Linguistic Unmotivation in Eurotra

Abstract

Translator rules in the Eurotra systems 1 and 2 have an inherent tendency to combinatorial explosion. This creates very serious difficulties in grammar implementation, for example in re-ordering English adverbs in ERS. The Eurotra system 3 seems to have avoided this problem.

Note

This document was originally written in March 1987 and refers to the "<C,A>,T" formalism for linguistic Machine Translation, which was used in the first cycle of the second phase of Eurotra for development work. The problem of the <C,A>,T formalism described - namely, the combinatorial explosion of translator rules - has been carried over in the same basic shape into the current "Eurotra Framework" formalism. (This is in contrast to its disappearance in the "Version 3" or "Relaxed Compositionality" formalism, which is not being used for Eurotra development work). This document thus continues to be of interest; hence its late appearance in Internal Memorandum format.

Ian Crookston, University of Essex

November 1988

Linguistic Unmotivation in Eurotra

This paper attempts to draw out a problem that is latent in the Eurotra linguistic system: this latter term here refers to Reference Manuals 1.n and 2.n. (See most particularly Krauwer & King (1986), Arnold & des Tombe (1987)). The problem can be put thus: there is no linguistic motivation in a Eurotra translator-grammar. The problem can also be seen more broadly: linguistic motivation of Eurotra formalisms exists in either t-grammars or g-grammars, but cannot be spread across both. The difficulties this produces in actual grammar-writing seem likely to crucially affect Eurotra's feasibility as a working system. The linguistic system of Reference Manual 3.0 (Krauwer & King (1987)) may not suffer from this problem, a fact which arguably gives imperturbable motivation for proceeding to adopt this system.

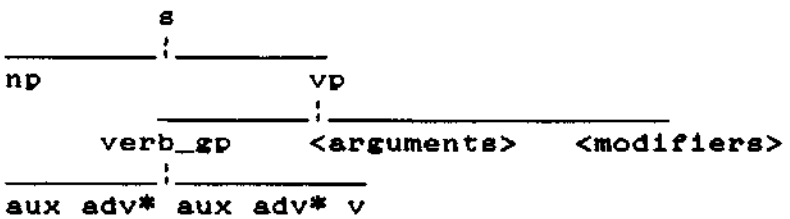
Does it matter? Eurotra ideology has constantly maintained that generators must be susceptible of, to quote one verbalisation, "a coherent linguistic description" (Krauwer & King (1986)). What we have found in the Essex group is that a linguistically coherent grammar is an inspectable, maintainable, and updatable grammar, relatively speaking. (Paul Bennet confirms from his experience of reviewing implementation reports that inspectability is very low in translators of the current type.) It is in all respects easier to work with a linguistically better organised grammar: and the advantages of this for a large-staff short-period project should be obvious.

Let us proceed to describe and exemplify the translator unmotivation problem. It is difficult to compress it into a few words, but it can be expressed thus: each single linguistic phenomenon has to be expressed n times in a translator grammar. I have no mathematical idea as to what determines the size of n, but there is certainly a lot of multiplication in the formula. If linguistic motivation ruled the translator grammars, there would only be addition and each linguistic phenomenon would need to be expressed once.

Exemplification can most readily come from the ECS and ERS generators and the ECS=>ERS translator submitted in the March 1987 British Group analysis module (Ananiadou et al (1987:17-38)).

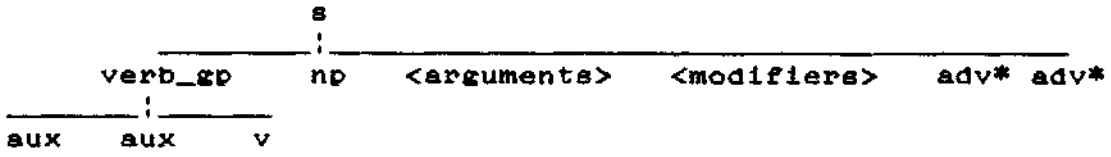
The simplest example is a process of English Eurotra-grammar which can be described as "adverb stripping". The adverbs in question are those that appear among the verbal elements in English, and the stripping is what happens to them between ECS and ERS. In brief, we need to get from an ECS of the form

[a]



(keeping to two auxiliaries for simplicity) to an ERS of the form

[b]



In other words, the adverbs need to be stripped out from among the verbal elements and placed among the modifiers under S. The choice of a "flat" verbal group structure to parse verbal elements at ECS does not affect the present issue: less flat structures would be no easier to handle. This adverb stripping is motivated in two ways. Firstly these adverbs are modifiers of the verb (or at least are mods in the Eurotra sense: it would be linguistically preferable to call some of them tranc's, but of course tranc is an unworkable idea in current prototypes) and must under the ERS code of conduct be represented as sisters of the other mods of the verb. Secondly, stripping is an essential step towards interface representations, where all the verbal elements must be congealed into a single node. There is thus ample linguistic and translational motivation for the relevant features of [a] and [b].

For completeness, below is a t-rule which will perform adverb-stripping:

```
[c]
t10         =       s.[ $ADV! ^ adv,
                        $PP! ^ pp,
                        $SUBJ!np,
                    vp.[vgrp.[ $ADV1! * (?,{cat~=v}),
                               $V1! ^ v,
                               $ADV2! * (?,{cat~=v}),
                               $V2! ^ v,
                               $ADV3! * (?,{cat~=v}),
                               $V3! ^ v,
                               $ADV4! * (?,{cat~=v}),
                               $V4! ^ v,
                               $ADV5! * (?,{cat~=v}),
                               $GOV! (v, {frame = v1}),
                               $ADV6! * (?,{cat~=v})
                              ],
                        $OBJ!(^ np),
                        $MOD!*]]

   =>       cs1(cvgrp($GOV, $V1, $V2, $V3, $V4),
                    $SUBJ, $OBJ, $PP, $MOD,
                    $ADV1, $ADV2, $ADV3, $ADV4, $ADV5, $ADV6,
                         $ADV),
```

(Ananiadou et al (1987:18))

What is important about this rule is the multiplicity of times this general form of rule appears in the translator. We have a single operation, one linguistic fact, which can be verbally described as "strip adverbs" and informally described as "turn [a] into [b]". It is realised in [c]: how it is realised is not easy to define, but it could be said that the location of the variables $ADV1 to $ADV6 within the two sides of the rule realises the operation "strip adverbs". In Ananiadou et al (1987) those variables have to be manipulated in an identical fashion to that in [c] in THIRTY-EIGHT rules. The single linguistic phenomenon of adverb-stripping is expressed n times, n=38.

Fundamentally, this is not a problem of bad organisation of the grammars concerned. It is true that certain choices were made in the ERS which led to n being 38 for adverb-stripping, where other choices would have led to a lower value of n. For example, the choice to have eight b-rules for eight sentence-frames multiplied n by eight. (In actual grammatical practice there are various reasons why the final figure is not divisible by eight, which need not concern us in the present connection.)

But firstly, those choices which were made were linguistically well-motivated. There is no criterion of ERS writing which could have enabled British Group workers to avoid n rising to a high figure like 38. That is, out of several linguistically coherent ERS', one entails this figure of 38 adverb strippings. Linguistic coherence in a generator CAN co-exist with chaos in the neighbouring translator. To put it differently, good grammar-writing does not guarantee easy translator writing.

Secondly, any prospect of reducing n to 1 is very remote. There will always be at the very least a handful of sentential-type t-b-rules, and thus the minimum value of n is extremely likely to be more than one in the case of adverb stripping. Linguistic coherence in a generator is EXTREMELY LIKELY to co-exist with chaos in the neighbouring translator. Adverb stripping is only an example: no-one

can say for certain that some clever way of doing it will not be found which will reduce n to one. What can be said is that translators are designed in such a way that single linguistic entities, such as for example adverb stripping, naturally tend to have to be represented n times. Easy translator writing can be engendered only by a combination of extreme luck and a kind of ingenuity which is not linguistic thinking

It is worth reinforcing here the overall point of linguistic motivation. What does it matter if we have three or thirty representations of a single process like adverb stripping? It matters for inspectability, maintainability and updatability. Suppose some new phenomenon were to be added to adverb stripping, such as floated quantifiers; that is, suppose the English module has to be expanded to cover such cases as

The committe have all been warned of this

The unit "all" now has to be stripped rather like an adverb, but put in a different place. It obviously matters now that there is more than one rule which strips adverbs, particularly if the person who wrote the rules has since found a permanent job. The thinking will be in terms that are linguistically motivated: "we now have to de-float quantifiers"; while the praxis will be a multiple operation which has to be tediously calculated from the organisation of the translator. The potential for error, compound error and simple running out of time is far higher.

The significance of this problem cannot be overemphasised: it seems to me that unless action is taken this will be the death of Eurotra. The grammars that have been written so far are from the point of view of practical MT toy grammars: they cover a neatly-defined core of well-understood phenomena. The second cycle implementation worker and the industrial implementor are going to have to perform hundreds of actions analogous to adding floated quantifiers to the English adverb-stripping rules, because all practical grammar is full of untidy peripheral phenomena. If these hundreds of actions are something like as difficult as this one presently is, that is, if we do not achieve a system where each single phenomenon is almost bound to be represented once in each grammar, the project will simply grind to a halt. An uninspectable research grammar is no use for anything but black humour.

Further examples of the multiple representation of a single linguistic concept in a translator are not hard to find in our chosen example field of the ECS=>ERS of Ananiadou et al (1987:17-38). A good example is the interaction between frame and construction type in sentential-type structures. There are eight verbal frames and four sentential-type structures are covered (the main clause, the relative with missing subject, the relative with missing object, and the reduced relative). (Again, the reasons why the final number of sentential-type t-rules is not 8 x 4 are not relevant here.) This entails two things: first that each translation of a frame is expressed in (roughly) four t-rules, and secondly that each sentential type is expressed in (roughly) eight t-rules. n=4 for each frame and 8 for each sentential type. For example, below are the four rules in which the simple monotransitive (SVO) verb frame is translated:

```
t10          =        s.[ $ADV! ^ adv,
                       $PP! ^ pp,
                       $SUBJ!np,
                  vp.[vgrp.[ $ADV1! * (?,{cat~=v}),
                             $V1! ^ v,
                             $ADV2! * (?,{cat~=v}),
                             $V2! ^ v,
                             $ADV3! * (?,{cat~=v}),
                             $V3! ^ v,
                             $ADV4! * (?,{cat~=v}),
                             $V4! ^ v,
                             $ADV5! * (?,{cat~=v}),
                             $GOV! (v, {frame = v1}),
                             $ADV6! * (?,{cat~=v})
                            ],
                      $OBJ!(^ np),
                      $MOD!*]]

    =>       cs1(cvgrp($GOV, $V1, $V2, $V3, $V4),
                      $SUBJ, $OBJ, $PP, $MOD,
                      $ADV1, $ADV2, $ADV3, $ADV4, $ADV5, $ADV6,
                           $ADV),

t10_srel_s =        s.[ %  no subject %
                  vp.[vgrp.[ $ADV1! * (?,{cat~=v}),
                             $V1! ^ v,
                             $ADV2! * (?,{cat~=v}),
                             $V2! ^ v,
                             $ADV3! * (?,{cat~=v}),
                             $V3! ^ v,
                             $ADV4! * (?,{cat~=v}),
                             $V4! ^ v,
                             $ADV5! * (?,{cat~=v}),
                             $GOV! (v, {frame = v1}),
                             $ADV6! * (?,{cat~=v})
                            ],
                      $OBJ! ^ np.       % optionality removed for test
                      $MOD! *]]

    =>       cs1(cvgrp($GOV, $V1, $V2, $V3, $V4),
                      crel_trace( cempty ), $OBJ, $MOD,
                      $ADV1, $ADV2, $ADV3, $ADV4, $ADV5, $ADV6),
```

```
t10_srel_o =        s.[ $SUBJ! np,
                vp.[vgrp.[ $ADV1! * (?,{cat~=v}),
                           $V1! ^ v,
                           $ADV2! * (?,{cat~=v}),
                           $V2! ^ v,
                           $ADV3! * (?,{cat~=v}),
                           $V3! ^ v,
                           $ADV4! * (?,{cat~=v}),
                           $V4! ^ v,
                           $ADV5! * (?,{cat~=v}),
                           $GOV! (v, {frame = v1}),
                           $ADV6! * (?,{cat~=v})
                          ],
                            $MOD! *]]

   =>       cs1(cvgrp($GOV, $V1, $V2, $V3, $V4),
                    $SUBJ, crel_trace( cempty ), $MOD,
                    $ADV1, $ADV2, $ADV3, $ADV4, $ADV5, $ADV6),

tredrel1 = ap.[vp.[vgrp.[ $ADV1! * (?,{cat~=v}),
                           $V1! ^ v,
                           $ADV2! * (?,{cat~=v}),
                           $V2! ^ v,
                           $ADV3! * (?,{cat~=v}),
                           $V3! ^ v,
                           $ADV4! * (?,{cat~=v}),
                           $V4! ^ v,
                           $ADV5! * (?,{cat~=v}),
                           $GOV! (v, {frame = v1}),
                           $ADV6! * (?,{cat~=v})
                          ],
                    $OBJ!(^ np),
                    $MOD!*]]

   =>       credrel1(cvgrp($GOV, $V1, $V2, $V3, $V4),
                    $OBJ, $MOD,
                    $ADV1, $ADV2, $ADV3, $ADV4, $ADV5, $ADV6),
```

(Ananiadou et al (1987:18, 23-24, 30))

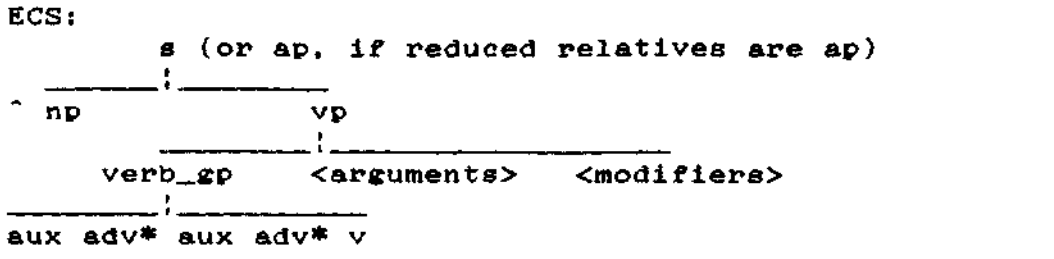
Again, it is not hard to see ways in which the number of sentential-type t-b-rules in this particular translator could be reduced. But the points made above in connection with adverb stripping apply with equal force. The decisions which led to this lapine breeding were perfectly sensible linguistic decisions about the shape of ERS, so that good grammar-writing does not guarantee easy translator writing. And there is an inherent tendency for combinatorial breeding of t-rules, so that luck and non-linguistic ingenuity are required to keep the value of n low for the single phenomenon of a frame or a sentence type.

It is also worth pointing out that since number of rules has an effect on speed, and since the current prototypes are designed so that t-rules multiply combinatorially, translators in the current prototypes are doomed to be slow-running inherently. Slow speed, as planning committee has noted, has already taken its toll in preventing thorough testing of the first cycle implementations: in current prototypes it will continue to impede research. Moreover, it will do so to an exponentially increasing degree as coverage expands, because expanding coverage means expanding translators.

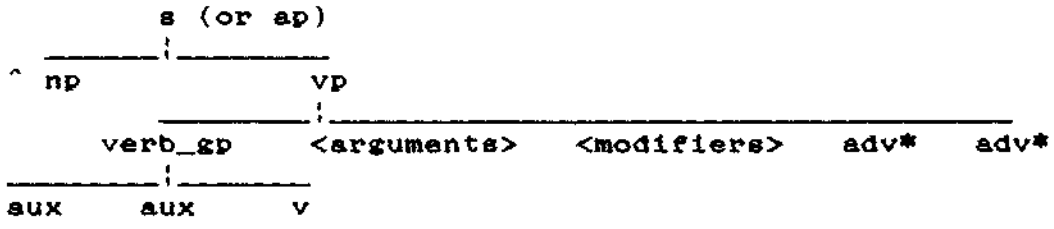
There is, however, one possibility in language-module design which would cleanly and neatly solve the problem under discussion using current software. If this possibility were followed, it would become natural and almost automatic to express each linguistic phenomenon once in translators. This possibility is that of increasing the number of generators in the module.

Take the examples of adverb stripping, sentential construction types, and frames. Eurotra prototypes allow these translation cases to be expressed in sequence, in, say, the following way:

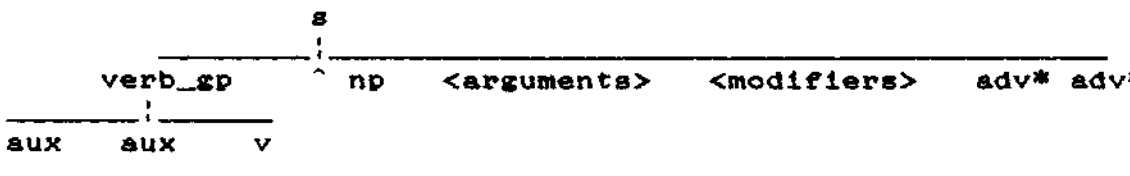
[d]
ECS:



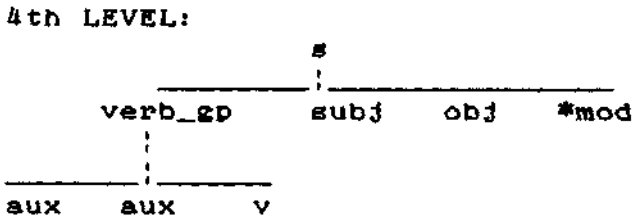
[e]
2nd LEVEL:



[f]
3rd LEVEL:



[g]
4th LEVEL:



4th LEVEL could just as well be called ERS: it would correspond very closely to what we presently know as ERS and would be able to do so exactly. In the scheme of things represented in [d]-[g] each linguistic phenomenon could be represented once in some translator. At ECS=>2nd LEVEL, there would be a single sentence t-rule which would perform adverb stripping: n=1 for adverb stripping. At 2nd LEVEL=>3rd LEVEL, there would be rules that turned each sentential-type node into something which would have a verb-group as first daughter: essentially a VP-flattening rule. n=1 for sentential types, possibly. At 3rd LEVEL=>ERS, there might be one sentence t-rule for each verb frame: n=1 for each frame.

The drawbacks of such an approach seem to me to be so fundamental that it should not be seriously considered for a moment. The inspectability problem would shift from individual translators to the module as a whole. There would be an absence of linguistic reasoning within each generator and a related lack of it behind the decision of which translator to do a particular job in. In [d]-[h], there is no very solid linguistic characterisation of 2nd LEVEL and 3rd LEVEL: in a full module written with this methodology there would be many such levels. In that sense, [d]-[h] is a very straightforward violation of mu-2 theory. More broadly, to add a new phenomenon in the real three-level Eurotra framework is a problematic task, for the reasons we have been arguing all along; while to add a new phenomenon is

multi-level framework in which the levels of [d]-[h] might play a part would demand complex and linguistically unmotivated decisions as to, essentially, how to order the transformations. The effects of some decisions would be ramified beyond all inspectability. Sharing of grammar-writing between many people would touch impossibility.

The reason why a multi-level system is expounded here is that it illustrates a rather fundamental fact about the existing Eurotra framework. To avoid expressing adverb-stripping 38 times (or, at best, several times) in translators, increase the number of generators. Linguistic motivation can be achieved in the translators, at the expense of losing a linguistic characterisation of each level. The problems of poor linguistic organisation in translators which were the subject of the earlier discussion are now seen to be a reflex of the more fundamental problem that one cannot have linguistic motivation throughout a Eurotra framework: the design of the system limits it to bounded areas, the location of which may be chosen by the linguists in the project. There is a lump of linguistic unmotivation in the Eurotra system, which can be squeezed out of the translators, but only into the generators. We have all agreed to locate linguistic motivation in a set of three levels of representation and thus in three generators, but that does not mean that we have a linguistically well-organised framework as a whole.

The question must finally be asked of where version 3.0 of the system stands in all this. Does it suffer from the same inefficiency of translator design? My own understanding of this system is limited by its newness and by the overconciseness of the existing documentation (Krauwer & King (1987), Arnold et al (1986), Arnold (1987)), but it seems fairly clear that the problem does not recur.

There is the clear case of one of the phenomena discussed above, the verb frames, which receive multiple representation in the ECS=>ERS of Ananiadou et al (1987:17-38). The whole frame problem is radically better treated in version 3.0: that is one of the central pillars around which 3.0 is built. Multiple representation is certainly refined out here, because frames are translated totally independently of other factors.

More generally, it seems to be true that the notion of extraction in this system exists precisely to avoid the multiple representation of a single phenomenon in translators. An extraction is a process which isolates an arbitrary subset of nodes on the lhs of a translation, and specifies what their translation is, independently of the translation of the rest of the lhs. This seems to give us exactly the required ability to isolate single phenomena and translate them once and once only.

For example, a single t-rule using extraction could perform adverb-stripping, no matter how many t-rules treated the sentential nodes as a whole. The syntax of such a rule would be very similar to [c], but the translation of the subject NP and other arguments could be totally independent of the adverb-stripping rule. One rule similar to [c] would perform stripping and other, much simpler, rules would flatten VP's and create subjects, objects, etc.

Bearing in mind the difficulty of writing translator grammars in the present system, a difficulty which I have argued above is crucial, it seems to follow that 3.0 is the Eurotra framework of the future. If, that is, Eurotra is to have a future.

References

Ananiadou, E, B Ashman, A Betts, I Crookston, L Hammond, L Humphreys, L Juarez, E Rigler, J Shelton & N Underwood (1987) "Appendix to the British Group Final Implementation Report on the First Cycle", DGXIII, CEC, Luxembourg

Arnold, D J (1987) "Difficult T-cases in Version 3.0", ms, University of Essex

Arnold, D J, S Krauwer, D Petitpierre, L des Tombe & N Varile (1986) "Proposal for a New Core Framework for Eurotra: November 1986", ms, University of Essex

Arnold, D J & L des Tombe (1987) "Basic Theory and Methodology in Eurotra", in S Nirenburg (ed) Machine Translation: Theoretical and Methodological Issues, CUP, Cambridge, 114-135

Krauwer S & M King (1986 eds) The Eurotra Reference Manual Version 1.2 DGXIII, CEC, Luxembourg

Krauwer S & M King (1987 eds) The Eurotra Reference Manual Version 3.0 DGXIII, CEC, Luxembourg