of these constraints:

a. positive constraint	{case=nom}
b. negative constraint	{case"=gen}
c. disjunctive constraint	{case=(dat;acc)}

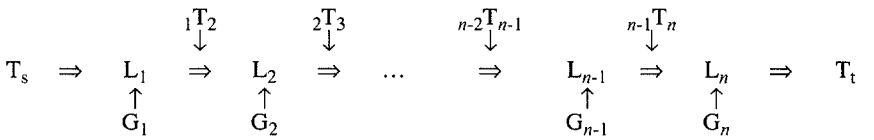
If a given feature has no value at the time a negative or disjunctive constraint is applied to it, or, in the case of complex features, if its value is not sufficiently instantiated to resolve the constraint, the evaluation is postponed until such time as more information becomes available. This form of "lazy evaluation" greatly improves the efficiency, since unnecessary and costly processing may thereby be avoided.

The application of f-rules to a structure is carried out in the order they occur in the grammar. This ordered relation enables one to define default values to features after other f-rules have covered the explicit cases. This adds a procedural element to an otherwise

fully declarative grammar formalism, meaning the linguist must be cognizant of the correct order in which rules are to be applied. At the same time, the linguist can, for example, order filtering rules earlier in order to reject invalid structures before other rules are unnecessarily applied.

1.2. TRANSLATORS

Translators map one structure onto another by a recursive process of decomposition, transfer and recomposition. For example, a tree structure created by the parser is transformed into a new structure with possibly differing feature structures. The new structure may reflect another aspect of the language, *e.g.* a semantic or pragmatic representation, or it may be a representation appropriate to a target language. Schematically, we have the following situation where a source text T_s is translated to a target text T_t by transforming it through a series of intermediate structures L_i, where each L_i is defined by a generator G_i:



Translators consist of tb-rules and tf-rules, analogous to b-rules and f-rules, respectively. That is, tb-rules translate structures, and tf-rules translate features.

Syntactically, the tb-rule and tf-rule have the form:

$$\text{ROOT.BODY} \Rightarrow \text{ROOT.BODY}$$

where the lefthand side of the "⇒" symbol is an arbitrarily deep partial description of a source structure and the righthand side is an arbitrarily deep partial description of the target structure to be created. Constituents within the lefthand side BODY may be marked for recursive translation and repositioned within the righthand side BODY. Additionally, complex structures may be mapped onto atoms, and vice versa.

The simplest tb-rule expresses the relation holding between two atomic objects, as found, for example, in a French-to-English translator:

```
{lex=aller}.[] => {lex=go}.[]
```

A more complex tb-rule looks at deeper structures. For example, the following rule selects certain constituents on the lefthand side for recursive translation and rearranges the results on the righthand side:

```
{}.[ s, {}.[ v, *o ]] => {}.[ v, s, o ]
```

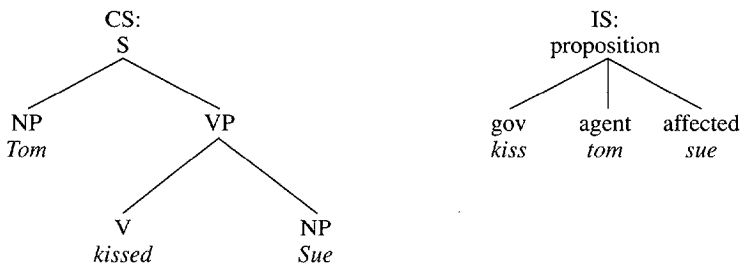
Such a flattening rule is conventionally used to convert a syntactic tree structure into a dependency tree, where the governor and its dependents are all on the same structural level.

2. LINGUISTIC APPLICATION

The grammars currently being tested are comprised of two generators, CS (Constituent Structure) and IS (Interface Structure). The CS generator is used to express the morpho-syntactic structure of a sentence, while the IS generator abstracts away from language-specific structural relations and expresses more semantically based relationships, such

as predicate-argument-adjunct structure, NP determination and quantification, sentential time reference, etc. In analysis, a translator CS ⇒ IS transforms the CS to IS, and in synthesis, an IS ⇒ CS translator performs the reverse transformation.

For example, the following illustrates the CS and IS tree structures that are generated from the sentence *Tom kissed Sue*:



The actual translation step from one language to another takes place between the IS of a source language and the IS of a target language, as defined by an $IS_s \Rightarrow IS_t$ translator. An ongoing debate within machine translation theory centers around the feasibility and advisability of making this relation bidirectional; as yet, we have left this a unidirectional relation in CAT2 pending further research. The IS is so designed that the translation process from IS to IS is maximally simplified. Tree structures are maintained unchanged so long as the predicate-argument-adjunct relations are invariant, and feature structures are copied over intact except where explicitly altered, for example in the lexical values.

This correspondence between tree and feature structures across languages at the IS level allows a major part of the IS description to be actually shared among the different language components, so that the main language-specific component is the lexicon. This means that, for example, the German IS generator consists of the German IS lexicon and the shared IS component, the French IS generator consists of the French IS lexicon and the shared IS component, etc. Furthermore, shared "concepts", such as terminological entries, geographical place names, months, days of the week, currency names, etc., are included in a shared IS lexicon, further reducing the language-specific lexicons. This action eliminates a major portion of the redundancy encountered in multilingual language descriptions, promoting linguistic consistency and stability in the development process, while still maintaining a transfer-based approach to machine translation.

As a consequence of this grammar sharing, a small number of language-independent (*i.e.* "universal") b-rules determine phrasal structure based on language-specific lexical information. The main principles determining grammatical structure are reflected in three complex features found in the lexicon: the HEAD, FRAME, and RESTR features. We look at the definition of these features in the following section.

2.1. PHRASE STRUCTURE

Three b-rules from the shared IS generator are responsible for describing tree structures. One describes the projection of a head including its arguments and adjuncts; a second describes the coordination of phrasal structures, and a third describes support verb (or "light" verb (Grimshaw and Mester 1988)) constructions, in which the principal argument-bearing element is a predicative nominal, adjective, or adverbial, the verb playing only a supporting role and is lexically specified by the predicative element; examples include:

French: *faire un miracle*
German: *ein Wunder vollbringen*
English: *perform a miracle*

For reasons of space, we will concentrate on the first of these rules, the head projection rule; a discussion of the implementation of the support verb construction in CAT2 is covered in Mesli (1991).

The head projection rule is given below; it instantiates simultaneously a number of linguistic principles:

```
{head=HEAD}
   .[ {head=HEAD,frame={arg1=ARG1,arg2=ARG2,arg3=ARG3}},
     ^ARG1, ^ARG2, ^ARG3,
     *{head={restr=HEAD}}]
```

The first constituent, whose presence is obligatory, is designated as the head of the construction, and its head features {head=HEAD} are percolated to the mother node. The head acts as governor of the arguments described in its FRAME feature. The arguments, ARG1, ARG2, and ARG3, occur as optional sisters to the head, as do adjuncts. The head of each adjunct places restrictions ({restr=HEAD}) on its governor. In this one rule, we can express the interdependency between heads, their arguments and their adjuncts; structural well-formedness is determined solely by unification.

2.1.1. HEAD

Our notion of "head" corresponds basically to that described in the HPSG (Head-Driven Phrase Structure Grammar) formalism of Pollard and Sag (1987). According to their Head Feature Principle, those features marked in the lexicon as head features are percolated through all projections of the head to its maximal projection. These include the syntactic category, the ϕ features person, number, and gender, a referential index, a *wh*-feature, voice, tense, aspect, and several others.

In our implementation, we also assume that the functional categories D(eterminer) and C(omplementizer) form projections, taking an N and V projection, respectively, as arguments.

2.1.2. FRAME

The frame feature specifies the possible arguments of a lexical head. (We have arbitrarily allotted a maximum of three arguments; this is based purely on pragmatic considerations of implementation, and could be changed according to need.) An argument specification unifies directly with the features that actually appear in the constituent in question (*i.e.* they do not specify merely a mapping function), and include, among others, a thematic role (AGENT, AFFECTED, etc.), syntactic category, selectional restrictions (ANIMATE, CONCRETE, etc.), and sometimes lexical restrictions (*e.g. depend+on*).

An example of a frame specification is given below for the verb *adopter* as in *adopter une décision*:

```
{ lex=adopter,
  head={cat=v,stative=no,...},
  frame={arg1={role=processor,head={cat=d,semf={anim={type=hum}}}},
         arg2={role=phenom,head={cat=d,semf={abstract~=concr}}}}}.[]
```

The completeness of an argument structure is controlled by f-rules, which filter out any partial structure in which obligatory arguments are not realized. Obligatoriness or

optionality is indicated in the lexicon as an explicit feature; *e.g.* the verb *manger* assigns its second argument the feature {oblig=no}. In practice, all arguments to verbs are assigned the feature {oblig=yes} by default. The coherence of the structure is controlled by assigning a null argument structure by default, which is only overridden by explicit inclusion of arguments in the lexical entry.

2.1.3. RESTR

The RESTR (restriction) feature is taken from HPSG and refers to the restrictions that adjuncts have over the sorts of governors they may modify. Our rule allows for any number of adjuncts in any order, so long as their respective RESTR features unify with those of the head of the governor. The RESTR feature is a component of the adjunct's HEAD feature, so that by percolation it is available for unification with the governor's head.

For example, the French adverb *hier* contains a RESTR feature specifying that it adjoins to a V projection with certain time and aspect values. A sentence such as *Il vient hier* cannot succeed in unifying due to conflicting values of time. Similarly, the sentence *Il vient*, which is ambiguous as to present or future tense, would be disambiguated via the RESTR feature if followed by the word *demain*.

2.1.4. OTHER LEXICAL DEPENDENCIES

The power of unification is further exploited to express (1) control relations, (2) temporal dependencies between main and subordinate clauses, and (3) sharing of information between the head of a governor and the head of its argument.

As an example of subject control, we consider the verb *menacer*, which subcategorizes for three arguments, the "threatener", the "threatened", and the "threat":

```
{ lex=menacer,
  head={cat=v,...},
  frame={arg1={role=sender,head={cat=d,semf=SEMF,index=I,agr=AGR}},
         arg2={role=receiver,head={cat=d,semf={anim={type=hum}}}},
         arg3={role=message,head={cat=c,lex=de,
                                  ctl={semf=SEMF,index=I,agr=AGR}}}}}.[]
```

The "threat" can be realized as an infinitival clause, as in *Le patron l'a menacé de le renvoyer*, in which the subject of *renvoyer* is controlled by the subject of *menacer* via the feature CTL. A subordinate clause can only be accepted as an argument if the control features of the subject of *menacer* (SEMF, INDEX and AGR) unify with the control features percolated to the clausal projection through CTL.

Similarly, the FRAME and RESTR features can be bound together directly in the lexicon, as in a subordinator like *pendant que* whose clausal argument must share the same temporal relations as the main clause. This is illustrated below:

```
{ lex=pendant_que,
  head={cat=c,...,restr={event_speech=TEMP}},
  frame={arg1={head={cat=v,event_speech=TEMP}}}}.[]
```

This entry will then accept the sentence *Pendant que Paul lisait, Marie entra* and reject the sentence **Pendant que Paul lisait, Marie entre*, without recourse to any additional rules.

The correspondence between the head feature of a governor and the head feature of its argument is exemplified by the relationship between a preposition P and its DP (or NP) argument. The semantic features of a head noun are passed to those of the governing preposition, so that selectional restrictions of a predicate may be satisfied on the PP. This is coded in the lexical entry for the preposition as:

```
{head={cat=p,semf=SEMF},frame={arg1={head={cat=d,semf=SEMF}}}}.[]
```

Other examples include the relationships between functional categories and their lexical arguments, as in D — NP and C — VP constructions. For example, in the latter case, features of tense, voice and aspect are shared between the C and the head of the VP construction, using the same technique as above.

2.2. F-RULES AT IS

While the head projection rule controls not only structure but to a large extent the feature content within structure, it cannot adequately handle certain linguistically interesting phenomena, such as distinguishing between anaphoric, inherent, and middle-voice reflexives, or the determination of nominal specificity, so that the correct article, or absence of article, can be generated in the target language. For such purposes, f-rules are employed at the IS level. Here, we present one example of an f-rule which treats temporal boundedness in noun phrases.

Certain semantic values may be modified compositionally. The nouns *temps, time, Zeit* are lexically specified as temporally unbounded, so they may not appear with prepositions requiring temporally bounded arguments: **depuis temps*, **for time*, **seit Zeit.* However, if these nouns undergo quantification, they become bounded: *depuis quelque temps*, *for some time*, *seit einiger Zeit*. This modification of temporal boundedness takes place within the D — NP construction by way of an f-rule, given as:

```
{}.[ {head={cat=d,semf={temp={type=bound}}}},
     {head={cat=n,semf={temp={type=unbound}}}}.[*,{head={cat=quant}},*]]
```

Other f-rules are used to determine features of genericity, control the binding of relative clauses, pronouns, and anaphors to antecedents, state preferences of argument readings over adjunct readings in cases where ambiguity occurs, etc.

2.3. TRANSFER

The HEAD, FRAME, and RESTR features also serve to disambiguate the transfer process between languages. By using information stemming from adjunct restrictions, the correct translation from among several possible translations can be chosen, while still maintaining simple word-for-word translation rules. For example, the adverb *beaucoup* translates to the German adverb *sehr* when in the context of a stative verb, and to the adverb *viel* in the context of a dynamic verb:

Il l'aime beaucoup ⇒ *Er liebt sie sehr*
Il mange beaucoup ⇒*Er ißt viel*

This information is made available through the unification of the adverb's RESTR feature with the governing head of the clause.

The corresponding tb-rules are written as:

```
{lex=beaucoup,head={restr={stative=yes}}}.[] => {lex=sehr}.[]
{lex=beaucoup,head={restr={stative=no}}}.[] => {lex=viel}.[]
```

The Russian subordinating conjunction *kogda* is translated as *après que* or *pendant que*, depending on whether the subordinate clause carries a perfective or imperfective verb. As above, this is expressed in tb-rules, this time referring to the FRAME feature, here shown in simplified form:

```
{lex=kogda,frame={arg1={head={aspect=imperf}}}}.[] => {lex=pendant_que}.[]
{lex=kogda,frame={arg1={head={aspect=perf}}}}.[] => {lex=apres_que}.[]
```

3. EVALUATION

As the examples above have shown, the CAT2 formalism has succeeded in treating a number of interesting linguistic phenomena with the means of a simple but powerful formalism and carefully defined shared linguistic resources which follow general principles. But modification of the CAT2 prototype and the linguistic resources continues, and most of the grammatical specifications we gave above are likely to be modified in more or less radical ways.

The most obvious shortcoming of the translation methodology as depicted in section 1.2. is the stratificational model adopted by Eurotra and carried over into the CAT2 framework. This model posits multiple autonomous levels of representation for the grammar of each language, with translators linking each level. Former CAT2 (and current Eurotra) implementations defined three levels: (1) a syntactic structure, CS, (2) a functional structure, RS, and (3) a semantic structure, IS. This entailed that analysis at CS and RS invariably incurred overgeneration, often wildly excessive, until the IS level filtered out all semantically illicit readings. Our research has shown that at least the RS level is superfluous, and we have greatly simplified the processing by distributing the functional information over the CS and IS levels. This was the first step in reducing the inefficiencies engendered by stratification.

At the same time, a separate morphological level is being introduced in CAT2 grammars. The consequence of this move is to liberate the CS definition of morphological data, so that syntax and semantics can be integrated at a single level, *i.e.* CS, again with the obvious improvements in overall efficiency and simplicity.

The status of IS then comes into question. At present, the IS is designed to facilitate structural transfer between languages. However, transfer seems to have little to do with structure, once we look at the variety of languages and their forms of "translationally equivalent" expressions. Let us consider NP determination, as just one phenomenon among many that present structural differences cross-linguistically.

In an agglutinative language like Turkish, determination is indicated by morphological affixation, whereas in an inflectional language like French, determination is lexically realized:

(Turkish)		(French)
ev	⇒	*maison*
ev-im	⇒	*ma maison*
ev-ler	⇒	*les maisons*
ev-ler-im	⇒	*mes maisons*

Bulgarian also indicates determination morphologically, where the first [+N] element (noun, possessive pronoun or adjective) takes an affix:

(Bulgarian)		(English)
moliv	⇒	pencil
molivăt	⇒	the pencil
nobijat moliv	⇒	the new pencil
mojat nob moliv	⇒	my new pencil

More complicated is the situation in Classical Arabic, where the definiteness of a noun is marked by a genitive phrase, a possessive pronoun or the definite article *al*:

(Arabic)		(English)
al-baytu	⇒	the house
batyī	⇒	my house
batu raǧulin	⇒	the house of a man

As well, every adjective in the definite NP takes the definite article as does the noun:

al-baytu'l-kabīpu'l-wāsi'u ⇒ the big spacious house

Word order is another means to mark definiteness. In Mandarin Chinese, for example, a thematic NP in phrase-initial position cannot be indefinite if not specially marked by the operator *yu*, which gives it an existential *there* reading. Similarly in Russian, an NP to the left of the verb must be regarded as definite if no indefinite pronoun of the type *odin, kto-nibud', ktolibo, kto-to, koe-to* is present.

Up to now, the approach taken has been to convert the source language structures into a uniform abstract structure, the IS, from which transfer is carried out compositionally, *i.e.* by recursively transferring the subparts of the structure. The translated IS is then converted to a target structure corresponding to the target language. The difficulty with this approach is two-fold: (1) to design an appropriate "interlingua"-like IS structure, and (2) to transform language-specific structures to this uniform IS in analysis and from the IS in synthesis. As to the first difficulty, this entails not only establishing the set of interlingual features but also the structure, and the relationship between features and structure (*e.g.* deciding whether semantic determination is represented as an interlingual feature or as a mother-daughter relation).

As to the second difficulty, such tree transformations involve complex processes such as restructuring, deletion and insertion, and the creation of abstract structures, *i.e.* structures without lexical content, in order to satisfy the purely structural requirements of the IS. The description of the input conditions and the output structures associated with each transformation become highly complex and redundant as the grammatical coverage of the system increases. This puts a heavy burden not only on the computing facilities but more importantly on the linguist/translator developing the MT system.

It seems, then, that tree structures are not optimal for the transfer of interlingual information, and yet we want to retain simple transfer rules of the type *maison* ⇒ *house*. Once semantic features have been merged with syntactic structure at a single level of representation, *i.e.* CS, we can entertain the possibility of transfer being a feature-to-feature operation between CSs.

Interlingual features of a semantic nature, *e.g.* determination, are then copied over unchanged. Synthesis will no longer be directed by complex language-specific translator rules, but rather will be a completely automatic process of generating the appropriate syntactic and/or morphological structure using the CS rules of the target language, instantiating — purely by unification — the semantic features delivered in transfer.

By combining linguistic information into a single level from which we perform transfer, we hope to achieve a more efficient formalism for sentence-based translation with minimal space and time requirements. Only then can we realistically look at extending the formalism to multi-sentence units, when we can then investigate intersentential anaphora resolution, temporal resolution, and context analysis.

4. CONCLUSION

We have described a simple formalism for describing complex linguistic and translational phenomena. By using a highly restricted set of notational devices, we can achieve a modest degree of linguistic coverage in a computationally efficient environment. The single basic operation is constraint satisfaction (*i.e.* unification augmented by negative and disjunctive resolution) of features within tree structures. No macros or arbitrary calls to some programming language are involved, and no lists or set values have been introduced, keeping the data representation and processing algorithms maximally simple.

The grammar implementations take advantage of shared resources, which is essential in any multilingual MT or NLP system. The grammars have been designed to maximize the role of the lexicon in determining well-formed linguistic structures, following the trend in most current theories of linguistics. This simplifies the description of any one language, and facilitates the inclusion of new languages.

The further developments planned for CAT2 involve merging morphological, syntactic and semantic data into a single level of representation, further simplifying the analysis and synthesis of linguistic structures, both for the system and for the linguist. This will have a beneficial impact on the overall space and time requirements, making CAT2 even more viable as a tool for conducting serious applied research into computational linguistics and translation.

BIBLIOGRAPHY

ALSHAWI, H., ARNOLD, D. J., BACKOFEN, R., CARTER, D. M., LINDOP, J., NETTER, K., PULMAN, S. G., TSUJII, J. and H. USZKOREIT (1991): *ET6/1 Formalism Study — Final Report*, CEC, DG XIII, Luxembourg.

ARNOLD, D., JASPAERT, L., JOHNSON, R., KRAUWER, S., ROSNER, M., DES TOMBE, L., VARILE, G. B. and S. WARWICK (1985): "A *MU1* View of the <C,A>,T Framework in EUROTRA", *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Hamilton, NY, pp. 1-14.

ARNOLD, D., KRAUWER, S., ROSNER, M, DES TOMBE, L. and G. B. VARILE (1986): "The <C,A>,T Framework in EUROTRA: A Theoretically Committed Notation for MT", *Proceedings of COLING'86*, Bonn, pp. 297-303.

ARNOLD, D. and L. SADLER (1990): "The Theoretical Basis of MiMo", *Machine Translation*, 5-3, pp. 195-222.

BALARI, S., DAMAS, L., MOREIRA, N. and G. B. VARILE (1990): "CLG(n): Constraint Logic Grammars", *Proceedings of COLING'90*, Helsinki, vol. 3, pp. 7-12.

BECH, A. and A. NYGAARD (1988): "The E-Framework: A Formalism for Natural Language Processing", *Proceedings of COLING'88*, Budapest, pp. 36-39.

GRIMSHAW, J. and A. MESTER (1988): "Light Verbs and θ-Marking", *Linguistic Inquiry*, 19-2, pp. 205-232.

MESLI, N. (1991): *Analyse et traduction automatique de constructions à verbe support dans le formalisme CAT2*, EUROTRA-D Working Paper 19b, IAI, Saarbrücken; also as *Funktionsverbgefüge in der maschinellen Analyse und Übersetzung: linguistische Beschreibung und Implementierung im CAT2-Formalismus*, EUROTRA-D Working Paper 19.

POLLARD, C. and I. SAG (1987): *Information-Based Syntax and Semantics*, CSLI Lecture Notes 13, Stanford, CSLI.

SHARP, R. (1988): "CAT2 — Implementing a Formalism for Multi-Lingual MT", *Proceedings of the 2nd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, Carnegie Mellon University, Pittsburgh, PA.

SHARP, R. (1991): "CAT2: An Experimental Eurotra Alternative", *Machine Translation*, 6-3, pp. 215-228.

SHIEBER, S. (1984): "The Design of a Computer Language for Linguistic Information", *Proceedings of COLING'84*, Stanford, pp. 362-366.

SHIEBER, S. (1986): *An Introduction to Unification-Based Approaches to Grammar*, CSLI Lecture Notes 4, Stanford, CSLI.