Dutch and German verb clusters in Performance Grammar

Gerard Kempen
Department of Psychology, Leiden University
Max Planck Institute for Psycholinguistics, Nijmegen
The Netherlands
kempen@fsw.leidenuniv.nl

and

Karin Harbusch
Computer Science Department
University of Koblenz-Landau
Germany
harbusch@informatik.uni-koblenz.de

Prefinal version, October 2002

The final version of this manuscript has been published in: Seuren, P. & Kempen, G. (Eds.) (2003). *Verb constructions in German and Dutch*. Amsterdam: Benjamins.
Within the psycholinguistically motivated syntactic framework of *Performance Grammar*, we develop a linearization model that we claim captures a broad range of linear order phenomena in Dutch and German clauses, including the verb clustering phenomena focused in this volume.

In Section 1, we lay out the essentials of Performance Grammar (PG hereafter). Sections 2 and 3 are devoted to the PG treatment of verb clusters in Dutch and German respectively. In Section 4, we draw some conclusions.

1. Essentials of Performance Grammar

Performance Grammar (Kempen & Harbusch, 2002; Harbusch & Kempen 2002)\(^1\) consists of separate components generating, respectively, the hierarchical and the linear structure of sentences\(^2\). Hierarchical structures in PG are unordered trees (‘mobiles’) composed out of elementary building blocks called *lexical frames*. Every lexical entry (*lemma*) in the *Mental Lexicon* has associated with it a lexical frame encoding its information concerning word class (part of speech), subcategorization features, and morphological diacritics (person, gender, case, etc.). Lexical frames are retrieved from the Lexicon by a *lexicalizer* in response to a conceptual ‘message’ (during language production) or to a word string (during language perception). Associated with every lexical frame is a one-dimensional array called *topology*, which specifies a fixed number of positions (or *slots, landing sites*) where segments (branches, constituents) of lexical frames can be stored in left-to-right order. The segments stored in a topology may include constituents originating from a lexical frame lower in the hierarchy. As we will see below, this happens without affecting the frame hierarchy.

In this Section, we discuss the hierarchical and linear grammar components in turn. For the empirical psycholinguistic arguments in support of the separation between hierarchical and linear structure we refer to Kempen & Harbusch (2002).

1.1 Hierarchical structures in Performance Grammar

PG’s hierarchical component generates unordered trees by combining 3-tiered ‘mobiles’ called lexical frames. Figure 1 shows the eight lexical frames expressing how each of the words in Dutch example (1) is used\(^3\).

---

1 For predecessors of PG, see Kempen & Hoenkamp (1987) and De Smedt & Kempen (1990a, b).
2 Other grammar formalisms that assign hierarchical and linear computations to separate rule systems, are GPSG (Gazdar, Klein, Pullum & Sag, 1985) and HPSG (Pollard & Sag, 1994).
3 For readers who are more familiar with German than with Dutch: the syntactic structure of the German translation of this example (*Denkst du, dass er das Auto repariert hat*) is identical in all currently relevant aspects.
(1) Denk je, dat hij de auto gerepareerd heeft?

Think you that he the car repaired has

‘Do you think that he has repaired the car?’

The top layer of a frame consists of a single phrasal node (the ‘root’; e.g. S, NP, DP, CP), which is connected to one or more functional nodes in the second layer (e.g., SUBject, HeaD, Direct OBJect, CoMPlement, PREDicate, MODifier). At most one exemplar of a functional node is allowed in the same frame, except for MOD nodes, which may occur zero or more times. Every functional node dominates exactly one phrasal node (‘foot’) in the third layer, except for H(ea)D which immediately dominates a lexical (part of speech) node. Each lexical frame is ‘anchored’ to exactly one lexical item, which is printed below the categorial node serving as the frame’s HeaD.

![Simplified lexical frames underlying the eight words of sentence (1).](image)

Figure 1. Simplified lexical frames underlying the eight words of sentence (1). Left-to-right order of branches is arbitrary. The basic shape of lexical frames is retrievable from the Mental Lexicon in response to contents of the to-be-expressed conceptual message; however, certain branches (e.g. CMPR) are added as a consequence of local syntactic constraints (CMPR = CoMplementizeR; CP = Complementizer Phrase).

Lexical frames are combined to form larger mobiles by a substitution operation which replaces phrasal foot nodes by lexical frames. More precisely, substitution involves a simple, non-recursive form of unification called feature unification, and merges the root of one frame with one phrasal foot of another frame (Figure 2). Unification of two feature matrices proceeds as follows:

(A) For each attribute shared by the two matrices, take the value sets, compute their intersection, and delete the members that do not belong to the intersection. (Attributes occurring in one matrix only will not be affected.)

(B) If at least one of the intersections is the empty set, unification fails.

(C) Otherwise, unification succeeds, and a feature matrix containing one token of each of the shared or non-shared attribute-value pairs is returned.

For instance, the following matrices have two attributes in common:
The intersections of these disjunctive value sets are \{1st, 3rd\} and \{nominative\} respectively. Since none of these sets is empty, unification succeeds and the following matrix is returned:

\[
\begin{align*}
\text{case} &= \{\text{nominative}\} \\
\text{number} &= \{\text{singular}\} \\
\text{person} &= \{3\text{rd}, 1\text{st}\}
\end{align*}
\]

It follows that if a feature matrix contains at least one feature whose value set is empty, it cannot successfully unify with any other feature matrix. In Section 2 and 3 below we introduce order-related features and apply unification as a mechanism for assigning constituents to positions in a linear structure.

Unification uses the lexical frame as its domain of locality and serves, among other things, to select the value of agreement features (e.g. entailing second person singular \textit{denk} of the verb \textit{denken}; agreement features are not shown in the figure). Unification operates on features stored in the feature matrix that is associated with every categorial node (i.e., lexical or phrasal node). Such a matrix is a set of pairs, each consisting of an attribute and a finite set of values. Features are instantiated with a non-empty value set. An attribute is a character string (e.g., “gender”, “person”, “number”). A value set contains a non-zero finite number of character strings (e.g., \{\text{sing}\}, \{1\text{st}, 2\text{nd}, 3\text{rd}\}), each representing a possible value of the attribute (disjunctive value sets).

Figure 2. Verb frame hierarchy underlying example (1). The root S-node of the verb frame associated with \textit{heeft} and the CoMPlement S-node of \textit{denk} have merged as a result of substitution, and so have the S-CMP of \textit{heeft} and the root S-node dominating \textit{gerepareerd}. Left-to-right order of branches is arbitrary.

1.2 Linear structure in PG

In order to assign a left-to-right position to the branches of lexical frames, we introduce an additional type of data structure. Associated with every lexical frame is a \textit{topology}, a
one-dimensional array specifying a fixed number of positions (or slots, landing sites) for its constituents. The topology of a verb frame (i.e., of a finite or non-finite clause) allocates storage space for each of various grammatical functions that can be fulfilled by its constituents, e.g., to the HeaD verb, to the SUBJect NP, the Direct OBJect NP, etc. The topology that we use for Dutch and German clauses specifies nine different slots, labeled as indicated in Figure 3.

![Figure 3. Slot labels used in topologies for Dutch and German clauses.](image)

Table 1 illustrates which clause constituents select which slots as their landing sites. Constituents may select different positions depending on their shape. For instance, if the Direct OBJect role is fulfilled by a Wh-phrase, it will end up in the Forefield of the clause rather than in the Midfield. A CoMPlement clause will land in slot E2 if it is finite; non-finite CoMPlements have several additional placement options.

We show linearization at work on sentence (2), an abbreviated version of (1). As we assume that every lexical item launches its own lexical frame and topology, a sentence containing more than one verb instantiates several clausal topologies. This applies to verbs of any type, whether main, auxiliary or copula. It follows that sentence (2) needs two topologies (Figure 4)

(2) Denk je dat hij de auto repareert?
    ‘Do you think that he repairs the car?’

![Figure 4. Linearization of example (2). The dashed slots of the embedded topology denotes ‘topology sharing’, to be discussed below.](image)

Slot F1 of the interrogative main clause remains empty in the absence of a Wh-constituent. The pronominal Subject NPs of the main and the subordinate clause go to slot M2 of their respective topologies. The HeaD verb selects slot M1 in the main clause, and M6 in the CoMPlement. The CoMPlementizeR dat goes to M1 of the CoMPlement’s topology. Finally, the root S-node dominating the finite CoMPlement clause lands in E2, as indicated by the bullet and upward double arrow.
Table 1. Examples of topology slot selection rules (Dutch and German clauses). Precedence between constituents landing in the same slot is marked by "<<". The details of the placement of HeaD verbs, CoMPlenment clauses, and PaRTicles are explained in Sections 2 and 3.

<table>
<thead>
<tr>
<th>Slot</th>
<th>Filler</th>
</tr>
</thead>
</table>
| F1   | Declarative main clause: SUBJect, Topic or Focus (one constituent only)  
Interrogative clause: Wh-constituent, including Du. *of* and Ger. *ob* 'whether'  
Complement clause: Wh-constituent |
| M1   | Main clause: HeaD verb  
Complement clause: CoMPlenmentizeR *dat/om* (Du.) and *dass* (Ger.) 'that' |
| M2   | Subject NP (iff non-Wh), Direct Object (iff personal pronoun) |
| M3   | Indirect OBJect (iff non-Wh) << Direct OBJect (iff non-Wh and non-pers.pro.) |
| M4   | PaRTicle (Du. only) |
| M5   | Non-finite CoMPlenment of Verb Raiser |
| M6   | Subordinate clause;  
Du.: Pre-INFinitive *te* 'to' << HeaD verb  
Ger.: PaRTicle << Pre-INFinitive *zu* 'to' << HeaD |
| E1   | Non-finite Complement of Verb Raiser (Du. only) |
| E2   | Non-finite CoMP of VP Extraposition verb  
Finite Complement |

In case of more than one verb, the topologies associated with them are allowed to share certain identically labeled slots, conditionally upon several restrictions. After two slots have been shared, they are no longer distinguishable; in fact, they are the same object. In example (2), the embedded topology shares its F1 and E2 slots with their namesakes in the matrix topology. This is indicated by the dashed border of the two bottom slots.

Now consider example (3), where the Direct OBJect of *repareert* is a Wh-phrase (*welke auto* 'which car'). This constituent lands in the shared F1 slot: see Figure 5. As a consequence, the Wh-phrase gets ‘fronted’ and seems to have been ‘extracted’ from the complement. This behavior, caused by lateral topology sharing manifests itself as upward movement of constituents in shared slots. We will call this effect promotion. In the topology diagrams, promotion is indicated by an upward single arrow, and a bullet located at the landing site of the promoted constituent.

(3) *Welke auto denk je dat hij reparert?*  
‘Which car do you think that he repairs?’
Example (1) embodies a more radical case of cross-clause topology sharing. The topologies associated with *heeft* and *gerepareerd* share the entire region extending from F1 through M4. The result is promotion of Direct OBJect *de auto*, as shown in Figure 6.

The overt constituent order of a sentence is determined by a *Read-out module* that traverses the hierarchy of topologies in depth-first, left-to-right manner, scanning shared slots only once. Any lexical item it 'sees' in a slot, is appended to the output string. E.g., *welke auto* in example (2) is seen while the Reader scans the matrix topology rather than during its traversal of the embedded topology. Consider the artificial example of lateral topology sharing in Figure 7. The uppermost clause shares lateral (peripheral) slots at, respectively, positions P1 and P5 of its topology with its downstairs neighbor. The lower topology has landed in slot P2 to the right of item *A*. The Read-out module first processes item *a* at P1 and *A* at P2, then proceeds to the non-shared region P2 through P4 of the lower topology, and outputs the sequence *bcd*. Having reached shared slot P5 it returns to slot P2 of the upper topology and processes there the remaining slots P3 through P5, which yields the sequence *Be*. The resulting string is *aAb-cdBe* (cf. lower panel of Figure 7.)
The two shared peripheral regions will be called "lateral", the non-shared region "central". (Notice that the central region need not coincide with the Midfield; actually, in most cases they are different.)

\[ \begin{array}{ccccc} 
P1 & P2 & P3 & P4 & P5 \\
\bullet & A & \bullet & B & \bullet \\
\hline \\
& a & b & c & d & e \\
\hline \\
& a & A & \bullet & B & e \\
\end{array} \]

Figure 7. Read-out of topologies in case of lateral topology sharing. Upper panel: P1 through P5 are labels of the slots in both the upper and lower topology. Shared slots are indicated by dashed rectangles. Bullets above single arrows represent the position of items that have been ‘promoted’ into the upper topology. The double arrow points to the landing site of the non-shared region of the topology at its tail. Lower panel: The shared slots P1 and P5 of the embedded topology has not been drawn because they are the same objects as slots P1 and P5, respectively, of the governor’s topology. Therefore, items \( a \) and \( e \) are ‘seen’ during read-out of the matrix topology. The output sequence reads \( aAbcdBe \).

(4) General Constraints on cross-clause lateral topology sharing in Dutch and German

a. A verb frame can left-peripherally share part of its clausal topology with that of another verb frame only if they are located at adjacent levels of the verb frame hierarchy.

b. Only lateral (i.e., left- and right-peripheral) regions of a topology are sharable.

c. The HeaD of a lexical frame, i.e. the verb, does not participate in topology sharing: it cannot be promoted outside of its own topology, nor can the slot serving as its landing site be the promotion target of any constituent from a lower topology.

Whether or not two clausal topologies share some lateral region(s), and which slots will be affected, depends on a language-specific cross-clause lateral topology sharing rule. The topology sharing rules for Dutch and German are shown in Table 2. The variables \( LS \) and \( RS \) stand for, respectively, the number of left- and right-peripheral slots that the complement clause shares with the topology of its governor. For instance, the first row of the Table indicates that if a Dutch or German complement clause is interrogative, it shares no slots with its upstairs neighbor. The last row deals with declarative non-finite complements of Verb Raisers (auxiliaries in particular; see beginning of Section 2): In both Dutch and German, the right-peripherally shared area includes one slot only, i.e.
The left-peripherally shared areas in the two languages differ: In German it covers five slots (F1 through M4); in Dutch it varies optionally between four (F1 through M3) and six (F1 through M5). Figure 4 and Figure 5 above illustrate sharing of the F1 slot of a finite declarative complement clause⁴. Figure 6 shows sharing of the entire left-peripheral region F1 through M4, due to the fact that *de auto gerepareerd* is the declarative non-finite complement of the auxiliary *heeft* ‘has’.

Table 2. Size of the left- and right-peripheral shared topology areas (LS and RS) in diverse complement constructions. Numbers separated by a colon denote the minimum and the maximum size of a shared region.

<table>
<thead>
<tr>
<th>Type of Complement clause</th>
<th>Dutch</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrogative</td>
<td>LS=0</td>
<td>LS=0</td>
</tr>
<tr>
<td></td>
<td>RS=0</td>
<td>RS=0</td>
</tr>
<tr>
<td>Declarative &amp; Finite</td>
<td>LS=1</td>
<td>LS=1</td>
</tr>
<tr>
<td></td>
<td>RS=0</td>
<td>RS=0</td>
</tr>
<tr>
<td>Declarative &amp; Non-Finite, VP Extraposition</td>
<td>LS=1</td>
<td>LS=1</td>
</tr>
<tr>
<td></td>
<td>RS=0:1</td>
<td>RS=0:1</td>
</tr>
<tr>
<td>Declarative &amp; Non-Finite, Third Construction</td>
<td>LS=1:6</td>
<td>LS=1:6</td>
</tr>
<tr>
<td></td>
<td>RS=0:1</td>
<td>RS=0:1</td>
</tr>
<tr>
<td>Declarative &amp; Non-Finite, Verb Raising</td>
<td>LS=4:6</td>
<td>LS=5</td>
</tr>
<tr>
<td></td>
<td>RS=1</td>
<td>RS=1</td>
</tr>
</tbody>
</table>

The final ingredient of PG’s linearization system utilizes the non-recursive feature unification mechanism introduced in Section 1.2. We allow lexical frames to carry not only agreement features but also positional features. While their values are determined as part of the unification process. A simplified example is provided in Figure 8 by the placement of the past participle *gerepareerd* of example (1). The “CMPtype” feature on the Complement S-node in the lexical frame of the Dutch auxiliary *heeft* ‘has’ specifies the disjunctive value “{M5, E1}”. This means that the complement of *heeft* is free to land either in slot M5 or in slot E1. Because the root S-node of *gerepareerd* happens to have a CMPtype feature with one of these values, its unification with *heeft*’s complement S-node succeeds. This licenses the Complement clause headed by *gerepareerd* to land in slot E1 of *heeft*’s topology (see Figure 6). As will become clear shortly, unification of positional features offers a simple but effective method to control verb order in Dutch and German clause-final verb clusters.

⁴ The complement clause in Figure 5 is declarative notwithstanding the fact that it includes a Wh-phrase. Compare sentence (3) with the following English paraphrase, which brings out the scope of the Wh-phrase: ‘For which car is it the case that he repairs x?’.
Now that the essentials of PG’s linear ordering component are in place, we can proceed to the verb clustering phenomena in the target languages.

2. Verb clustering in Dutch

The rule for cross-clause topology sharing in Dutch is shown in Table 2. It contains provisions for three different types of non-finite CoMPliment clause constructions traditionally called ‘Verb Raising’, ‘VP Extraposition’, and ‘Third Construction’ (Den Besten & Rutten, 1989). We now discuss some examples that illustrate the joint effect of slot selection (Table 1) and topology sharing.

In sentence (5), the finite middle clause lands in slot E2 of the matrix. The lower clause is governed by the verb moeten ‘must, have to’, a Verb Raiser. Therefore, it shares the entire left-peripheral region from F1 through M3 \((LS=4; \text{cf. Table 2})\), and possibly even M4+M5 \((LS=4:6)\), with its upstairs neighbor. This causes promotion of the Direct OBJect and the PaRTicle of the lower clause. If slot M4 of the lower topology were not shared, PaRTicle op would end up after moeten.

---

5 Verbs that allow their non-finite complements to be introduced by CoMPlimentizeR om will be treated as Extraposition Verbs.
(5) a. Zei je dat ik haar op moet bellen?
    said you that I her up have-to call
    ‘Did you say that I have to call her up?’

The examples in (6) exemplify the Third Construction, with vergeten ‘forget’ as governing verb. The topology-sharing rule now licenses varying amounts of topology sharing in the left periphery, and thereby several alternative output strings. Sentences (6a) and (6b) represent the extremes: no promotion of any constituent in (6a) vs. all constituents dependent on the infinitival verb promoted in (6b). Intermediate amounts of topology sharing yield acceptable sentences provided that sharing is indeed left-peripheral: see the grammaticality contrast between items (6b/c) on the one hand and (6d/e) on the other. The grammaticality contrast between examples (6f/g) illustrates VP Extraposition: vragen ‘ask’ governs a VP Extraposition construction, like dwingen ‘force’. This rules out promotion of Direct OBJect de fiets into the same slot as vragen’s Indirect OBJect Jan.

(6) a. ... dat Jan vergat Marie het boek terug te geven
    that John forgot Mary the book back to give
    ‘... that John forgot to give Mary the book back’

b. ... dat Jan Marie het boek terug vergat te geven

b. ... dat Jan Marie het boek terug vergat te geven

c. ... dat Jan Marie vergat het boek terug te geven
d. *... dat Jan het boek vergat Marie terug te geven

```
<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M6</th>
<th>E2</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>Jan</td>
<td></td>
<td></td>
<td>vergat</td>
<td></td>
</tr>
</tbody>
</table>
```

```
Marie het boek terug te geven
```

e. *... dat Jan het boek terug vergat Marie te geven

```
<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M6</th>
<th>E2</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>Jan</td>
<td></td>
<td></td>
<td>vergat</td>
<td></td>
</tr>
</tbody>
</table>
```

```
Marie het boek terug te geven
```

f. ... dat Marie Jan vraagt (om) de fiets te repareren

that Marie Jan asks (CMPR) the bike to repair

```
<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M6</th>
<th>E2</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>Marie</td>
<td>Jan</td>
<td></td>
<td>vraagt</td>
<td></td>
</tr>
</tbody>
</table>
```

```
(om) de fiets te repareren
```

That Marie asks Jan to repair the bike

g. *... dat Marie Jan de fiets vraagt te repareren

```
<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M6</th>
<th>E2</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>Marie</td>
<td>Jan</td>
<td></td>
<td>vraagt</td>
<td></td>
</tr>
</tbody>
</table>
```

```
(om) de fiets te repareren
```

That Marie asks Jan to repair the bike

The rules discussed so far may cause a congestion of verbs at the end of a finite clause. Such a verb cluster contains one finite verb, one or more non-finite dependent verbs (infinitive or past participle), possibly with one PaRTicle and/or one or more Pre-INFinitives te ‘to’. Some representative examples are given in (7). They include three Verb Raisers (the auxiliaries hebben ‘have’, zullen ‘will/shall’ and kunnen ‘can’, to be able to’), one main verb that cannot take a non-finite complement clause (zingen ‘to sing’), and a Third Construction verb (hopen ‘to hope’). The sentences also exemplify the ‘infinivitivus pro participio’ (IPP) phenomenon. If the perfective auxiliary hebben takes another Verb Raiser (VR) as its complement, this verb surfaces as an infinitive rather than as a past participle. In (7m), for instance, the infinitival form kunnen substitutes for the past participle gekund as complement of hebben. The subscripts in the examples denote levels in the verb frame hierarchy.
... dat ze deze liedjes zullen zingen
that they these songs will sing
‘... that they will sing these songs’

b. ... dat ze deze liedjes zingen, zullen

h. *... hebben, zullen, gezongen

... hebben, zullen, gezongen
‘... will have sung’

i. *... hebben, gezongen, zullen

j. *... gezongen, hebben, zullen

k. ... zullen, kunnen, zingen
‘... will be able to sing’

l. *... zullen, zingen, kunnen (all other permutations unacceptable)

m. ... hebben, kunnen, zingen
‘have been able to sing’

n. *... hebben, zingen, kunnen (all other permutations unacceptable)

o. ... hebben, kunnen, zingen (all other permutations unacceptable)

... would-have will sing

‘... would have intended/promised/planned to sing’

p. ... zullen, hebben, kunnen, zingen
‘will have been able to sing’

q. *... zingen, zullen, hebben, kunnen (all other permutations bad)

r. ... dat ze deze liedjes hopen, te zingen

that they these songs hope to sing

... ‘that they hope to sing these songs’

s. *... dat ze deze liedjes te zingen, hopen

... dat ze deze liedjes te kunnen, zingen
(all other permutations unacceptable)

... that they hope to be able to sing these songs’

u. ... zouden, kunnen, hebben, gezongen

‘it might be possible that they have sung (these songs)’

v. ... zouden, kunnen, gezongen, hebben

w. ... zouden, gezongen, kunnen, hebben

x. ... gezongen, zouden, kunnen, hebben

y. *... hebben, gezongen, zouden, kunnen (all other permutations unacceptable)

z. *... hebben, kunnen, gezongen, zouden (all other permutations unacceptable)
Our approach to controlling word order in verb clusters deploys the unification mechanism for positional features introduced at the end of Section 1.1 (cf. Figure 8). We postulate that whenever a verb frame is instantiated as a finite or non-finite complement, its root S-node contains the CoMponent type feature. The values of the “CMPtype” feature simultaneously represents three aspects of the complement:

(A) verb status: the HeaD verb is a bare infinitive (coded as “Infin”), an infinitive preceded by a Pre-INFinitive (“PInf”) or a past participle (“PastP”) or a finite verb (“Fin”)

(B) landing site (slots M5, E1 or E2), and

(C) pattern of left-peripheral topology sharing allowed by the HeaD verb (cf. Table 2): VP Extraposition (VE), Verb Raising (VR) Third Construction (VT), or finite complementation (VF). Verbs that take no complement at all will be designated as 'simple verbs' (VS). In order to abbreviate notation we propose to let the symbol "V" in a CMPtype feature cover any of the verb types distinguished here: V={VE, VF, VR, VS, VT}.

For instance, the feature “CMPtype = {M5-VR-Infin}” represents the option “non-finite complement clause that is headed by an infinitival Verb Raiser without Pre-INFinitive and lands in slot M5”. The feature “CMPtype = E2-V(E,T)-PInf” indicates that the clause carrying it will be realized as a non-finite clause in slot E2 of its governer, that the HeaD verb of this clause takes a VP Extraposition or a Third Construction complement, and that the HeaD verb will be preceded by Pre-INFinitive te. Notice that characters between parentheses denote optional elements. E.g. “V(E, S)” is matched by an Extraposition verb or by a simple verb without a complement; the code “(P)Infin” stands for an infinitive with or without a PINF. “(M5, E1)” indicates that M5 or E1 can be chosen as landing sites. Importantly, the symbol "V" is an abbreviation of "V(E,F,R,S,T)" and matches any verb type from the set {VE, VF, VR, VS, VT}.

We assume that every root S-node of a complement clause is instantiated with a CMPtype feature whose value set includes one or more such tripartite options. Furthermore, the CMP-S foot node in the lexical frame of verbs that take finite or non-finite complements, also includes a CMPtype feature. Unification of such a foot S-node with the root S-node of another verb frame succeeds only if this operation yields a non-empty CMPtype value set (cf. the definition of feature unification in Section 1.1).

Figure 9 shows the CMPtype features for the verbs in (7). The modal auxiliaries have two entries each. Sentence (7a) involves unification of the CMP-S node in the lexical

---

6We borrow this term from Bech (1955).
7The PINF is realized as to in English, as te in Dutch, and as zu in German.
frame for zullen2 with the root node of the zingen frame. The one remaining option
“E1-V-Infin” directs the infinitive zingen to slot E1 in the topology of zullen2. The in-
verse order in (7b) ensues in the case of selection of verb frame zullen1, with the main
verb in slot M5 (zingen zullen). The past participle gezongen that complements the per-
fect auxiliary hebben in (7c/d), has two landing site options: M5 and E1.

The six examples in (7e-j) include all permutations of zullen, gezongen, and hebben. They are acceptable only if zullen precedes hebben; the position of gezongen seems ir-
relevant. Zullen2’s CoMPlement unifies successfully with hebben, forcing the latter to land in E1 and to trail behind its governor. Past participle gezongen is offered two possi-
ble landing sites: M5 and E1. Figure 10 portrays the latter option. If gezongen selects M5, the verb cluster surfaces as zullen gezongen hebben. This presupposes, in agreement with
the middle option in the topology sharing rule in Table 2, that zullen2 does NOT share its M5 slot with that of its complement hebben. If zullen2 chooses the other option and
does share its M5 slot with that of its complement, gezongen will end up as the first
member of the verb cluster.

The IPP effect exemplified in (7m-q) is due to the interaction between hebben and kunnen/zullen at unification. (Hadden is the past tense plural of hebben.) Hebben and kun-
nen2/zullen2 are successful unification partners, given that hebben is the governor and a verb raiser its complement. We assume that the IPP effect is restricted to VR comple-
ments of hebben. However, the effect also occurs — optionally — when hebben takes a complement clause headed by a Third Construction verb with all non-verbal constitu-
ents promoted. Sentence (8) is an example.

(8)   a. ... dat zij dit lied hebben proberen/geprobeerd te zingen
      that they this song have try/tried to sing
   ‘... that they have tried to sing this song’
  b. ... dat zij hebben *proberen/geprobeerd dit lied te zingen

We can handle this by granting verbs like proberen the double status of Third Construc-
tion verb and Verb Raiser:

(9)   [CMPtype=\{\text{M5, E1}-VT-Infin, E2-VT-(\text{Fin, PInfin}), E1-VR-PInfin}\}]

If the latter option is chosen, the infinitive proberen (heading the complement clause
landed in E1 of hebben’s topology) is obligatory, and the topology sharing rule forces

---

8 In order to distinguish different verb frames that belong to the same verb, we use numerical suffixes,
e.g. zullen1 and zullen2. They should not be confused with numerical subscripts that in sample sen-
tences refer to verb hierarchy level (e.g. zullen1 and zullen2). The numerical suffixes serve to differen-
tiate between lexical frames; they do not imply that the meanings associated with these frames are
also different.
promotion of all non-verbal constituents: see (8b). The grammatical alternative in (8b) with *geprobeerd* is analyzable as a Third Construction without promotion of its dependents.

The verb *hopen* in (7r-t) takes a Third Construction complement at slot E2. All other placements are illegal. E2 is also the only possible landing site for complements of VP Extraposition verbs (VE; not illustrated here).

Figure 9. Partial lexical frames for the verbs in (7). The CMPtype features, which control the placement of finite and non-finite clauses, have been printed between square brackets below the S-nodes.
Figure 10. Analysis of sentences (7e–j). The complement headed by *gezongen* selects position E1 from the two alternatives licensed by the unification of its root S with the CMP-S of *hebben*. (The legal M5 option, yielding *gezongen hebben*, is not depicted here.) Sharing of slots M4 and M5 in the middle topology is optional. If *zullen* selects the sharing option for M5, and *gezongen* indeed goes to M5 of *hebben*, the sequence *gezongen zullen hebben* results, i.e. example (7g). The black dots denote unification with substitution of S-nodes; the CMPtype values selected by unification have been underlined. Remember that "V" abbreviates for "V(E,F,R,S,T)".

Examples (7u–y) feature the past tense *zouden* of *zullen* because this form is better suited to bring out the *irrealis* meaning; this has no consequences with respect to word order. The various positions of *gezongen* depend on the level(s) at which a topology shares slot M5 with its upstairs neighbor — through the same promotion mechanism that is illustrated in example (7g) and Figure 10. The linear structure of the verb cluster of (7x) looks as follows:
We propose a similar approach to the placement of particles of so-called separable verbs. A case in point is sentence (10), which is identical to (7u) except for the particle mee ‘together’ preceding the past participle gezongen (meezingen = to sing together).

(10)  ... dat ze deze liedjes zouden, kunnen, hebben, meegezongen

The particle can be placed at any position in the verb cluster as long as it precedes the verb it belongs to (zingen). Since zouden, kunnen and hebben are only allowed to occur in this order while gezongen may occupy four different positions, there are ten different permutations. This is exactly the set licensed by the promotion mechanism:

• the CMPtype features in the three auxiliaries guarantee they occur in the correct order,
• gezongen may follow or precede its governor hebben, and may even precede the other auxiliaries because the sharing configurations permit a promotion path to slot M5 in any of the three higher topologies, and
• since the PaRTicle can only land in an M4 slot while it may be promoted into any of the higher topologies, it will always precede gezongen.

In the sentences discussed so far, topology sharing was restricted to the left-periphery of the clausal topology. Cases such as (11a), however, with ‘extraposed’ constituents, require right-peripheral topology sharing as well. The rule for right-peripheral topology sharing is stated in (12), which partly repeats Table 2. The verb beloven ‘promise’ precedes its governor zal while its complement de fiets te repareren is clause-final (Figure 11). This discontinuity is due to the fact that beloven is a Third Construction verb and does not fit out its complement with a license to land in M5 or E1 (i.e. the CMPtype feature of the complement S-node does not include a reference to M5 or E1). Instead, beloven directs its complement to extraposition slot E2.

(11)  a. ... dat Jan beloven zal de fiets te repareren
     that Jan promise will the bike to repair
     ‘... that Jan will promise to repair the bike’

     b. *... dat Jan de fiets te repareren beloven zal

     c. ... dat Jan zal beloven de fiets te repareren
d. \( \text{... dat Jan beloven zal aan Marie de fiets te repareren} \)

‘... that Jan will promise (to) Marie to repair the bike’

\[
\begin{array}{cccccccc}
\text{M1} & \text{M2} & \text{M3} & \text{M4} & \text{M5} & \text{M6} & \text{E1} & \text{E2} \\
\hline
\text{dat} & \text{Jan} & \bullet & \bullet & \text{zal} & \bullet & \bullet \\
\hline
\end{array}
\]

\( \text{beloven} \) \quad \text{aan Marie} \\
\( \text{de fiets} \) \quad \text{te repareren} \\

(12) \text{Right-peripheral topology sharing in complement clauses of Dutch and German}

a. Right-peripheral topology sharing is restricted to E2 slots (\( \text{RS}=1 \) in Table 2)

b. Lexical frames headed by a Verb Raiser \textit{obligatorily} share slot E2 of their topology with that of their complement (\( \text{RS}=1 \)).

c. Lexical frames headed by a VP Extraposition or a Third Construction verb \textit{optionally} share slot E2 of their topology with that of their complement (\( \text{RS}=0:1 \)).

d. A finite complement clause does not right-peripherally share its E2 slot with that of its governor (\( \text{RS}=0 \)).

e. If CoMPlement clauses from several clausal topologies land in the same E2 slot, they line up in increasing clause depth order.

\[
\begin{array}{cccccccc}
\text{CP} & \text{SUBJ} & \text{NP} & \text{M5} & \text{S} & \text{HD} & \text{M6} & \text{v} \\
\hline
\text{dat} & \text{Jan} & \bullet & \bullet & \bullet & \bullet & \text{zal} & \text{de fiets te repareren} \\
\hline
\text{M6} & \text{v} \quad \text{beloven} \\
\hline
\end{array}
\]

Figure 11. Right-peripheral topology sharing (shaded area E2) in sentence (11a)

The CMPtype features in the lexical frames of the modal auxiliaries in Figure 9 also account for the ordering phenomenon illustrated in (13) below. The order of modal auxiliary and main verb in clause-final verb clusters is arbitrary as long as the main verb does not take a sentential complement. If the auxiliary governs a complement-taking verb, Aux-MainV is the only acceptable sequence: see the grammaticality contrast between (13a/b) on the one hand and (13c/d) on the other.

(13)

a. \( \text{... dat Jan de fiets zal proberen te repareren} \)

‘... that Jan will try to repair the bike’

b. \( \text{*... dat Jan de fiets proberen zal te repareren} \)

c. \( \text{... dat Jan de fiets zal (uit)proberen} \)
... that Jan will try (out) the bike'

d. ... dat Jan de fiets (uit)proberen zal

'I don't know which bike Jan will try to repair'

e. *Ik weet niet welke fiets Jan zal proberen te repareren

'... that Jan will try to repair'

f. *Ik weet niet welke fiets Jan proberen zal te repareren

'I don't know which bike Jan will try (out)'

g. Ik weet niet welke fiets Jan zal (uit)proberen

'... that Jan will try (out)'

h. Ik weet niet welke fiets Jan (uit)proberen zal

The lexical frames for zullen1 and kunnen1 have a much narrower range of applicability
than their counterparts zullen2 and kunnen2. They can head finite clauses only, licensing
MainV-Aux order exclusively to ‘simple’ verbs, which do not take sentential comple-
ments. The phenomenon holds irrespective of whether the dependent NP is promoted
within the Midfield (de fiets in M3) or within the Forefield (welke fiets in F1) — cf. (13e-f).

A construction of Dutch which has no counterpart in German or English, is governed
by verbs such as zitten ‘sit’, staan ‘stand’ and liggen ‘lie’. They take an infinitival com-
plement expressing an ongoing action performed by the referent of the SUBJect NP (cf.
Progressive aspect in English) and simultaneously mention the bodily position assumed
by the SUBJect referent during the action. The examples in (14) demonstrate, interest-
ingly, that the status of the complement’s HeaD verb varies in function of the status the
governor: If the governing verb is finite, the complement has PInfin status (with te); in
all other cases, the complement's HeaD verb is a bare infinitive without te (status = In-
fin).

(14) a. ... dat Jan een boek zit/staat/ligt te lezen

    that Jan a book sits/stands/lies to read

    ‘... that Jan is reading a book (while sitting/standing/lying)'

b. ... dat Jan een boek zou zitten/staan/liggen lezen

    ‘... that Jan would be reading a book'

c. ... dat Jan een boek heeft zitten/staan/liggen lezen

    ‘... that Jan has been reading a book'

We propose to treat these complement-takers in a way similar to the modal verbs in
Figure 9 (e.g. kunnen and zullen), that is, to associate with them separate lexical frames
for use in finite and infinitival constructions. (The verbs cannot take participial status.)
Figure 12 shows the CMPtype features.
We finally turn to verb clustering in passive constructions, as exemplified by the examples (15) and (16), which are more or less synonymous. (15a-f) list all six permutations of two past-participles (aangeboden and geworden) and the finite passive auxiliary was. Three of these are ungrammatical. Version (15g) is the sole (marginally) acceptable ordering where the particle aan and the main verb geboden are not contiguous. The past-participle geworden is nearly always omitted in modern standard Dutch. So, in everyday usage, the verb cluster has three possible orderings: was aangeboden, aangeboden was and aan was geboden. The grammaticality ratings of the seven versions of (16) run parallel to their counterparts in (15). We take this as evidence for a parallel treatment of aangeboden gekregen had and aangeboden geworden was. That is, we analyze not only geworden but also gekregen as a complement-taking verb, and they both can take aangeboden as non-finite complement. The lexical frames of the verbs involved are depicted in Figure 13. (The entry for had ‘had’, the past tense of hebben ‘have’, has been copied from Figure 9; zijn ‘be’ is the infinitival form of was.) The reader may verify for her/himself that, in conjunction with the cross-clausal topology sharing rule (Table 2), these lexical frames generate exactly the set of (marginally) acceptable verb orderings.

(15) a. ... dat deze baan haar niet aangeboden, (geworden) was, that this job her not PRT-offered been was ‘... that this job had not been offered to her’
b. ... aangeboden was (geworden)
c. "... (geworden) was aangeboden
d. "... (geworden) aangeboden was
e. *...was (geworden) aangeboden
f. ... was aangeboden (geworden)
g. ?... aan was geboden (geworden)

(16) a. ... dat ze deze baan niet aangeboden, gekregen, had, that she this job not PRT-offered gotten had ‘... that she had not been offered this job’
b. ... aangeboden had gekregen
3. Verb clustering in German

Our treatment of word order in German clauses is based on the topology slot selection rules in Table 1, the topology sharing rule in Table 2, and CMPtype features such as those listed in the lexical items of Figure 14 and Figure 16. The German slot selection rules closely resemble the Dutch ones, the main difference being that the PaRTicle of separable verbs goes to M6 rather than M4. The left-peripheral topology sharing rule is essentially the same as the one for Dutch, except that Verb Raisers like wollen do not want their complements to share slot M5. As to right-peripheral sharing, we assume that the Dutch rule in (12) above also applies to German.

In designing the German linearization system we were guided by the numerous facts discussed by Kathol (2000) and Meurers (2000). They present detailed surveys of
both traditional and contemporaneous linguistic proposals for dealing with word order in German, and develop their own HPSG-style solutions. Other publications we consulted include Uszkoreit (1987); Engelkamp, Erbach & Uszkoreit (1992); Reape (1993); Haider (1993); Nerbonne, Netter & Pollard (1994); Hinrichs & Nakazawa (1994, 1998); Seuren (1996); Bouma & Van Noord (1998); Richter (2000), and Seuren (this volume). The confines of the present paper do not allow us to describe in detail the numerous phenomena and concomitant theoretical explanations in these sources.

Figure 15 illustrates how the value of CMPtype features constrains linear order within the verb clusters in (17). Figure 16 shows the CMPtype feature values of the lexical frames referenced in (18) — examples mostly due to Seuren (this volume). Figures 17 and 18 illustrate the analyses of four of them. The notational conventions are identical to those for Dutch (see previous Section). Notice that the figures do not contain lexical frames for past participles of modal auxiliaries. We have assumed that, in standard German, forms like gekonnt 'been able' and gewollt 'wanted' do not subcategorize for non-finite clausal complements. Thus we can account for the unacceptability of examples like (17j) and (18c).

(17)  

a. \( \ldots \text{dass sie das Lied singen, k"onnen,}/werden, \)  
that they the song sing be-able/will  
\( \ldots \text{that they can/will sing the song} \)
b. *\( \ldots \text{dass sie das Lied k"onnen,}/werden,\) singen  
that they the song sing be-able will  
\( \ldots \text{that they will be able to sing the song} \)
c. \( \ldots \text{dass sie das Lied singen, k"onnen,}/werden, \)  
that they the song sing be-able will  
\( \ldots \text{that they will be able to sing the song} \)
d. \( \ldots \text{werden, singen,}\) k"onnen  
\( \ldots \text{they will be able to sing} \)
e. *\( \ldots \text{singen, werden,}\) k"onnen  
\( \ldots \text{they can sing be-able} \)
f. *\( \ldots \text{werden, k"onnen,}\) singen  
\( \ldots \text{they will be able to sing} \)
g. *\( \ldots \text{k"onnen, werden,}\) singen  
\( \ldots \text{they will be able to sing} \)
h. *\( \ldots \text{k"onnen, singen,}\) werden  
\( \ldots \text{they will be able to sing} \)
i. \( \ldots \text{haben, singen,}\) k"onnen  
\( \ldots \text{they will have been able to sing} \)
j. *\( \ldots \text{singen, gekonnt,}\) haben  
\( \ldots \text{they will have been able to sing} \)

(18)  
a. \( \ldots \text{dass sie ausgehen, will,}\)  
that she out-go wants  
\( \ldots \text{that she wants to go out} \)
b. ... dass sie hat, ausgehen, wollen, 
  that she has out-go want
  ‘... that she has wanted to go out’

c. *... dass sie ausgehen, gewollt, hat, 
  that she out-go wanted has
  ‘... that she has wanted to go out’

d. ... dass sie mich ausgehen, sah, 
  that she me out-go saw
  ‘... that she saw me go out’

e. ... dass sie mich hat, ausgehen, sehen, 
  that she me has out-go see
  ‘... that she has seen me go out’

f. ... dass sie mich ausgehen, gesehen, hat, 
  that she me out-go seen has
  ‘... that she has seen me go out’

g. *... dass sie wird, ausgehen, 
  that she out-go will
  ‘... that she will go out’

h. ... dass sie ausgehen, wird, 
  that she out-go will
  ‘... that she will want to go out’

i. ... dass sie wird, haben, ausgehen, versuchen, wollen, 
  that she will have out-go try want
  ‘... that she will have wanted to try to go out’

k. *... dass sie wird, haben, versuchen, wollen, ausgehen, 
  that she will have wanted to try to go out

l. ... dass ich sie habe, tanzen, gehen, lassen, 
  that I her have dance go let
  ‘... that I have let her go dance’
Figure 14. The CMPtype features which control the placement of complement clauses in the lexical frames for the verbs in (17).

Figure 15. Left-hand side: unification in the verb cluster of examples (17c-h). Right-hand side: idem for (17i). **W**erden licenses both M5 and E1 as legal landing sites for its VR complement, yielding *werden singen können* and *singen können werden*. The order *singen werden können* ensues if the upper and middle topologies share their M5 slots. Of the two lexical entries for **haben**, only **haben2** can unify with infinitival **können**.
Figure 16. Partial lexical entries for the verbs in examples (18).
The lexical entry of the future auxiliary *werden* in Figure 14 assumes — in line with a suggestion by Meurers (2000, p. 223) — that this verb can only occur in finite form. This property accounts, among other things, for the grammaticality contrast between (19a) and (19b). *Können* but not *werden* is allowed to head the complement of *haben*₂, as shown in the CMPtype features of these auxiliaries (Figure 14). Furthermore, *haben*₂ is not allowed to govern a complement clause with PInfin status. The verb *behaupten* — similarly to *versuchen* (Figure 16) — licenses slot E₂ but not E₁ as a landing site for its complement. This rules out (19d) while (19c), with *haben*₁, is unproblematic.

\[(19)\]

\(\text{a.  } \ldots \text{ dass } \text{er } \text{es } \text{hätte}_{1} \text{singen}_{3} \text{können}_{2}\)

\(\text{b. } \ast \ldots \text{ dass } \text{er } \text{es } \text{hätte}_{1} \text{singen}_{3} \text{werden}_{2}\)

\(\text{c. Er behauptete, es gesungen}_{2} \text{zu haben}_{1}\)

he claimed it sung to have

‘He claimed to have sung it’

\(\text{d. } \ast \text{Er behauptete, es zu haben}_{1} \text{singen}_{3} \text{können}_{2}\)

he claimed it to have sing be-able-to

(intended) ‘He claimed to have been able to sing it’
The latter observation relates to the fact that haben2 cannot head the non-finite complement of a preposition or a noun (Haider, 1993). While (20a) below, with a finite complement of ohne ‘without’, is perfectly grammatical, the IPP version (20b) is ruled out. Ohne can take a non-finite complement headed by haben2, as illustrated by (20c). Example (20d), with the complement-taking noun Gelegenheit ‘opportunity’, is ill-formed for the same reason as (20b). We can account for these phenomena on the assumption that CMP-S nodes of nouns and prepositions contain a CMPtype feature that specifies slot E2 as the only possible landing site for non-finite complements, thus preventing unification of these CMP-S nodes with the haben2 frame (see Figure 19).\footnotetext[10]{Interestingly, if we make the same CMPtype assumption for CMP-S nodes of Dutch nouns and prepositions, the Dutch counterparts of (20a) and (20d) are predicted to be grammatical: The CMPtype}

---

\textsuperscript{9} Interestingly, the Dutch translation equivalent of (19b), with clause-final verb cluster had, zullen, zingen\textsubscript{2}, is perfectly acceptable. It requires the lexical frame zullen\textsubscript{2} rather than zullen\textsubscript{1}. See example (7o) above.

\textsuperscript{10} Meurers (2000, p.70-71) argues that this construction occurs frequently enough to consider it well-formed, contra Haider (1993) who does not accept it.
Various scrambling phenomena in German involve fronting of constituents — in our terminology: placement of focused or topicalized constituents in slot F1 of a main clause. Examples (21a) through (21f), from Engelkamp, Erbach & Uszkoreit (1992), represent Simple Fronting: slot F1 hosts exactly one constituent — a single NP, AdvP, Head verb, or clause. In cases of Complex Fronting, slot F1 seems to accommodate several phrases, as exemplified by (21g) and (21h), also from Engelkamp et al. (1992). Such cases seem to challenge the widely held view that in main clauses of German no more than one constituent is allowed to precede the finite verb. The mechanism of left-peripheral topology sharing can solve this problem on the assumption that the focus/topic relations within the complement clause overrule the topology sharing defaults stated in Table 2. Auxiliaries like wollen and sollen standardly cause sharing of slots F1 through M4 of the complement topology. This default presumably does not apply to focused or topicalized constituents of the complement. These do not undergo promotion and are
fronted while the promoted parts of the complement are slotted into the standard Mid-field positions of the matrix topology\(^{11}\).

(21) a. Der Kurier sollte nachher einem Spion den Brief zustecken
   the courier should later a spy the letter slip
   *The courier was later supposed to slip a spy the letter.

b. Zustecken sollte der Kurier nachher einem Spion den Brief

c. Den Brief sollte der Kurier nachher einem Spion zustecken

\[
\begin{array}{cccccc}
F1 & M1 & M2 & M3 & M5 & M6 \\
\bullet & \text{sollte} & \text{der Kurier} & \bullet & \bullet & \bullet \\
\end{array}
\]

d. Einem Spion sollte der Kurier nachher den Brief zustecken

e. Nachher sollte der Kurier einem Spion den Brief zustecken

f. Nachher einem Spion den Brief zustecken sollte der Kurier

g. Einem Spion den Brief zustecken sollte der Kurier nachher

\[
\begin{array}{cccccc}
F1 & M1 & M2 & M3 & M4 & M6 \\
\bullet & \text{sollte} & \text{der Kurier} & \bullet & \bullet & \bullet \\
\end{array}
\]

h. Den Brief zustecken sollte der Kurier nachher einem Spion

\[
\begin{array}{cccccc}
F1 & M1 & M2 & M3 & M4 & M6 \\
\bullet & \text{sollte} & \text{der Kurier} & \bullet & \bullet & \bullet \\
\end{array}
\]

Complex Fronting may involve a stack of non-finite complement clauses. The sentence in (22a), from Kathol (2000), contains two complements headed by the verbs *können* and *finden*. The seven logically possible variants embodying simple or complex fronting are shown in (22b) through (22h). These are all grammatical, except for (22d) and (22f). When drawing the linear structures, we have assumed that the complements either go to slot M5 of their governor or, if focused or topicalized, to F1\(^{12}\). All five grammatical cases can be generated by our linearization system. With respect to (22e) we assume, like we did in the context of some of the examples in (21) above, that focus/topicalization can prevent promotion of phrasal arguments (here the Direct OBJect

\(^{11}\) Table 1 does not specify landing sites for Adverbial Phrases. Here, we assume that the adverbial modifier *nachher* is assigned a late position in slot M2, unless focused or topicalized.
The ill-formedness of variants (22d) and (22f) follows from the fact that können cannot be fronted without taking finden along: if finden is not going to F1, it can only land in M5 — a slot that Verb Raisers like können never share with their complement. Interestingly, the system does not rule out a (marginally) acceptable case such as (23), from Kathol (2000, p. 234). The basic reason is that the complement of versprochen, unlike the complement of können, is free to land in E2 as well as in M5.

12 The lexical item werden, shown in Figure 14, also allows its complement to land in slot E1. This additional possibility does not affect our account of the examples in (22).
f. *Das Buch können wird Peter finden

```
F1  M1  M2  M3  M5  M6
• wird Peter  • können
↑     ↑
das Buch  ↑  finden
```

g. Finden können wird Peter das Buch

```
F1  M1  M2  M3  M5  M6
• wird Peter  • können
↑      ↑
das Buch  ↑  finden
```

(23) ?Versprochen wird er ihr nicht haben den Wagen zu waschen

promised will he her not have the car to wash

‘He will not have promised her to wash the car’

We now turn to some special German word order phenomena whose explanation exceeds the scope of the PG framework developed so far but may be accommodated by slight extensions of the formalism. The first example is provided by the Verbal Complex Split, a construction that seems to be acceptable to many speakers of German. Kathol (2000, pp. 204-205) gives four illustrations, one of them sentence (24a)\textsuperscript{13}. They all feature the auxiliaries \textit{werden} or \textit{haben} (IPP) in the role of finite verb. It is clear that the current linearization model only generates the variant (24b). In order to enable the Verbal Complex Split, we need to add a special subcategorization property to the lexical items \textit{werden} and \textit{haben}, namely, that they license the embedded clause to share slot M5 — over and above the slots F1 through M4 that are already shared by virtue of the default rule in. The clause S3 headed by \textit{bestehen} will then get promoted into to S1,
there preceding hat/wird in slot M5. However, because an extension of the model along this line may have undesirable side-effects, and in view of the insecure status of the Verbal Complex Split as a construction of standard German, we have not explored this approach any further. (Interestingly, the modified version of the German cross-clause topology sharing rule resembles the Dutch one more closely than the original one; see Table 2.)

(24) a. … dass er das Examen bestehen wird/hat können
   ‘…that he will be/has been able to pass the exam

   \[\begin{array}{cccccc}
   & M1 & M2 & M3 & M5 & M6 & E1 \\
   \hline
   \text{dass} & \text{er} & \cd & \cd & \cd & \cd & \cd \\
   \cd & \cd & \cd & \text{können} & \cd & \cd & \cd \\
   \cd & \cd & \cd & \text{das Examen} & \cd & \cd & \cd \\
   \end{array}\]

   b. … dass er das Examen wird/hat bestehen können

   \[\begin{array}{cccccc}
   & M1 & M2 & M3 & M5 & M6 & E1 \\
   \hline
   \text{dass} & \text{er} & \cd & \cd & \cd & \cd & \cd \\
   \cd & \cd & \cd & \text{hat/wird} & \cd & \cd & \cd \\
   \cd & \cd & \cd & \text{das Examen} & \cd & \cd & \cd \\
   \end{array}\]

The rule for left-peripheral topology sharing in Table 2 states that Verb Raisers like wollen and other auxiliaries cause their complements to share the slots F1 through M4. Because, in subordinate clauses, the finite Verb Raiser and the governed non-finite verb land in adjacent slots (the Verb Raiser itself in M6, the HeaD verb of the governed clause in M5 or E1), it follows that the region M5 through E1 of the subordinate clause can only be populated by verbs. This implication is falsified by examples like (25), from Kathol (2000, p. 233), where geben’s Direct OBJect interrupts the verb cluster. Such cases, although relatively rare, call for a somewhat more lenient version of the topology sharing rule for Verb Raisers — one that offers a wider range of Midfield locations where to put the boundary between shared and non-shared constituents than just at the end of slot M4. It is not clear, however, which conditions control the exact position of the boundary.

---

13 Kathol (o.c.) refers to an unpublished workshop paper by Meurers (1994) as the original source of these examples. See Meurers (2000) for additional examples.
We have presented a linearization formalism capable of capturing a broad range of verb clustering phenomena in Dutch and German. It is part of the psycholinguistically motivated formalism of Performance Grammar, which has separate but interrelated components for assembling the hierarchical and the linear structure of sentences. Particularly importantly in the present context, PG's linearization component deals not only with 'horizontal' position assignment (left-to-right arrangement of constituents within clauses) but also with a broad class of (vertical) 'movement' (constituents being left-right arranged at a clause level higher than their level in clause hierarchy). Unlike what happens in other grammar formalisms, these movement types take place without affecting the hierarchical structure. (Another class of movement phenomena, we assume, does affect the clause hierarchy, in particular the PG equivalents of Subject-to-Subject and Subject-to-Object Raising. The distinction between movement classes at issue here corresponds to the distinction between $\bar{A}$- and A-movement in Chomskyan Generative Grammar.) Responsible for this behavior are a data structure called topology and the mechanism of lateral topology sharing. A topology is a one-dimensional array of positions ('slots') serving as landing sites for syntactic constituents. For every pair of a grammatical category (NP, PP, V, etc.) and a grammatical function (Subject, Object, Head, etc.), there is a set of one or more landing sites. The topology sharing mechanism causes unification of left- or right-peripheral slots of clausal topologies at adjacent levels of the clause hierarchy, and thereby virtual vertical movement of constituents landed in a shared slot.

The treatments we proposed for the two target languages are uniform both from a cross-linguistic and from an intralinguistic perspective (see Harbusch & Kempen, 2002, for a treatment along the same lines of movement phenomena in English). The topologies, the topology sharing rules, the constraints on topology sharing, and the CoMPle-
ment-type features are very similar across languages. Word order contrasts between the target languages largely reduce to a few narrowly confined and relatively minor differences concerning topology slot fillers and topology sharing rules. The most salient contrasts between Dutch and German verb clusters are caused by lexical factors — the detailed composition of the CMPtype value in the S-nodes of verb frames.

Acknowledgement

We are indebted to Camiel van Breugel (Leiden Institute for Advanced Computer Science, Leiden University) for implementing, in his Performance Grammar Workbench, a PG for Dutch that includes the rule system described in Section 2. The PGW is accessible through the internet and can be run as an applet in JAVA-enabled browsers (www.liacs.nl/home/cvbreuge/pgw).

References


