# Word order scrambling as a consequence of incremental sentence production

**Gerard Kempen**
Psychology Department, Leiden University
Max-Planck-Institute for Psycholinguistics, Nijmegen
The Netherlands

&

**Karin Harbusch**
Computer Science Department
University of Koblenz-Landau
Germany

**Draft, 13th July 2001**

To be published in Haertl, H., & Tappe, H. (Eds.), *Mediating between concepts and language — Processing structures*. Berlin: Mouton De Gruyter.

## 1. Introduction

The incremental nature of human grammatical encoding is an important source of word order variation even in languages with relatively strict word order. A well-known example is the 'heavy constituent shift' that moves a long phrase, wholesale or in part, to a clause-final position. It seems reasonable to assume that longer constituents on average need more assembly time than shorter ones. An incremental grammatical encoder can make for greater output fluency by assigning a heavy exemplar of such a constituent a late position in the utterance. Conversely, output fluency is promoted also when the encoder allows a light constituent to occupy an early position, e.g. a pronoun referring to the discourse topic. Stated more generally: given a language with some word order flexibility and an incremental grammatical encoder, constituents whose shape is determined at an earlier point in time will tend to precede constituents that 'arrive' later. The correlation between the arrival time of a constituent (the moment its shape is fixated) and its linear position tends to be stronger the greater the amount of flexibility. Arrival time may depend not only on syntactic factors such as the complexity of the syntactic assembly process but also on lexical factors (e.g., different retrieval times for low- and high-frequency words) and on semantic factors (e.g., salient fragments of the to-be-expressed meaning being conceptualized prior to less salient ones; see Yamashita & Chang (2001)). Now suppose that, for some pattern of word order variation in a (semi-)free word order language, there is independent evidence that the actual order of constituents covaries directly with their arrival times. This would eliminate the need for *syntactic* rules that explicitly control the order of the constituents involved — thereby benefiting theoretical parsimony.

In this chapter we test the viability of this approach by developing a statistical model that generates a detailed pattern of constituent order variation on the basis of the hypothetical arrival times of the constituents. Our test case is the well-known 'scrambling' phenomenon of Subject (*S*), Indirect Object (*I*) and Direct Object NP (*O*) in the 'Mittelfeld' (Midfield) of German clauses. None of the six possible permutations of these NPs are definitely ruled out, although the grammaticality (acceptability) ratings they elicit tend to vary widely. We show that simple model assumptions about the typical arrival times of *S*, *I* and *O* yield accurate estimates of the acceptability of the permutations. Actually, the predictions of our model are at least as accurate as those derived from models based on ranked or weighted ordering constraints, e. g. within an Optimality Theoretical framework. We conclude that our incrementality-based approach presents a viable alternative to current approaches in terms of a hierarchy of ordering constraints, and that satisfactory accounts of the scrambling phenomenon under scrutiny may well be found outside the domain of explicit syntactic ordering rules.

Modeling 'word order freedom in restraint' is relevant not only to psycholinguistic theories of grammatical encoding but also to the design of computational sentence generators striving for natural and varied output (e.g., in a computer-supported language training environment). Such systems should neither select the same permutation at all times, nor produce the various grammatical permutations on a strictly random basis. Instead, they must be sensitive to empirical data that reflect speaker ordering habits and preferences, and select permutations accordingly.

In the grammatical encoding model presented below, we apply *Performance Grammar* (*PG*), a psycholinguistically motivated grammar formalism containing separate components generating, respectively, the hierarchical and the linear structure of sentences. In Section 2 we outline the psycholinguistic motivation underlying this distinction. Sections 3 and 4 present the essentials of PG's hierarchical and linearization components. Section 5 introduces the experimentally obtained acceptability ratings for permutations of Subject, Direct Object and Indirect Object NPs in the Midfield of German clauses. In Section 6 we account for these ratings in terms of a probabilistic model. Section 7, finally, summarizes our approach and the conclusions we reached.

## 2. A psycholinguistic argument for topology-based linearization

In an *incremental* grammatical encoder, word order can be affected by the 'order of arrival' of the various syntactic constituents. Constituents that become available for being ordered at an earlier point in time, may precede constituents emerging later — as long as grammar rules do not intervene. Order of arrival is controlled by several groups of factors: by pragmatic and conceptual factors residing in a 'conceptualizer' component and/or in the 'semantic-syntactic' interface; and by lexical and syntactic factors in the grammatical encoder ('formulator') itself. For instance, 'old' information can be conceptualized more easily than 'new' information and therefore tends be available for being linearized earlier. A word finding problem may delay the formulation of a conceptual fragment; 'heavy' constituents take more assembly time than 'light' constituents. Linearization methods for (semi-)free word order languages should be responsive to order of arrival of syntactic constituents.

Furthermore, we consider that linearization methods should comply with general psychological properties of sequence generation processes. Models of serial order are conveniently divided into two types depending on whether or not they aim at generating novel sequences that have never been seen before (Dell, Burger & Svec (1997)). ‘Closed’ models only deal with a restricted and invariable set of sequences. ‘Open’ models can handle potentially unlimited sets of sequences, including novel ones. In open models, the representation of a to-be-generated sequence is typically split into two parts: (1) a one-dimensional array containing a number of empty slots, and (2) the set of items to be inserted into the slots. Items are ordered by *binding* them to a slot. Generating a sequence involves traversing the array from beginning to end, at each slot reading out any item(s) bound to it.

Open sequence generation models are informally called slot-and-filler models. In this paper, we use the term *topology* to refer to a row of slots. Filler items are terminal nodes of an unordered syntactic tree called *mobile*. In the simplest case, the mobile consists of two layers: the bottom layer specifying the filler items, which are all connected to a single root node in the top layer. In more complicated cases, the mobile may span additional layers of nodes. In a typical model application, the ‘lexicon’ of filler items contains many different entries. However, these entries belong to a small number of classes, and the item-to-slot binding process is sensitive to class membership. That is, certain slots only accept items of certain classes.

Already in 1975, Merrill Garrett proposed a slot-and-filler model for syntactic speech errors that involve item miss-orderings of words or phrases (as in *Although murder is a form of suicide*). He had discovered that the likelihood of such errors is independent of the distance between the permuted elements in the surface string, whereas other types of exchanges (e.g. between phonemes) occur predominantly between elements in neighboring positions. He took this contrast as evidence for two distinct levels of processing: a ‘functional’ level where grammatical relationships between constituents are established, vs. a ‘positional’ level where constituents are ordered from left to right. This distinction has been adopted, in one way or another, by many students of grammatical encoding (e.g. Kempen & Hoenkamp (1987), Levelt (1989), Bock & Levelt (1994) and Kempen & Harbusch (1998)). In Kempen & Hoenkamp’s version, the constituents of unordered functional structures acquired their linear position by binding themselves to a slot in a ’holder’ which is similar to a topology in more recent literature (see Kathol (2000) in particular).

Before turning to our proposal for a topology-based linearization model, we need to explain the essential features of PG’s hierarchical grammatical structures.

## 3. Hierarchical structures in Performance Grammar

PG's hierarchical component generates unordered trees by combining 4-tiered 'mobiles' called lexical frames. Figure 1 shows the eight lexical frames corresponding to the words of example (1).

(1) *Denkst du, dass er das Auto repariert hat?*
Think you that he the car repaired has
'Do you think that he has repaired the car?'

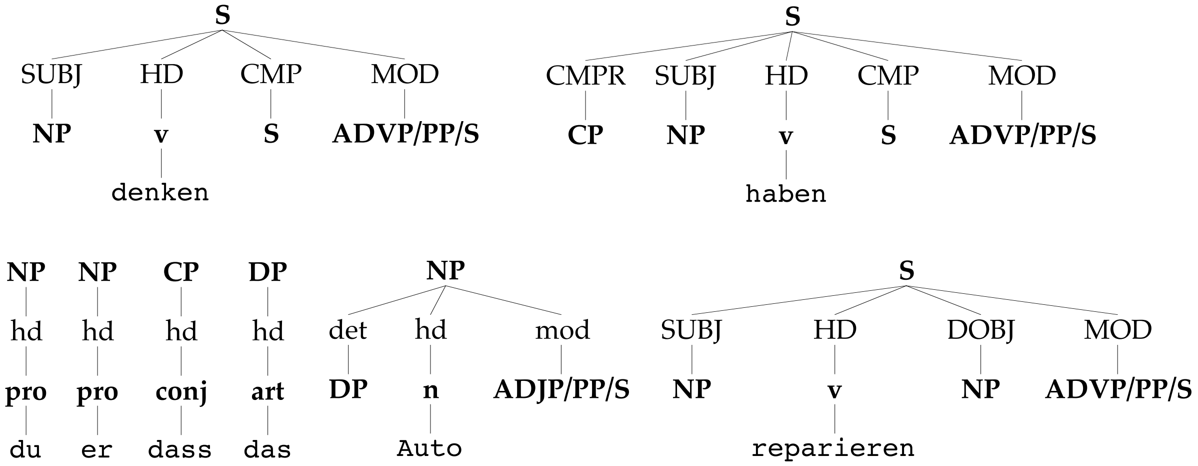


Figure 1. Simplified lexical frames underlying the eight words of sentence (1). Left-to-right order of branches is arbitrary. The basic shape of lexical frames is retrievable from the Mental Lexicon in response to contents of the to-be-expressed conceptual message; however, certain branches (e.g. CMPR)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

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, MODifier). At most one exemplar of a functional node is allowed in the same frame, except for MOD nodes, which may occur several 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 constitutes the fourth layer and is printed below the lexical node serving as the frame's HeaD.

Categorial nodes (i.e. lexical and phrasal nodes in the first and third layers of a lexical frame) have associated with them a *feature matrix*, i.e., a list of pairs that consist 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 finite, non-zero number of character strings (e.g., {sing}, {1st, 2nd, 3rd}), each representing a possible value of the attribute (disjunctive value sets). Lexical frames are combined to form larger mobiles by a non-recursive form of unification, called *feature unification*. This operation yields a merger of the root node of one frame with one foot node of another frame. As illustrated in Figure 2, unification also serves to select the value of agreement features (e.g. entailing second person singular *denkst* of the verb *denken*).

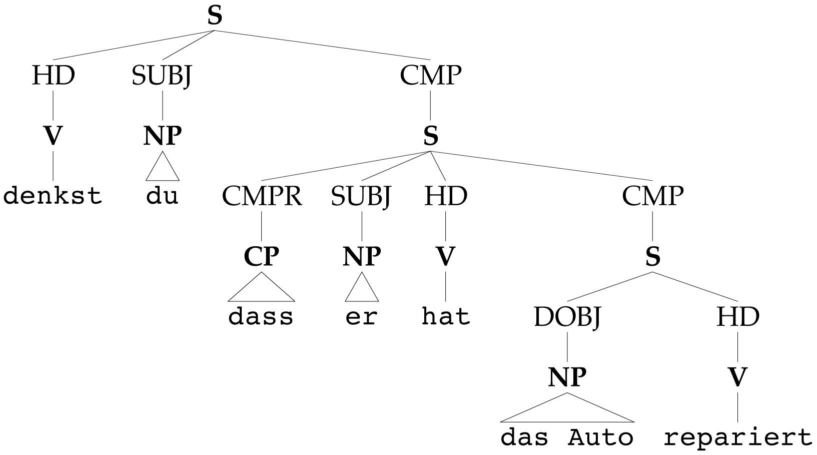


Figure 2. Verb frame hierarchy underlying example (1). The root S-node of the verb frame associated with *hat* and the CoMPlement S-node of *denkst* have merged as a result of unification, and so have the S-CMP of *hat* and the root S-node dominating *repariert*. (In this paper we do not discuss unification of nodes other than S.) Left-to-right order of branches is arbitrary.

## 4. 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 one-dimensional '*linearization array*' specifying a fixed number of positions (or *slots*, *landing sites*) for its constituents. In line with certain traditional grammars of German, we will use the term *topology* to refer to a linearization array. The topology of a verb frame (i.e., of a finite or non-finite clause) allocates 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 German clauses specifies nine different slots, labeled as indicated in Figure 3.

Constituents may be assigned 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'. We assume that these constraints are applied by a finite-state automaton (FSA), called '*linearizer*', which traverses the slots from left to right like a kind of cursor.

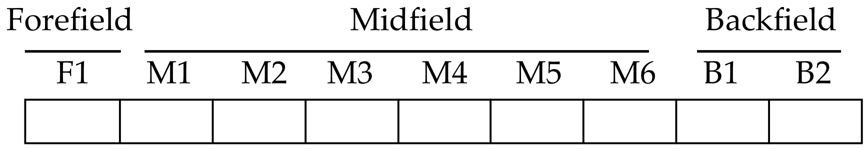


Figure 3. Slot labels used in topologies for German clauses. The terms *Forefield*, *Midfield* and *Backfield* are translations from German *Vorfeld*, *Mittelfeld* and *Nachfeld*, respectively.

As stated in Section 2, the linearizer FSA is supposed to operate as part of an incremental grammatical encoder where different syntactic constituents may 'arrive' at different points in time. That is, the grammatical function and shape of certain phrases may be finalized earlier than that of other phrases. We assume that the various constituents which have to be ordered by the same linearizer, queue up in order of arrival and are treated by the linearizer on a first-in-first-out basis. More precisely, whenever the linearizer enters a slot that serves as landing site for several different types of constituents, it inspects the queue and places the first suitable candidate into that slot. For instance, consider two constituents C1 and C2 to whom the grammar allows the same slot *Sa*, which can accommodate at most one constituent and is located to the right of the linearizer's current position in the topology. When the linearizer enters slot *Sa*, it will select either C1 or C2 as filler for *Sa*, depending on which one is first in the queue. If, furthermore, these constituents are also allowed to land in slot *Sb* located to the right of *Sa*, then *Sb* will be occupied by the other phrase. (In Section 6, we will meet a concrete example.)

Incremental grammatical encoding entails the possibility that, when the linearizer arrives at a certain slot, some constituent that is supposed to land there, is still unavailable. If such a constituent is an obligatory member of the construction (e.g., the finite verb in a finite clause), and the linearizer would not wait until the verb had presented itself, massive ungrammaticality would result. The linearizer therefore has the important duty to check whether the current slot is the landing site of an obligatory phrase; if so, it should postpone jumping to the next slot until the phrase has shown up. Another task of the linearizer is to apply the *linear precedence* function associated with the current slot. This is required if the grammar directs several constituents to the same slot. In sum, the linearizer performs three actions:

- *slotting* (distributing grammatical functions over the slots of a topology),
- *jumping* (moving to the next slot or waiting for an obligatory constituent), and
- *sorting* (applying linear precedence rules within a slot).

Figure 4 shows a somewhat simplified version of the linearization FSA for German clauses that has been proposed by Kempen & Harbusch (2001). The states of the automaton (circles) and their labels correspond to slots of the topology (except for slots that are distinguished by a lower case letter; e.g. M1*a* and M1*b* both correspond to slot M1). Placement conditions are annotated above the arcs; slot fillers are printed in bold below them. E.g., one of the arcs connecting state F1 to state *M1a* says that in case of an interrogative main clause slot F1 serves as the destination for a Wh-constituent. Single bars denote inclusive ORs; double bars are exclusive ORs ('either … or'). For instance, in a declarative main clause, slot F1 can be filled by the SUBJect or a Topicalized (+TOP) NP, but not both. Each of slots M2 through M4 may host at most one constituent: a SUBJect or an (In)Direct OBJect. Forward slashes refer to special conditions on the shape of a constituent. E.g., "CMP-S/non-finite" means "destination for a non-finite CoMPlement S". Underlined constituents denote obligatory slot fillers: that is, the linearizer should not jump to the next state without having placed a constituent of the requested kind. The symbol "<" indicates the precedence relation between multiple constituents sharing a slot.

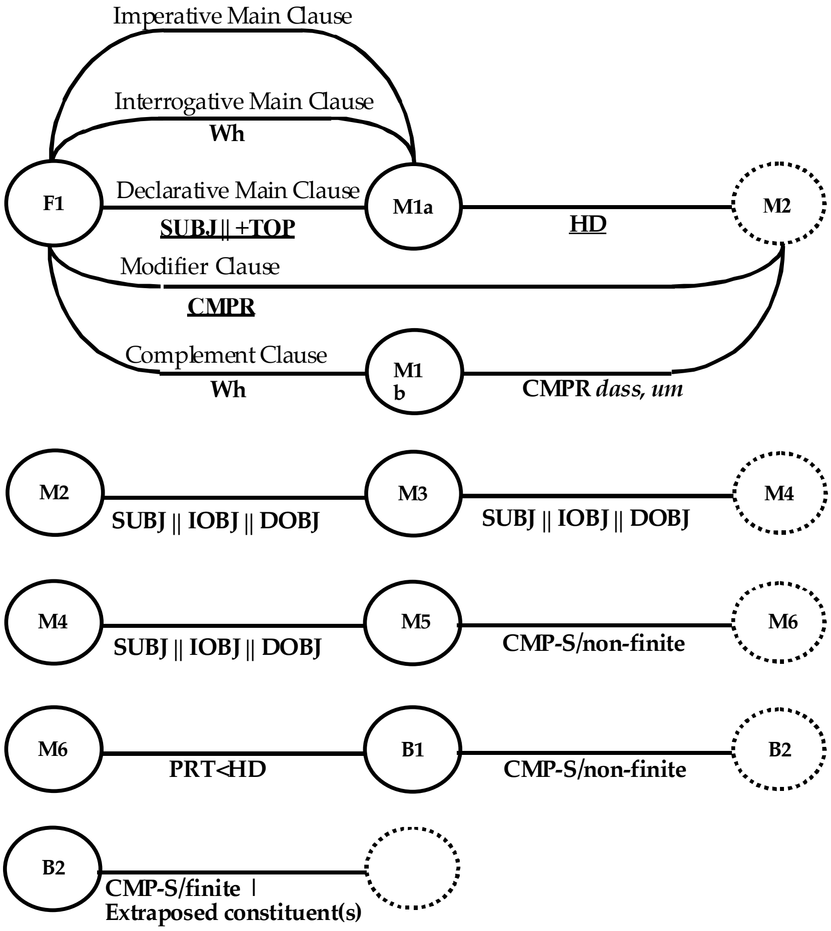


Figure 4. Finite-state automaton for linearizing constituents in German clauses. For explanation see text.

We show linearization at work on an abbreviated version of example (1).

(1') *Denkst du, dass er das Auto repariert?*

'Do you think that he repairs the car?'

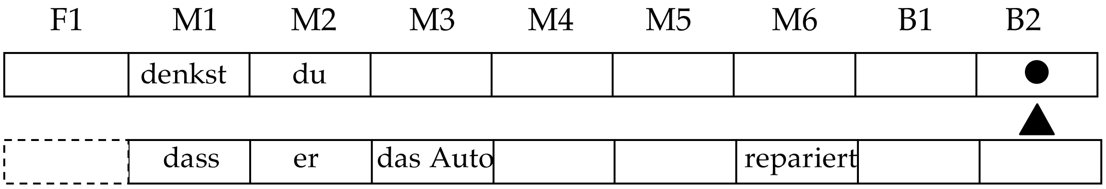


Figure 5. Linearization of example (1′).

Slot F1 of the main clause remains empty because the 'interrogative clause' arc leaving state F1 does not require a Wh-phrase as an obligatory filler. In state M1*a*, the linearizer deposits the mandatory HeaD verb into slot M1. Finally, the root S-node of the finite CoMPlement clause lands in B2, as indicated by the black dot and triangle. The linearizer of the finite CoMPlement starts in state F1 but immediately skips to M1*b* without leaving anything behind in slot F1 of the embedded topology. The CoMPlementizeR *dass* is dropped in slot M1. The SUBJect and the Direct OBJect are allowed to land in M2 and M3, respectively (see Section 6 for details), and the finite verb goes to M6.

If a sentence contains more than one verb, each of the verb frames concerned instantiates its own topology and linearizer FSA. This applies to verbs of any type, whether main, auxiliary or copula. In such cases, the topologies are allowed to *share* certain identically labeled slots, conditionally upon several restrictions to be explained shortly. *After two slots have been shared, they are no longer distinguishable; in fact, they are the same object.* In example (1′), the embedded topology shares its F1 slot with the F1 slot of the matrix. This is indicated by the dashed border of the bottom F1 slot.

Now consider example (2), where the Direct OBJect of *repariert* is a Wh-phrase (*welches Auto* 'which car'). The linearizer FSA drops this constituent in the shared F1 slot: see Figure 6.

(2) *Welches Auto denkst du, dass er repariert?*
'Which car do you think that he repairs?'

As a consequence, the Wh-phrase is already 'seen' by the linearizer of the matrix topology. The Wh-constituent gets 'fronted' and seems to have been 'extracted' from the complement. *Topology sharing* manifests itself as upward movement of constituents in shared slots. We will call this effect *promotion*. (Harbusch & Kempen (2000) describe a polynomial method for handling promotion.)

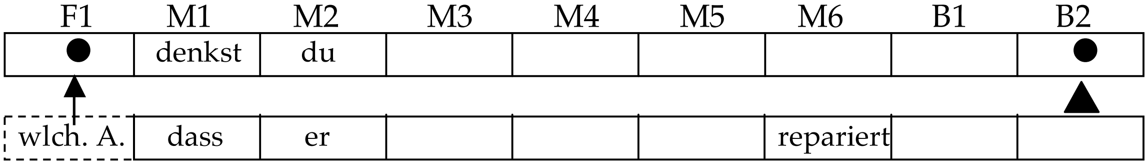


Figure 6. Linearization of example (2).

Example (1) embodies a more radical case of topology sharing. The topologies associated with *hat* and *repariert* share the entire region extending from F1 through M4. The result is promotion of Direct OBJect *das Auto*, as shown in Figure 7. We refer to Kempen & Harbusch (2001) for a detailed description of the German word order phenomena that can be explained by 'left-peripheral topology sharing'.

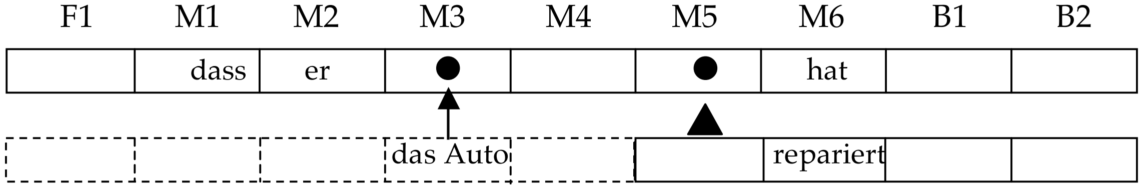


Figure 7. Linearization of example (1).

Now that the essentials of PG's linear ordering component are in place, we can return to the word order phenomena sketched in the Introduction.

## 5. Semi-free word order in the Midfield of German clauses

Several experimental investigations into the acceptability of word order variation in the Midfield of subordinate clauses of German have recently been published in the psycholinguistic literature (Pechmann, Uszkoreit, Engelkamp & Zerbst (1994, 1996); Rösler, Pechmann, Streb, Röder & Hennighausen (2000) and Keller (2000a, b)). The data patterns emerging from these experiments are very similar. While none of the six possible permutations of Subject (*S*), Indirect Object (*I*) and Direct Object (*O*) are definitely ruled out, some are judged considerably more acceptable than others. Furthermore, the acceptability varies in function of whether the constituents are full NPs or pronominal NPs.

Theoretical accounts for the obtained data patterns usually employ a ranked or weighted set of Linear Precedence (LP) constraints (see Pechmann *et al.* (1994, 1996), Müller (1999) and Keller (2000a, b)). A typical example is given in (3). The symbol "<<" denotes linear precedence, and the constraints are listed in order of decreasing rank/weight.

(3) (A) *[+NOM] << [–NOM]* (Subject NP precedes other NPs)
(B) *[+PRO] << [–PRO]* (pronominal NPs precede full NPs)
(C) *[+DAT] << [+ACC]* (Indirect Object precedes Direct Object)

Keller's experiments yield numerical weight values for the constraints. These confirm, in line with the Optimality Theoretical framework[1] he adopts, that

- violation of a higher ranked / weighted constraint reduces the acceptability ratings more seriously than violation of lower ones, and
- multiple violations affect the ratings additively.

Although models based on ranked / weighted LP constraints are descriptively adequate, they are not necessarily psychologically plausible. There is no evidence that the linearization system actually applies constraints like those in (3) at all. Actually, the phenomenon of incremental grammatical encoding suggests an alternative without explicit syntactic ordering rules for constituents whose position is flexible.

Consider the slots M2, M3 and M4 in the German linearizer FSA (Figure 4). Each of these accepts exactly one constituent of type Subject, Indirect Object, or Direct Object (henceforth *S*, *I* and *O*). Their actual order depends on their order of arrival, (their position in the queue), as explained in the previous Section. Now let us assume that — due to the combined effect of conceptual, lexical, and syntactic processing factors — Subject NPs are likely to precede Indirect Objects in the queue and, likewise, that *I* usually precedes *O*. That is:

$p(S<<I) > .5)$ and $p(I<<O) > .5$

implying that the surface order *SIO* will occur much more frequently than any other order. This suggests we can predict the actual probability of occurrence of the six possible orderings of *S*, *I* and *O*, if we know the probability of each of these constituent types preceding each other in the queue. Conversely, we can estimate these probabilities on the basis of the frequency distribution of the six *S+I+O* orderings in a corpus of subordinate clauses.

As far as we know, no such corpus is presently available. However, we can make the assumption that the frequency distribution of *S+I+O* permutations has a high positive correlation with acceptability ratings: frequently occurring sequences are likely to be judged as more acceptable than rarely occurring sequences. We therefore decided to utilize the ratings published recently by Keller (2000a, b).

In his Experiments 6 and 10, Keller applied the psychophysical method of magnitude estimation to elicit particularly fine-graded grammaticality judgments (see Bard,

[1] For a recent overview of Optimality Theory, see Dekkers, Van der Leeuw & Van Weijer (2000).

Robertson & Sorace (1996) for details)[2]. The sentence material Keller prepared for his Experiment 6 are illustrated in (4). All NPs refer to human protagonists and are introduced by case-marked articles. The sentences were presented without context.

(4) *Ich glaube, dass der Produzent dem Direktor den Schauspieler vorschlägt*
I believe that the-NOM producer the-DAT director the-ACC actor proposes
'I believe that the producer proposes the director the actor'

The material for Experiment 6 also included sentences with two full and one pronominal NP in nominative, dative or accusative case (*er*, *ihm*, *ihn*). So, every sentence like (4) was presented in 24 different shapes: 6 (permutations of *S*, *I* and *O*) times 4 (NP versions: 1 full, 3 pronominal). The transformed[3] acceptability ratings for all 24 data points are presented in Table 1.

Table 1. Relative frequency estimates of *S+I+O* permutations based on Keller's (2000a) acceptability ratings. See footnote 3 for the applied linear data transformation method. Rows refer to NP types (full or pronominal), columns represent permutations of the constituents. For a graphical rendering of these data, see the dark bars in Figure 8.

| | $S_f,I_f,O_f$ | $S_p,I_f,O_f$ | $S_f,I_p,O_f$ | $S_f,I_f,O_p$ |
|---|---|---|---|---|
| **SIO** | .2842 | .2782 | .2640 | .1370 |
| **SOI** | .2405 | .2840 | .1829 | .3043 |
| **ISO** | .1719 | .1541 | .2506 | .1011 |
| **IOS** | .0936 | .0899 | .0974 | .0636 |
| **OSI** | .1189 | .1272 | .0893 | .2734 |
| **OIS** | .0909 | .0666 | .1157 | .1206 |

Keller's Experiment 10 yielded acceptability ratings for sentences with only two constituents — *S* and *O*, full or pronominal. The Subject NPs were animate, the Direct Objects inanimate (see (5) for an example). Table 2 presents the average acceptability ratings after application of the same linear transformation as on the data of

[2] Following Keller (*o.c.*) and Bard *et al.* (*o.c.*), we use the terms 'acceptability' and 'grammaticality' interchangeably, considering both as referring to a graded rather than a discrete (binary) notion.

[3] The acceptability values reported by Keller range from about –.36 to about +.21. In order to transform them to (quasi-)frequencies, we added .5 to each rating, thus projecting them into the interval [0:1] in a linear manner. Then we summed up the (quasi-)frequencies of the six possible *S+I+O* permutations in a given sentence type (i.e., for each column in Table I, we added the six values), and divided each (quasi-)frequency by the total of its column. It follows that the six values in each column add up to 1. The resulting proportions (i.e., relative frequency estimates) are shown in Table 1. We applied this linear transformation in order to make Keller's original acceptability ratings comparable to the outcome of the probabilistic model simulations below.

Experiment 6 (see footnote 3). That is, first the possibly negative values (ranging from –.11 to +.42) were projected into the interval [0:1] by adding .5. Second, we summed up the (quasi-)frequencies of the two possible *S+O* permutations (i.e., the two values in each column of the table), and computed their relative (quasi-)frequencies.

(5) *Maria glaubt, dass der Vater den Wagen kauft*
Maria believes that the-NOM father the-ACC car buys
'Maria believes that the father buys the car'

Table 2. Relative frequency estimates of *S+O* permutations based on Keller's (2000a) Experiment 10. See footnote 3 for the applied linear data transformation method.

| | $S_f,O_f$ | $S_f,O_p$ | $S_p,O_f$ | $S_p,O_p$ |
|---|---|---|---|---|
| **SO** | .5920 | .5023 | .6906 | .6713 |
| **OS** | .4080 | .4977 | .3094 | .3287 |

## 6. A probabilistic model of the acceptability judgments

Under the assumption that Keller's acceptability judgments are indeed interpretable as reflections of the relative frequency of the various ordering patterns, we now construct a probabilistic model capable of filling the slots M2–M4 (cf. Figure 4) with frequencies that approximate the data in Tables 1 and 2. In line with the idea of incremental grammatical encoding, we assume that the constituents land in slots M2 through M4 of the topology in their order of arrival, and interpret the quantities in the cells of Tables 1 and 2 as estimates of the probability of occurrence of constituent strings in a corpus. Furthermore, we hypothesize that these 'string probabilities' are predictable on the basis of the 'precedence probability' of *pairs* of constituents. For every pair of constituents A and B belonging to the same clause, we need to estimate the probability that A arrives earlier than (i.e. becomes available for being ordered before) B. Given that a clause contains at most one *S*, *I* or *O* constituent, and that each of these may be full or pronominal, we have to estimate precedence probabilities for 12 pairs of constituents. Actually, because the sentence materials used in Keller's experiment contained at most one pronominal NP, we have no target values for precedence pairs listing two pronominal NPs : $p(S_p<<I_p)$, $p(S_p<<O_p)$ and $p(I_p<<O_p)$. This reduces the number of free parameters to 9.

The probability of a three constituents string ABC, *p("ABC")*, can be predicted from the the precedence probabilities of the pairs A<<B, B<<C, and A<<C, via the

equation[4]

$$p(\text{“}ABC\text{”}) = \frac{p(A<<B) * p(B<<C) * p(A<<C)}{1 - p(A<<B) * p(B<<C) + p(A<<C) * (p(A<<B) + p(B<<C) - 1)}$$

For a derivation of this string probability formula we refer to the Appendix.

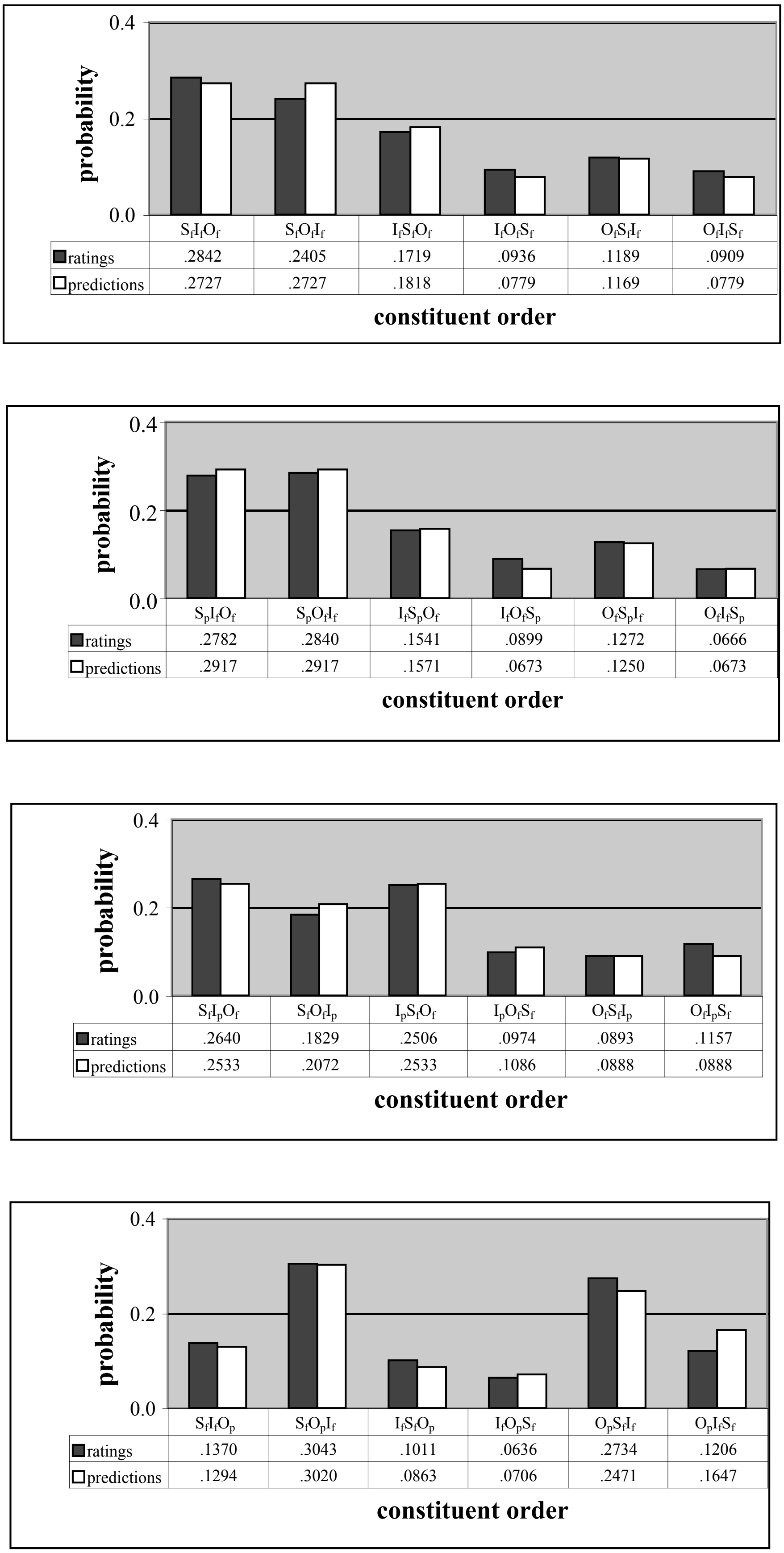
Which combination of values of the nine precedence probabilities yields the closest fit with the 24 string probabilities in Table 1? And is this fit close enough to justify the claim that the model provides a satisfactory account of the observations? To answer these questions we implemented a computer simulation program that searched virtually the entire parameter space defined by the 9 precedence probabilities. Each of these variables was assigned values between .05 and .95 (incrementing by .05). For every setting, the program computed the probability of each of the 24 strings referred to in Table 1. Using a Least Squares method, we determined the parameter setting giving the best prediction of the string probabilities in Table 1. The best solution is presented in Table 3.

Table 3: Optimal values of the precedence probabilities estimated by the simulation model.

| **Order** | Optimal value | **Order** | Optimal value | **Order** | Optimal value |
|---|---|---|---|---|---|
| $\mathbf{S_f<<I_f}$ | .60 | $\mathbf{S_p<<I_f}$ | .65 | $\mathbf{S_f<<I_p}$ | .50 |
| $\mathbf{S_f<<O_f}$ | .70 | $\mathbf{S_p<<O_f}$ | .70 | $\mathbf{S_f<<O_p}$ | .55 |
| $\mathbf{I_f<<O_f}$ | .50 | $\mathbf{I_p<<O_f}$ | .55 | $\mathbf{I_f<<O_p}$ | .30 |

Figure 8 displays the ‘predicted’ string probabilities determined by inserting the parameter settings in Table 3 into the string probability formula. The differences between the target values (Table 1) and the ‘predicted’ values is indeed very small. The correlation between obtained and predicted string probabilities is .98. We conclude that the incrementality-driven approach we took in this chapter is viable. Inspection of the optimal parameter settings in Table 3 reveals that they are in line with constraints proposed earlier, such as those in (3). The Subject tends to precede both Indirect and Direct Object with an average precedence probability of .62 (first and strongest constraint in (3)). Pronominal NPs tend to arrive earlier than full NPs (average probability .59; second constraint in (3)). There is no clear preference for the Indirect Object to outwin the Direct Object (average precedence probability .48; third and weakest constraint in (3)). Pronominal Direct Objects even strongly prefer to pre-

[4] In order to obtain the equation for another permuation one only replaces the numerator; e.g., for CAB: *p(C<<A)*p(C<<B)*p(A<<B)*. The denominator remains the same.



| | $S_fI_fO_f$ | $S_fO_fI_f$ | $I_fS_fO_f$ | $I_fO_fS_f$ | $O_fS_fI_f$ | $O_fI_fS_f$ |
|---|---|---|---|---|---|---|
| ■ratings | .2842 | .2405 | .1719 | .0936 | .1189 | .0909 |
| □predictions | .2727 | .2727 | .1818 | .0779 | .1169 | .0779 |

| | $S_pI_fO_f$ | $S_pO_fI_f$ | $I_fS_pO_f$ | $I_fO_fS_p$ | $O_fS_pI_f$ | $O_fI_fS_p$ |
|---|---|---|---|---|---|---|
| ■ratings | .2782 | .2840 | .1541 | .0899 | .1272 | .0666 |
| □predictions | .2917 | .2917 | .1571 | .0673 | .1250 | .0673 |

| | $S_fI_pO_f$ | $S_fO_fI_p$ | $I_pS_fO_f$ | $I_pO_fS_f$ | $O_fS_fI_p$ | $O_fI_pS_f$ |
|---|---|---|---|---|---|---|
| ■ratings | .2640 | .1829 | .2506 | .0974 | .0893 | .1157 |
| □predictions | .2533 | .2072 | .2533 | .1086 | .0888 | .0888 |

| | $S_fI_fO_p$ | $S_fO_pI_f$ | $I_fS_fO_p$ | $I_fO_pS_f$ | $O_pS_fI_f$ | $O_pI_fS_f$ |
|---|---|---|---|---|---|---|
| ■ratings | .1370 | .3043 | .1011 | .0636 | .2734 | .1206 |
| □predictions | .1294 | .3020 | .0863 | .0706 | .2471 | .1647 |

Figure 8. Comparison of observed and predicted acceptability ratings for three-constituent sentences. First panel: *S*, *I* and *O* full NPs; second: *S* pronominal; third: *I* pronominal; fourth: *O* pronominal.

cede full Indirect Objects: $p(O_p{<}{<}I_f) = .70$ (in agreement with well-know observations; *Ich habe Maria es gegeben* is considerably worse than *Ich habe es Maria gegeben*).

The probability estimates for the *S+O* precedence pairs can be compared directly with the observed data in Table 2. Table 4 shows that the predictions (or rather 'postdictions'), although less accurate, are satisfactory (correlation .93). Remember that this comparison involves two experiments with different sentence materials (cf. examples (4) and (5)).

Table 4. Comparison of rated and predicted values for sentences with Subject and a Direct Object NPs (Keller's (2000a) Experiment 10). The $S_p,O_p$ values cannot be estimated from the simulations of Experiment 6, where the sentences contained at most one pronominal NP.

| | | **$S_f,O_f$** | **$S_p,O_f$** | **$S_f,O_p$** | **$S_p,O_p$** |
|---|---|---|---|---|---|
| **SO** | ratings | .59 | .69 | .50 | .67 |
| | predictions | .70 | .70 | .55 | n.a. |
| **OS** | ratings | .41 | .31 | .50 | .33 |
| | predictions | .30 | .30 | .45 | n.a. |

## 7. Summary and conclusion

We have presented a linearization formalism capable of capturing a broad range of clausal constituent order phenomena in semi-free word order languages such as German. It is part of the psycholinguistically motivated formalism of Performance Grammar (PG), which has separate components for assembling the hierarchical and the linear structure of sentences. The basic data structure used by the linearization component is called *topology*: 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. A finite-state automaton called *linearizer* actually assigns constituents to slots while traversing topologies from left to right. Focusing on the phenomenon of 'scrambling' of Subject (*S*), Direct Object (*O*) and Indirect Object (*I*) in the 'Midfield' of German clauses, we assumed a clausal topology with three special slots, each accommodating at most one of these constituents in any order (see Figure 4). Furthermore, we hypothesized that, due to various processing factors at the level of the conceptualizer and/or the semantic-syntactic interface, these constituents tend to 'arrive' at different points in time. That is, they get ready to be assigned a slot by the linearizer sequentially rather than in parallel (incremental grammatical encoding). If the overt order of *S*, *I* and *O* indeed mirrors their order of arrival, then it should be possible to predict the probability of the various permutations on the basis of the precedence probabilities of pairs of constituents, e.g. on how likely it is for *S* to arrive before *I*, etc. Computer simulation of a model based on these assumptions produced a very good fit with experimental data that we took to reflect probabilities of occurrence of *S*, *I* and *O* permutations.

The success of this approach suggests, first of all, that satisfactory accounts of the scrambling phenomenon under scrutiny may well be found within the conceptualizer or the semantic-syntactic interface rather than in the domain of explicit syntactic ordering rules. If so, the syntax of German needs no provisions at all for dealing with Midfield scrambling. Secondly, our incrementality-based approach appears to offer a viable alternative to published accounts in terms of a hierarchy of ordering constraints.

In conclusion, the present study argues that certain — not necessarily all — scrambling phenomena in (semi-)free word order languages can profitably be treated as a consequences of factors operative in the semantic-syntactic interface rather than as the result of applying explicit word order rules.

## Appendix. Estimating string probabilities from precedence pair probabilities

The constituent sequences considered in this paper (SIO, ISO, etc.) can be represented by strings of at most three symbols — call them A, B, and C. Their probability of occurrence, we assume, is a function of the precedence probabilities *p(A<<B)*, *p(B<<C)* and *p(A<<C)*. Imagine three (biased or unbiased) coins with a precedence relation inscribed on one side and its inverse on the other side. Tossing these coins will reveal the probabilities of the precedence pairs, e.g. *p(A<<B)* and *p(B<<A) = 1 - p(A<<B)*. When three coins are tossed, there are eight possible outcomes; see the three leftmost columns in this Table:

| **Precedence alternatives** | | | **Strings** |
|---|---|---|---|
| A<<B | B<<C | A<<C | ABC |
| A<<B | B<<C | C<<A | — |
| A<<B | C<<B | A<<C | ACB |
| A<<B | C<<B | C<<A | CAB |
| B<<A | B<<C | A<<C | BAC |
| B<<A | B<<C | C<<A | BCA |
| B<<A | C<<B | A<<C | — |
| B<<A | C<<B | C<<A | CBA |

Six of these outcomes define unique strings — enumerated in the rightmost column.

Two outcomes (the second and the seventh) are 'illegal' because they violate transitivity of the precedence relation (if A<<B and B<<C then A<<C).

If we now let $x = p(A<<B)$, $y = p(B<<C)$ and $z = p(A<<C)$, then the probability of getting the first tossing result, which yields string "ABC", is expressed by the product $xyz$. The sum of the probabilities of the six possible legal (i.e. transitive) outcomes therefore can now be written as

$$xyz+x(1-y)z+x(1-y)(1-z)+(1-x)yz+(1-x)y(1-z)+(1-x)(1-y)(1-z)$$

or, equivalently, as

$$1-xy(1-z)-(1-x)(1-y)z = 1-xy+z(x+y-1).$$

It follows that:

$$\textit{the probability of a specific string} = \frac{\textit{the probability of the corresponding tossing result}}{\textit{the probability of any legal tossing result}}$$

$$= \frac{\textit{the probability of the corresponding tossing result}}{1-xy + z(x+y-1)}$$

For instance,

$$p(\text{“}ABC\text{”}) = \frac{xyz}{1-xy + z(x+y-1)}$$

$$p(\text{“}CBA\text{”}) = \frac{(1-x)(1-y)(1-z)}{1-xy + z(x+y-1)}$$

Substituting the variables $x$, $y$ and $z$ by the original precedence probabilities gives, for example

$$p(\text{“}ABC\text{”}) = \frac{p(A<<B) * p(B<<C) * p(A<<C)}{1 - p(A<<B) * p(B<<C) + p(A<<C) * (p(A<<B) + p(B<<C) - 1)}$$

There are six such formulae, one for each permutation of A, B and C. In order to obtain estimators for the 24 string probabilities in Table 1, one replaces the symbols A, B and C by the real constituents: $\{S_f, I_f, O_f\}$, $\{S_p, I_f, O_f\}$, $\{S_f, I_p, O_f\}$ and $\{S_f, I_f, O_p\}$.

## References

BARD, E. G., ROBERTSON, D. & SORACE, A. (1996). Magnitude estimation of linguistic acceptability. *Language, 72,* 32-68.

BOCK, J.K. & LEVELT, W.J.M. (1994). Language production: grammatical encoding. In: Gernsbacher, M.A. (Ed.), *Handbook of psycholinguistics*. San Diego: Academic Press.

DEKKERS, J., VAN DER LEEUW, F. & VAN WEIJER, J. (Eds.) (2000). *Optimality Theory: Phonology, Syntax, Acquisition*. Cambridge: Oxford University Press.

DELL, G.S., BURGER, L. K. & SVEC, W.R. (1997). Language production and serial order: A functional analysis and a model. *Psychological Review, 104,* 123-147.

GARRETT, M.F. (1975). The analysis of sentence production. In: Bower, G.H. (Ed.), *The psychology of learning and motivation* (Vol. 9). New York: Academic Press.

HARBUSCH, K. & KEMPEN, G. (2000). Complexity of linear order computation in Performance Grammar, TAG and HPSG. In: *Proceedings of the Fifth International Workshop on Tree Adjoining Grammars and related frameworks (TAG+5)*, Paris, May 2000.

KATHOL, A. (2000). *Linear Syntax.* Oxford: Oxford University Press.

KELLER, F. (2000a). *Gradience in grammar: Experimental and computational aspects of degrees of grammaticality.* Ph.D. Thesis, University of Edinburgh.

KELLER, F. (2000b). Evaluating competition-based models of word order. In*: Proceedings of the 22nd Annual Conference of the Cognitive Science Society*. Mahwah NJ: Erlbaum.

KEMPEN, G. & HARBUSCH, K. (1998). A 'tree adjoining' grammar without adjoining: The case of scrambling in German. In: *Proceedings of the Fourth International Workshop on Tree Adjoining Grammars and related frameworks (TAG+4),* Philadelphia, August 1998. University of Pennsylvania, IRCS Report 98-12.

KEMPEN, G. & HARBUSCH, K. (2001). *A uniform topological model of clause linearization in Dutch, English and German.* Manuscript submitted for publication.

KEMPEN, G. & HOENKAMP, E. (1987). An incremental procedural grammar for sentence formulation. *Cognitive Science, 11,* 201-258.

LEVELT, W.J.M. (1989). *Speaking: From intention to articulation.* Cambridge MA: MIT Press.

MÜLLER, G. (1999). Optimality, markedness, and word order in German. *Linguistics, 37,* 777-815.

PECHMANN, T., USZKOREIT, H., ENGELKAMP, J. & ZERBST, D. (1994). *Word order in the German Middle fiel*d. Report Nr. 43, Department of Computational Linguistics, University of the Saarland.

PECHMANN, T., USZKOREIT, H., ENGELKAMP, J. & ZERBST, D. (1996). Wortstellung im deutschen Mittelfeld. Linguistische Theorie und psycholinguistische Evidenz. In: Habel, C., Kanngießer, S. & Rickheit, G. (Eds.), *Perspektiven der Kognitiven Linguistik.* Wiesbaden: Westdeutscher Verlag.

RÖSLER, F., PECHMANN, T., STREB, J., RÖDER, B. & HENNIGHAUSEN, E. (1998). Parsing of sentences in a language with varying word order: Word-by-word variations of processing demands are revealed by event-related brain potentials. *Journal of Memory and Language, 38,* 150-176.

YAMASHITA, H. & CHANG, F. (2001). "Long before short" preference in the production of a head-final language. *Cognition, 81*, B45-B55.