Modeling of Lexical Access in Speech Production: A Psycholinguistic Perspective on the Lexicon

Ardi Roelofs
Max Planck Institute for Psycholinguistics

1 MODELING A MENTAL SKILL

A basic skill of speakers is the access of words in memory. In producing utterances, speakers call on many facets of their stored knowledge about words, including the words' meaning, syntactic properties, morphological composition, and sound structure. Lexical access is the process by which this information about words is retrieved from memory in order to construct articulatory programs for the concepts to be verbally expressed. Lexical access is a very efficient process. It poses no problem to a speaker to retrieve three to five words per second from an active lexicon of some thirty-thousand words. Only a few errors are made, namely one per thousand words, approximately (Levelt, 1989). How is the efficient access of words in memory achieved?

Whereas linguistic models are designed to account for facts about language competence or knowledge per se (e.g., what facets of words speakers know), psycholinguistic models are designed to explain facts about language performance such as how this knowledge is stored in memory and how it is accessed (e.g., findings about the time to retrieve specific types of lexical information from memory and to compute articulatory programs). Many psycholinguists believe that the complexity of the processes and representations involved in lexical access requires that a model is complemented by computer simulations.

Over the past few years, I have developed a computational model of the skill of lexical access in speech production (Roelofs, 1992, 1993, 1996c, 1997c). This model is called WEAVER++. WEAVER (Word-form Encoding by Activation and VERification) was originally the name of a model for the memory retrieval of morphological and phonological properties and the construction of articulatory programs (Roelofs, 1997c). This model for form encoding has recently been combined with the model of lexical selection developed in Roelofs (1992, 1993), and together the two models are now called WEAVER++ (Roelofs, 1998). The WEAVER++ model has been constructed within the theoretical framework for lexical access developed by Levelt and colleagues (Levelt, 1989, 1992; Levelt, Roelofs, & Meyer, 1999; Levelt & Wheeldon, 1994). In this chapter, I review

WEAVER++ and I discuss two of its applications. These applications concern classical findings from the literature about access times and findings from new experiments by my own specifically designed to test the model. WEAVER++ is certainly not the only computer-implemented psycholinguistic model of lexical access in the literature. For example, Dell (1986, 1988) developed a computational model designed to account for facts about slips of the tongue (WEAVER++ is compatible with the error data but has not specifically been designed to account for errors). For an overview of these other models, I refer to Levelt (1989) and Roelofs (1996a). The aim of this chapter, however, is to illustrate some main characteristics of the psycholinguistic perspective on the lexicon by describing a particular psycholinguistic model and discussing some core issues, research methods, and findings about access times rather than to provide a comprehensive review of the state of the art. I start by briefly indicating the role of lexical access in the process of speech production.

2 Stages in Planning the Production of Speech

As with other skilled behavior such as playing the piano, the production of speech requires advance planning of action components. Psycholinguists working on language production assume that three major types of processes underlie speaking: conceptualization, formulation, and articulation (e.g., Kempen & Hoenkamp, 1987; Levelt, 1989). Figure 1 illustrates the major planning stages. Conceptualization processes plan messages, that is, conceptual structures to be verbally expressed. WEAVER++ assumes that messages are specified in terms of lexical concepts and their relationships. Messages may, for example, be derived from external scenes via object perception (e.g., perceiving a guitar). Formulation processes take the message, access appropriate words for the lexical concepts, and plan a syntactic structure in case of sentence production and a morpho-phonological structure for the utterance. The result is an articulatory program, also called a "phonetic plan". WEAVER++ assumes that phonetic plans consist of motor programs for the syllables in the utterance (cf. Levelt, 1989). Finally, articulation processes execute the articulatory program, which results in overt speech.

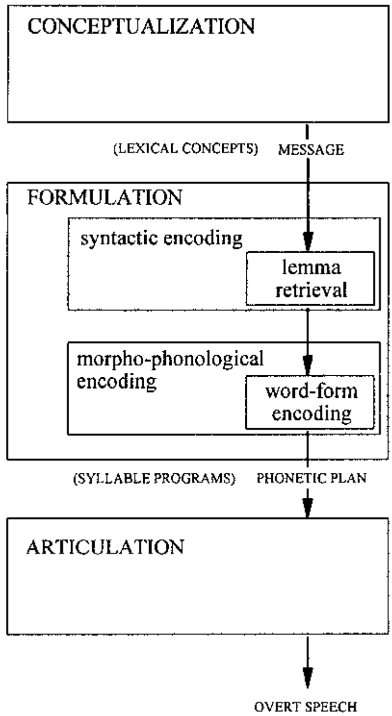
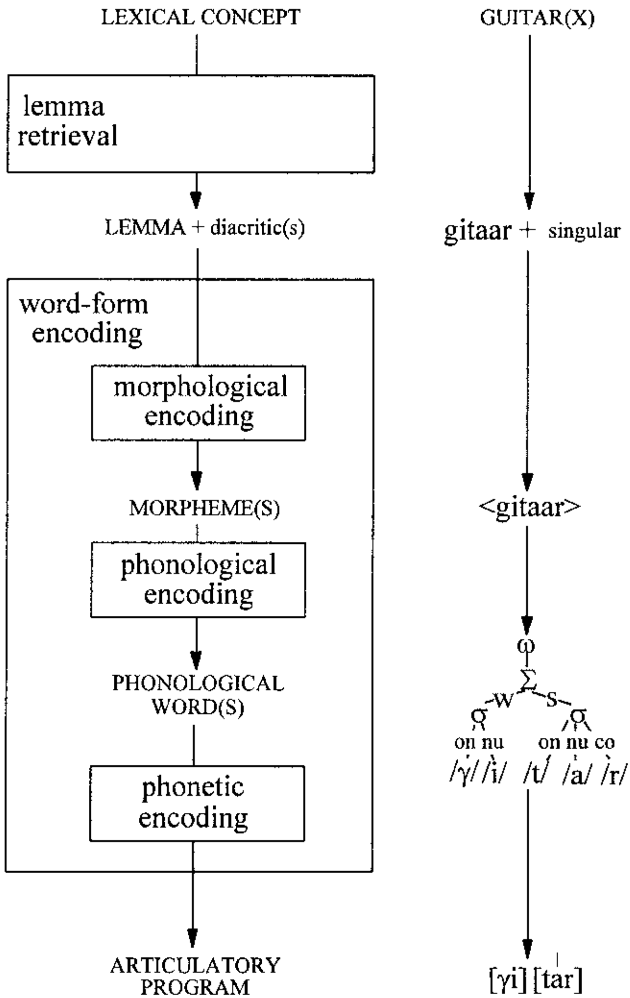


Figure 1: Stages underlying language production assumed by WEAVER++.

WEAVER++ assumes that accessing words in memory is accomplished in two major steps, corresponding to the two formulation stages (i.e., syntactic encoding and morpho-phonological encoding). The first access step is called lemma retrieval, and the second step is referred to as word-form encoding. In lemma retrieval, a message concept is used to retrieve the "lemma" of a corresponding word from memory. A lemma, as it is defined here (contrary to linguistic usage), is a memory representation of the syntactic properties of a word, crucial for its use in sentences. For example, the lemma of the Dutch word *gitaar* ('guitar') indicates that it is a noun and that it takes non-neuter grammatical gender. In word-form encoding, the morpho-phonological properties of the word are retrieved from memory in order to construct an articulatory program for the word, which will guide articulation. For example, the morpheme <gitaar> and the segments /γ/, /i/, /t/, /a/, and /r/ are retrieved and a phonetic plan for [γi.'tar] is generated.

Assume a Dutch speaker sees a guitar and wants to name it. Lexical access consists of mapping the lexical concept GUITAR(X) onto the articulatory program for *gitaar*. The stages of access in WEAVER++ are illustrated in Figure 2.



ω = phonological word, Σ = foot, s = strong, w = weak, σ = syllable, on = onset, nu = nucleus, co = coda.

Figure 2: Stages of lexical access in WEAVER++.

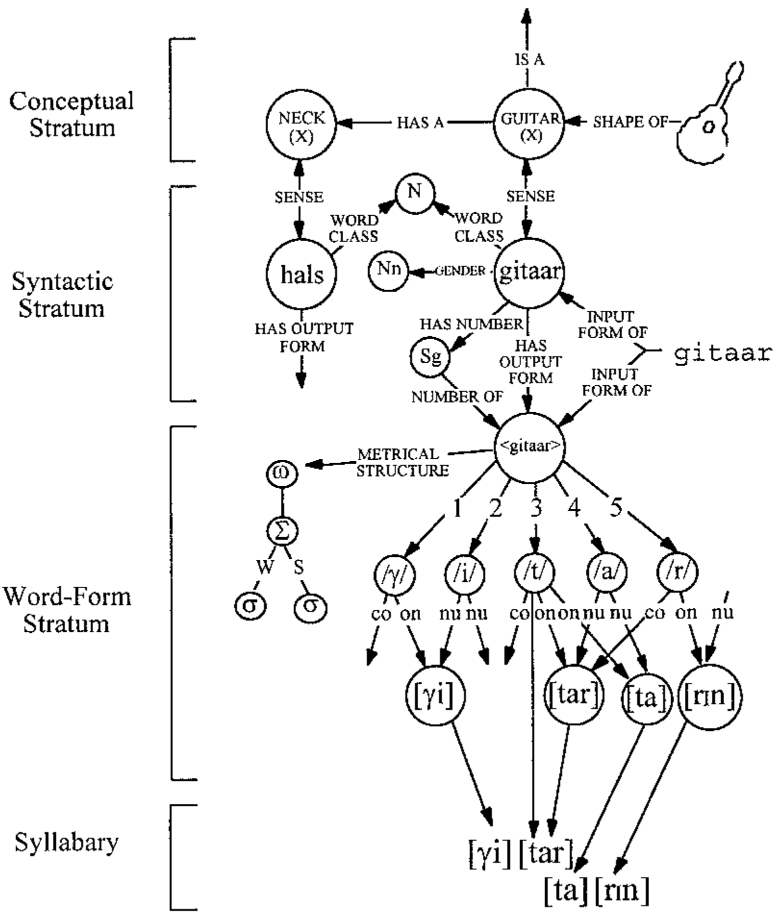
First, the lemma retriever takes the concept GUITAR(X) and recovers the lemma of *gitaar*. That is, it makes available the syntactic class "noun" and gender "non-neuter", and so forth. In order to derive the appropriate word form, singular *gitaar* instead of plural *gitaren*, the word's number has to be specified. The lemma plus the number diacritic are then input to word-form encoding.

The articulatory program is derived in three major steps: morphological encoding, phonological encoding, and phonetic encoding (Dell, 1986; Levelt, 1989). The morphological encoder takes the lemma of *gitaar* plus the diacritic "singular" and recovers the stem morpheme <gitaar>. The phonological encoder takes this morpheme and produces a phonological word representation. This representation describes the singular form of *gitaar* as a phonological word (ω) consisting of an iambic foot (Σ). The first syllable (σ) has /γ/ as onset and /i/ as nucleus, and the second syllable has /t/ as onset, /a/ as nucleus, and /r/ as coda. Finally, the phonetic encoder takes this phonological word representation and derives the corresponding articulatory program. It is assumed that articulatory programs are not constructed from scratch. Instead, when available, learned motor programs for the syllables in the utterance are accessed in a phonetic syllabary (Levelt, 1989, 1992; Levelt & Wheeldon, 1994). This is a store of ready-made motor programs for syllables.

3 A Model for Lexical Access: WEAVER++

The central questions for a psycholinguistic model of lexical access are: How is lexical memory organized and what is the access algorithm? In order to provide for an efficient spelling out of lexical information, WEAVER++ assumes that the mental lexicon is a network of nodes and links that is accessed by spreading of activation (cf. Dell, 1986).

Figure 3 illustrates the structure of entries in the lexical network, in this example, the memory representation of the Dutch word *gitaar* ('guitar') plus part of the representation of *hals* ('neck'). A lexical network with nodes and labeled links is connected to a syllabary with learned motor programs for syllables. The lexical network consists of three major strata: a conceptual stratum, a syntactic stratum, and a word-form stratum. The conceptual stratum contains concept nodes and labeled conceptual links. Each lexical concept in the language, for example GUITAR, is represented by an independent node. The links specify conceptual relationships, for example, between a concept and its properties (HAS A) and superordinates (IS A). The syntactic stratum contains lemma nodes (gitaar), syntactic property nodes and labeled links (gender: non-neuter, and word class: noun), and slots for the specification of diacritics (number: singular).



N = noun, Nn = non-neuter, Sg = singular, ω = phonological word, Σ = foot, s = strong, w = weak, σ = syllable, on = onset, nu = nucleus, co = coda.

Figure 3: A fragment of the lexical network of WEAVER++. Illustrated is the lexical entry of the Dutch word *gitaar* and part of the entry of *hals*.

The word-form stratum contains nodes for metrical structures, morphemes, phonemic segments, and syllable programs. Morpheme nodes (e.g., <gitaar>) are connected to a lemma and its diacritics. The links between morphemes and segments specify the serial position (1st, 2nd, 3rd, etc.) of the segments within the morpheme. The links between segments (e.g., /r/) and syllable program nodes (e.g., the node [rIn] in the network) specify possible syllabifications (onset, nucleus, coda) of the segments. For example, /r/ is the coda of the syllable *tar* but the onset of *rIn*. Each syllable program node (e.g., the node [rIn]) is linked to the actual motor program for the syllable in the syllabary (i.e., the program [rIn]).

Below, I explain the lexical access algorithm. First, I review some of the basic assumptions about lemma retrieval and one application, and next I discuss some of the basic assumptions about word-form encoding and one application.

3.1 Assumptions About the Retrieval of Lemmas

A basic claim of WEAVER++ is that lemmas are retrieved in a conceptually non-decomposed way. For example, the lemma of *gitaar* is retrieved on the basis of the abstract representation or "chunk" GUITAR(X) instead of features such as HAS-A-NECK(X) and HAS-STRINGS(X), which remain in memory as background information. Each lexical concept in the language is represented by an independent node in the network. There are good theoretical and empirical reasons for making this non-decomposition assumption in a model of production, which can be found in Roelofs (1997a). For example, non-decomposition prevents that in the retrieval of a word (e.g., *gitaar*) also all its hyperonyms (e.g., *instrument*) are retrieved (cf. Levelt, 1989). Furthermore, non-decomposition guarantees that messages can be verbalized. In the case of decomposition, verbalization is not guaranteed (Bierwisch & Schreuder, 1992).

Lemma retrieval in WEAVER++ starts by enhancing the level of activation of the node of the target lexical concept. Activation then spreads through the network, whereby each node sends a proportion of its activation to the nodes that are connected to it. The most highly activated lemma node is selected. For example, the activation level of the lexical concept node GUITAR(X) is enhanced. Activation spreads through the network. The lemma nodes of *gitaar* and *hals* will be activated. The lemma node for *gitaar* will be the most highly activated node, because it receives a full proportion of the activation of GUITAR(X), whereas the node for *hals* receives a proportion of a proportion. Thus, the lemma node of *gitaar* will be selected.

Details about the computer model such as the equations that formalize it, can be found in Roelofs (1992, 1993). There are equations for the spreading of activation and the instantaneous selection probability of a lemma node (i.e., the "hazard rate" of the retrieval process). The selection probability of a lemma node equals the ratio of its activation and that of all the other lemma nodes. Given the selection ratio, the mathematical expectation (i.e., theoretical mean) of the retrieval time can be computed.

3.2 Application of the Retrieval Algorithm

Chronometrical studies of mental processes have a long tradition in psychology. Many psycholinguists believe that analyzing the process of lexical access requires measurements of its time course. A psycholinguistic model for lemma retrieval should account for, among other things, how fast we retrieve lemmas for concepts to be expressed. Elsewhere, it has been shown that the retrieval algorithm of WEAVER++ explains classical findings about lemma retrieval times in picture naming, picture categorizing, and word categorizing. Furthermore, the model accounts for the time course of the inhibitory and facilitatory effects that external distractors (e.g., words) have on naming and categorizing times (see Roelofs, 1992, 1993). Naming and categorizing are among the simplest forms of conceptually driven lemma retrieval. Currently, picture-word interference is one of the major tasks in the chronometrical study of speech production.

In picture naming with distractors, speakers have to name pictured objects while simultaneously trying to ignore written words superimposed on the pictures, which are the distractors. For example, speakers have to say *chair* to a pictured chair and ignore the word *bed* (semantically related) or the word *fish* (semantically unrelated) superimposed on the picture. There is also an SOA (Stimulus Onset Asynchrony) manipulation. That is, the written distractors are presented 400, 300, 200, 100 milliseconds before picture onset (called negative SOAs), simultaneously with, or 100, 200, 300, 400 milliseconds after picture onset (called positive SOAs). The naming time (i.e., the interval between picture onset and speech onset) is the measurement of interest. The classical finding is that naming of pictures takes more time with related than with unrelated distractors at SOAs between -100 and +100 milliseconds: a semantic inhibition effect.

Figure 4 shows the SOA curves. I have plotted the semantic effect (i.e., the difference between the mean naming latencies with related and unrelated distractors) against SOA. A positive difference indicates a semantic inhibition effect. The semantic effects and the SOAs are in milliseconds. The real data are the classical measurements of Glaser and Düngelhoff (1984), which are indicated in the figure by the filled squares. Semantic inhibition is obtained at the SOAs of -100, 0, and 100 milliseconds. The predictions by WEAVER++ are indicated by the open squares. The model fits the data.

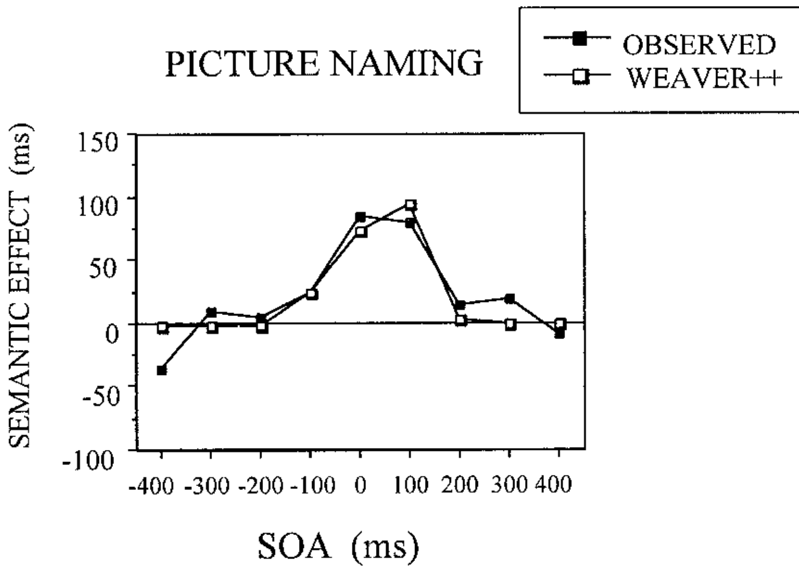


Figure 4: Semantic effect in picture naming as a function of stimulus onset asynchrony (SOA). Real data and WEAVER++ simulation results.

How does the model explain these classical findings? Assume *chair* is the target. In the case of a pictured chair and distractor *bed*, activation from the picture and the distractor word will converge on the lemma of the distractor *bed*, due to the connections between the lexical concept nodes for the words *chair* and *bed* in the conceptual stratum. In the case of the unrelated distractor *fish* there will be no such convergence of activation. As a consequence, *bed* will be a stronger competitor than *fish*, which yields semantic inhibition.

According to the model, distractors will only be competitors when they are permitted responses in an experiment. Speakers know which responses are permitted, because they are shown the target pictures and the names to be used before the beginning of an experiment. In the case of picture or word categorization, hyperonyms of the names of the pictures such as *furniture* and *animal* are the targets (e.g., the word *furniture* has to be produced to refer to a pictured chair). Now WEAVER++ predicts a semantic facilitation effect. For example, distractor *bed* will prime the target *furniture*, but *bed* will now not be a competitor itself because it is not a permitted response in a categorization experiment. By contrast, *fish* superimposed on a pictured chair will prime *animal*, which is a competitor for the target *furniture*. Thus, semantic facilitation is predicted, and this is exactly what Glaser and Düngelhoff (1984) empirically obtained. The same prediction holds for word categorizing (e.g., saying *furniture* to the word *bed* and ignoring the "distractor" picture). Again, the predicted semantic facilitation was empirically obtained.

The SOA curves discussed so far concerned classical findings from the literature. I also tested predictions of the model in new experiments (Roelofs, 1992, 1993). Distractors such as the hyperonym *furniture*, the co-hyponym *bed*, and the hyponym *throne* should facilitate the naming response *chair* if the pictured chair is the only piece of furniture to be named in the experiment. The distractors are not permitted responses (i.e., not names of other pictures), and therefore they will not be competitors for the target word. Thus, the distractors should yield facilitation if they are semantically related, as we saw in the categorization experiments. This prediction was confirmed by the empirical data (Roelofs, 1992). Furthermore, as predicted, there was no difference in the size of the facilitatory effect between hyperonym (*furniture*), co-hyponym (*bed*), and hyponym (*throne*) distractors. Predictions were also tested and validated for producing verbs in the naming of actions (Roelofs, 1993). For example, the production of the verb *laugh* is inhibited by *cry* (if it is the name of another pictured action) and facilitated by its hyponym *chuckle* (if it is not a permitted response).

Above, I have shown how the WEAVER++ model explains findings about the time course of lemma retrieval obtained with the picture-word interference paradigm. The remainder of this chapter is about the encoding of word forms in speech production, the process following lemma retrieval.

3.3 Assumptions About the Encoding of Word Forms

WEAVER++ assumes that in word-form encoding, lemmas are mapped onto learned syllable-based articulatory programs. This mapping is achieved by serially grouping the phonemic segments of morphemes into phonological syllables, which are then used to address the programs in a phonetic syllabary (cf. Levelt, 1992).

The memory representation of the form of the Dutch word *gitaar* was already illustrated in Figure 3. The word-form stratum contains a metrical part and a non-metrical part. The metrical part describes abstract groupings of syllables (σ) into feet (Σ) and feet into phonological words (ω). The non-metrical part of the network consists of three layers of nodes: morpheme nodes, segment nodes, and syllable program nodes. Morpheme nodes are connected to the lemma and its diacritics. For example, the stem <gitaar> is connected to the lemma of *gitaar* and "singular". A morpheme node points to two types of information, namely to its canonical metrical structure and to the segments that make up its underlying form. The links between morpheme and segment nodes indicate the serial position of the segments within the morpheme. Possible syllable positions (onset, nucleus, coda) of the segments are specified by the links between segment nodes and syllable program nodes. For example, the network specifies that /r/ can be the coda of [tar] and the onset of [rɪn]. These links are used to retrieve a motor program for a phonological syllable after the actual syllable positions of the segments of the utterance have been determined by a syllabification process.

The encoding starts when a morpheme node receives activation from a lemma and its diacritics. Activation then spreads through the network. In the production of connected speech, several word forms may be available at a particular moment in time. Thus, the encoding algorithm has to select the relevant nodes among all the activated ones in order to syllabify them. To accomplish this task, the form encoders follow simple selection rules. The rules are implemented in a parallel distributed manner. Attached to each node in the network, there is a procedure that verifies the label on the link between the node and a target node one level up.

The morphological encoder selects the morpheme nodes that are linked to a selected lemma and its diacritics. Thus, <gitaar> is selected for the lemma of *gitaar* with the diacritical value "singular". This first encoding stage thus concerns what linguists traditionally call the "syntax-morphology interface" (e.g., Spencer, 1991).

The phonological encoder selects the segments and the metrical structure that are linked to the selected morpheme nodes. Next, the segments are syllabified by associating them to the syllable nodes within the metrical frame. The association proceeds from the segment whose link is labeled first to the one labeled second, and so forth. In associating the segments to the metrical frame, syllable positions (onset, nucleus, coda) are assigned to the segments following the syllabification rules of the language. Thus, in the encoding of <gitaar>, the /t/ is made syllable onset, the /a/ nucleus, and the /r/ coda. In producing the plural form *gitaren* the metrical frames of the stem <gitaar> and the plural suffix <en> are combined. Similarly, in connected speech, a speaker may generate a cliticized form such as *gitarin*, which phonologically combines the words *gitaar* and *in* (cf. Booij, 1995). Then, the metrical frames of <gitaar> and the adjacent word <in> are combined. Following the maximal onset principle in syllabification, /r/ will be made onset of the new third syllable instead of coda of the second. In this way, WEAVER++ achieves syllabification across morpheme and word boundaries, which is an important property that other models in the literature are lacking (for an extensive discussion, see Roelofs, 1997b). The phonological encoding stage thus concerns what linguists traditionally call the "morphology-phonology interface" (e.g., Goldsmith, 1990).

Finally, the phonetic encoder selects the syllable program nodes whose labeled links to the segments correspond with the syllable positions assigned to the segments during the syllabification process. For example, the phonetic syllable, or syllable motor program, [tar] is selected for the second phonological syllable of *gitaar*, because the link in the network between [tar] and /t/ is labeled onset, between [tar] and /a/ nucleus, and between [tar] and /r/ coda. After selection, the phonetic encoder unpacks the syllable programs and it sets the parameters for loudness, pitch, and duration. The programs are then made available to the articulators for the control of the articulatory movements. This third encoding stage thus includes what linguists sometimes call the "post-lexical phonology" (e.g., Goldsmith, 1990).

Provided that the selection conditions of a syllable program node are met, the actual selection of the node at any moment in time is a random event. The probability of selecting a node at a particular moment in time is equal to the ratio of its level of activation and the sum of the activation levels of all syllable programs nodes in the network. Given the selection ratio, the mathematical expectation of the encoding time can be computed (Roelofs, 1997c).

The design of the form-encoding algorithm of WEAVER++ was motivated by three major problems with the existing models. Existing models such as Dell's (1986, 1988) fail to explain how to correctly retrieve the morphemes and segments of a word in the context of the activation of the morphemes and segments of another word, how to achieve syllabification across morpheme and word boundaries, and how to achieve assimilation and allophonic variation of segments. By contrast, WEAVER++ accounts for performance. In the model, binding is achieved by labeling links between morphemes and segments and verifying them, syllabification across morpheme and word boundaries is achieved by computing instead of storing syllabifications, and phonetic encoding is achieved by accessing a phonetic syllabary. For an extensive discussion, I refer to Roelofs (1997b, 1997c).

3.4 Application of the Encoding Algorithm

The WEAVER++ model has been applied to a large variety of findings about the time course of word-form encoding. It has been shown by computer simulation that the model accounts for key empirical findings about the time course of phonological facilitation and inhibition by spoken distractors in picture naming (Meyer & Schriefers, 1991). Furthermore, WEAVER++ accounts for findings on the order of encoding inside and between the syllables of a word (Meyer, 1990, 1991), for findings on the effect of word, morpheme, and syllable frequency (Jescheniak & Levelt, 1994; Levelt & Wheeldon, 1994; Roelofs, 1996b), and for speech errors (Nooteboom, 1969). Furthermore, novel predictions from the model have been tested and validated in new experiments (Roelofs, 1996b, 1996c, 1997b, 1998; Roelofs & Meyer, 1998).

Below, I discuss one application of the model. It is characteristic of WEAVER++ that its encoding algorithm operates in a rightward incremental fashion and that it requires morphologically decomposed form entries. Rightward incrementality means that a critical fragment rather than the complete output of a preceding stage initiates an encoding stage and that the stages operate from the beginning of an utterance to its end. Morphological structure is needed, because morphemes make explicit domains of syllabification (cf. Booij, 1995). For example, without morphological structure, the /r/ of the prefix *ver-* of *vereisen* ('to demand') would incorrectly be syllabified with the base *eisen*.

In testing predictions derived from rightward incrementality and morphological decomposition, I conducted experiments using the on-line preparation paradigm developed by Meyer (1990, 1991). Dutch speakers had to learn sets of word pairs (pairs of prompts and responses) and produce the second word of a pair (the response) as quickly as possible when the first word (the prompt) was presented on a computer screen. The order of prompts across trials was random. In an experiment, there were homogeneous and heterogeneous response sets. In homogeneous sets, the responses shared part of their form in common, whereas in heterogeneous sets they did not. For example, subjects learned prompt-response pairs such as *toneel - bijrol*, *long - bijnier*, etc. ('stage' – 'supporting role', 'lung' – 'kidney'), where the responses are nominal compounds that share the first morpheme *bij* (the homogeneous condition) or *long - bijnier*, *meel - deegrol*, etc. ('lung' – 'kidney', 'pancake' – 'dough roll'), with no overlap (the heterogeneous condition). The production latency (i.e., the interval between prompt onset and speech onset) was the main variable of interest.

Rightward incrementality arises from a suspend/resume mechanism in the model. The three encoding stages (i.e., morphological encoding, phonological encoding, and phonetic encoding) compute aspects of the form of a word in parallel from the beginning of the word to its end. For example, syllabification of a word can start as soon as the first few segments and metrical structure are available. The resulting partial representation can be buffered until the missing segments are available and syllabification can continue. When a stage has used the available information before reaching the end of the word, it stops and waits till it gets new input. When further information is provided, the encoding stage continues from where it stopped.

Assume the response set is *bijrol, bijvak, bijnier* ('supporting role', 'subsidiary subject', 'kidney'). Before the beginning of a trial (i.e., before the prompt word is presented on the screen), the morphological encoder can retrieve the first morpheme <bij> and the phonological and the phonetic encoders can compute the first syllable. During the trial itself, *toneel* (or another prompt word) is presented, the corresponding response word (*bijrol*) is retrieved, and the remainder of the form of the utterance is computed. In the heterogeneous condition, such preparation before a trial is not possible. So WEAVER++ predicts a facilitatory effect from response set homogeneity. The facilitatory effect should be larger if the word form can be prepared at more stages. Thus, the facilitatory effect should be larger when the shared syllable constitutes a morpheme than when it does not. If the syllable is not a morpheme, for example *bij* in the monomorphemic word *bijbel* ('bible'), only phonological preparation and no morphological preparation is possible. Furthermore, given the rightward mode of encoding, it should not be possible to prepare non-initial morphemes (e.g., the second morpheme).

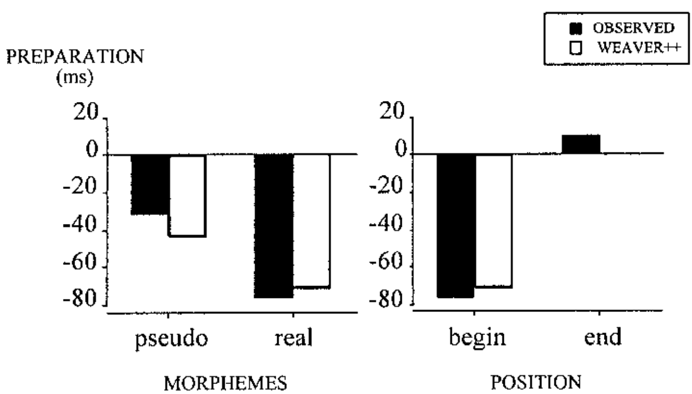


Figure 5: Preparation effects as a function of morphological status and place of overlap. Real data and WEAVER++ simulation results.

The results of the experiments confirmed these predictions. In producing disyllabic simple nouns (e.g., *bijbel*) or nominal compounds (e.g., *bijrol*), a larger facilitatory effect was obtained when the shared syllable constituted a morpheme than when it did not (i.e., the effect was larger for *bij* in *bijrol* than for *bij* in *bijbel*). No facilitatory effect was obtained from sharing the second morpheme (e.g., *bijrol, deegrol* sharing the morpheme *rol*). Figure 5 illustrates the results. The figure shows that the preparation effect for a shared syllable that constitutes a morpheme (the real condition) is larger than the preparation effect for a syllable that does not constitute a morpheme (the pseudo condition). Furthermore, a preparation effect is obtained for the first morpheme (the begin condition) but not for the second (the end condition).

4 PSYCHOLINGUISTIC VERSUS LINGUISTIC PERSPECTIVE

Whereas linguistic models are designed to account for facts about language competence or knowledge, psycholinguistic models aim at explaining facts about language performance such as how this knowledge is stored in memory and how it is accessed. WEAVER++ is a functional model that aims to account for mental processes and representations. The model is not designed to explain the working of real neural networks in the brain, but it is pitched at a more abstract level of explanation. Psycholinguistic models and linguistic models often play under different sets of rules. For example, linguistic models try to eliminate as much redundancy in representations as possible and try to capture redundancy in a rule. Psycholinguists often hold the opposite to be true: If speakers can memorize knowledge, why would they compute it? Furthermore, psycholinguistic models sometimes posit several representations for information that would be captured by one representation in a linguistic model. Syllables and conceptual chunks may illustrate these points.

The phenomenon of syllabification across morpheme and word boundaries requires that syllable structures ("phonological syllables") are on-line computed during production rather than stored with words in memory. This corresponds to linguistic models, which derive syllable structures by rule. However, in speech production there is not much use in constructing all articulatory programs ("phonetic syllables") from scratch time and again. Statistical analyses have shown that 500 different syllables already cover almost 85 percent of the syllable tokens in Dutch (and about 80 percent of those in English). Thus, even though phonetic syllables may be computed from scratch, it is advantageous for speakers to have a store of ready-made articulatory programs for syllables (Levelt, 1989, 1992; Levelt, Roelofs, & Meyer, 1999; Levelt & Wheeldon, 1994).

Linguistic models represent lexical concepts by conceptual features. These features make explicit the systematic semantic relations between words or between the meaning and the syntactic properties of a word (cf. Jackendoff, 1990). For example, speakers know that the words *bachelor* and *spinster* contrast in meaning because the former has the conceptual feature MALE and the latter not. It is a psycholinguistic issue, however, whether the memory representation of a lexical concept literally consists of nothing but the concept's features. In a decomposition view, the lemma of *bachelor* would be retrieved by the features UNMARRIED(X), HUMAN(X), ADULT(X), and MALE(X). By contrast, in a non-decomposition view, *bachelor* would be retrieved by the abstract representation or chunk BACHELOR(X). In that view, the representation BACHELOR(X) would give access to the conceptual features of BACHELOR, but would not contain them as proper parts. Note that chunking involves recoding. That is, the abbreviation BACHELOR(X) is assumed to replace a set of conceptual features. Chunking leads to a memory code that gives access to the memory codes it replaces, but which does not contain these replaced codes as proper parts. Otherwise, chunking would not have the computational advantages it has. For example, it reduces the load on working memory, because instead of several elements (e.g., UNMARRIED(X), HUMAN(X), ADULT(X), and MALE(X)) only a single element (i.e., BACHELOR(X)) has to be kept active. For the same reason, chunking diminishes attentional demands. Note that the representation BACHELOR(X) is redundant from a linguistic point of view. Whether such redundant representations are stored with words in memory is, however, a perfectly valid empirical issue from a psycholinguistic point of view.

It is important to emphasize these differences between linguistic and psycholinguistic models. For example, arguments for componential analyses are often taken to be at the same time arguments in favor of lexical decomposition (cf. Bierwisch & Schreuder, 1992; Jackendoff, 1990). A linguistic componential analysis, however, does not imply lexical decomposition. A memory system without lexical decomposition can easily implement a componential analysis and thereby account for the systematic semantic relations between words. For instance, that speakers know that the words *spinster* and *bachelor* contrast in meaning can be accounted for by assuming that the representations SPINSTER(X) and BACHELOR(X) are related in memory to the appropriate feature specifications, indicating the biological gender of a spinster and a bachelor (cf. Roelofs, 1997a).

5 SUMMARY

My aim was to illustrate the psycholinguistic perspective on the lexicon by giving an overview of a specific psycholinguistic model of the skill of lexical access in speech production, WEAVER++. The WEAVER++ model has been designed to explain facts about language performance such as the time to retrieve specific types of lexical information from memory and to compute articulatory programs rather than to account for facts about language competence. Lexical access was conceived of as a process consisting of two major steps: lemma retrieval and word-form encoding. I described the lemma retrieval algorithm of WEAVER++ and indicated how it explains relevant empirical findings about retrieval times. The discussion of conceptual chunks indicated that psycholinguistic models may posit types of representations for information that would typically be eliminated from linguistic models. In addition, I described the word-form encoding algorithm of WEAVER++ and I showed how it solves a number of computational issues and how it explains empirical findings about encoding times. The discussion of syllables indicated that psycholinguistic models may posit representations for information that would typically be captured by rules in linguistic models.

6 References

Bierwisch, Manfred, and Rob Schreuder. 1992. From concepts to lexical items. *Cognition* 42, 23-60.

Booij, Geert. 1995. *The phonology of Dutch.* Oxford, UK: Clarendon Press.

Dell, Gary. 1986. A spreading-activation theory of retrieval in sentence production. *Psychological Review* 93, 283-321.

Dell, Gary. 1988. The retrieval of phonological forms in production: Tests of predictions from a connectionist model. *Journal of Memory and Language* 27, 124-142.

Glaser, Wilhelm, and Franz-Josef Düngelhoff. 1984. The time course of picture-word interference. *Journal of Experimental Psychology: Human Perception and Performance* 10, 640-654.

Goldsmith, John. 1990. *Autosegmental and metrical phonology.* Cambridge, MA: Basil Blackwell.

Jackendoff, Ray. 1990. *Semantic structures.* Cambridge, MA: The MIT Press.

Jescheniak, Joerg, and Willem Levelt. 1994. Word frequency effects in speech production: Retrieval of syntactic information and phonological form. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 20, 824-843.

Kempen, Gerard, and Edward Hoenkamp. 1987. An incremental procedural grammar for sentence formulation. *Cognitive Science* 11, 201-258.

Levelt, Willem. 1989. *Speaking: From intention to articulation.* Cambridge, MA: The MIT Press.

Levelt, Willem. 1992. Accessing words in speech production: Stages, processes and representations. *Cognition* 42, 1-22.

Levelt, Willem, and Linda Wheeldon. 1994. Do speakers have access to a mental syllabary? *Cognition* 50, 239-269.

Levelt, Willem, Ardi Roelofs, and Antje Meyer. 1999. A theory of lexical access in speech production. *Behavioral and Brain Sciences* 22, 1-38.

Meyer, Antje. 1990. The time course of phonological encoding in language production: The encoding of successive syllables of a word. *Journal of Memory and Language* 29, 524-545.

Meyer, Antje. 1991. The time course of phonological encoding in language production: The phonological encoding inside a syllable. *Journal of Memory and Language* 30, 69-89.

Meyer, Antje, and Herbert Schriefers. 1991. Phonological facilitation in picture-word interference experiments: Effects of stimulus onset asynchrony and types of interfering stimuli. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 17, 1146-1160.

Nooteboom, Sieb. 1969. The tongue slips into patterns. In A.G. Sciarone, A.J. van Essen, and A.A. van Raad, eds., *Nomen: Leyden studies in linguistics and phonetics*, 114-132. The Hague: Mouton.

Roelofs, Ardi. 1992. A spreading-activation theory of lemma retrieval in speaking. *Cognition* 42, 107-142.

Roelofs, Ardi. 1993. Testing a non-decompositional theory of lemma retrieval in speaking: Retrieval of verbs. *Cognition* 47, 59-87.

Roelofs, Ardi. 1996a. Computational models of lemma retrieval. In T. Dijkstra, and K. De Smedt, eds., *Computational psycholinguistics: AI and connectionist models of human language processing*, 308-327. London: Taylor & Francis.

Roelofs, Ardi. 1996b. Morpheme frequency in speech production: Testing WEAVER. In G.E. Booij, and J. van Marle, eds., *Yearbook of Morphology 1996*, 135-154. Dordrecht: Kluwer Academic Publishers.

Roelofs, Ardi. 1996c. Serial order in planning the production of successive morphemes of a word. *Journal of Memory and Language* 35, 854-876.

Roelofs, Ardi. 1997a. A case for nondecomposition in conceptually driven word retrieval. *Journal of Psycholinguistic Research* 26, 33-67.

Roelofs, Ardi. 1997b. Syllabification in speech production: Evaluation of WEAVER. *Language and Cognitive Processes* 12, 657-693.

Roelofs, Ardi. 1997c. The WEAVER model of word-form encoding in speech production. *Cognition* 64, 249-284.

Roelofs, Ardi. 1998. Rightward incrementality in encoding simple phrasal forms in speech production: Verb-particle combinations. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 24, 904-921.

Roelofs, Ardi, and Meyer, Antje. 1998. Metrical structure in planning the production of spoken words. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 24, 922-939.

Spencer, Andrew. 1991. *Morphological theory*. Cambridge, MA: Blackwell.