

Chapter 7

STORAGE AND COMPUTATION IN SPOKEN WORD PRODUCTION

Ardi Roelofs

Abstract This chapter addresses the storage/computation issue in spoken word production. It is argued that word production is characterized by a storage versus computation trade-off that not only varies over the ensemble of planning stages but also within stages. Based on previous experience, the word production system has various aspects of words already prepared and stored away. To use such preparations, the system must access memory, retrieve the prepared material, and adapt it to the word at hand. The chapter reviews the WEAVER++ model of word production, spells out the model's stance on the storage/computation trade-offs, and describes relevant empirical evidence.

Keywords: Conceptual preparation, lemma retrieval, morphophonological encoding, phonetic encoding, underspecification, WEAVER++.

1. Introduction

The production of spoken words is one of our most highly exercised psychomotor skills. In normal conversation we produce about 2 to 3 words, which is about 4 syllables or 10 speech segments, per second (Levelt, 1989). This means that already with one hour talking per day, we produce 3 to 4 million word tokens, or over 5 million syllable tokens, per year. Statistics have shown that native speakers of English or Dutch produce over 80 percent of these tokens using no more than 500 different syllables, although these languages have over 10,000 different syllables (see Levelt et al., 1999b, for details). It seems therefore plausible that speakers have a memory store of motor programs for frequently used syllables and that only programs of low frequency are computed when needed. Quite impressively, in producing these large numbers of words to express our thoughts, we make only a few errors, namely about 1 error

per 1,000 words that are spoken (Levelt, 1989). An important question is what information is retrieved from memory, or is computed, to achieve the mapping of intended concepts onto motor programs for the corresponding words. And how are the concepts prepared from which words are accessed? Do speakers assemble message concepts for words out of a, possibly universal, vocabulary of primitive concepts (Bierwisch and Schreuder, 1992) or do they retrieve concepts for words from a store of prefabricated lexical concepts (Roelofs, 1992; 1997a)? The latter idea of retrieving stored lexical concepts seems plausible given that conceptually driven word retrieval does not appear to proceed by trial-and-error. Word meanings typically evade definition (e.g., Fodor et al., 1980), but speakers apparently know exactly what conceptual information to prepare to access words efficiently.

In this chapter, I address the issue of storage versus computation in spoken word production. I restrict myself to production in languages such as Dutch and English whose adult speakers typically produce word forms they have spoken before. By contrast, in languages such as Finnish and Turkish, the production of novel word forms is the rule rather than the exception (e.g., Hankamer, 1989; Koskenniemi, 1984). It is difficult to assess the relative merits of storage and computation without an explicit computational framework (e.g., Sandra, 1994). Therefore, I use a concrete model of spoken word production as a guideline for my discussion. The model is *WEAVER++* (Levelt et al., 1999a; 1999b; Roelofs, 1992; 1993; 1996b; 1996c; 1997c; 1998), which is a comprehensive, computational model of lexical access. Until recently, models of production have been designed to account for speech errors, which constitute the traditional database for production research and modeling (e.g., Dell, 1986; 1988; 1975; 1980; Shattuck-Hufnagel, 1979; Stemberger, 1985). In recent years, however, researchers have started to use chronometrical techniques and have collected latency data on production. The *WEAVER++* model recognizes the key insights from speech errors, but has specifically been designed to provide a unifying account of the increasing body of chronometrical data.

I start by describing a general theoretical framework for lexical access and indicate how access happens in *WEAVER++* (for an extensive review, see Levelt et al., 1999a; 1999b). In the model, a distinction is made between conceptual preparation, lemma retrieval, and word-form encoding, with the encoding of forms further divided into morphological, phonological, and phonetic encoding. For each of these levels of word planning, I discuss a number of theoretical and empirical arguments concerning storage and computation. But before reviewing the planning stages and the arguments, it is important to be clear about the goals.

Whereas linguistic models are designed to account for facts about language competence or knowledge per se, psycholinguistic models such as WEAVER++ aim at explaining facts about language performance such as how this knowledge is computed or stored in memory and, in case of storage, how it is accessed by speakers. WEAVER++ has been designed to account for facts about mental processes and representations, such as how long it takes to plan a particular type of word. Psycholinguistic models and linguistic models commonly play under different sets of rules. Linguistic models typically try to eliminate as much redundancy in representations as possible and try to capture redundancy in a rule. But this is usually not a major theoretical concern in developing a psycholinguistic model. On the contrary, psycholinguists often hold that when speakers can memorize a certain piece of knowledge, especially when it is frequently used, there is no reason to compute it, even when this would mean that redundant information is stored in memory. I indicate why certain design decisions have been taken in developing WEAVER++ and review relevant empirical evidence.

The view that I advocate is one in which storage versus computation in word production is not an all-or-none matter. Instead, word production can be characterized by a storage versus computation *trade-off*. Planning a word calls forth some mixture of computation and retrieval from storage. That mix not only varies over the ensemble of planning stages but also within stages. Based on previous experience, the word production system has various aspects of words already prepared and stored away. To use such preparations, the system must access memory, retrieve the prepared material, and then adapt it as appropriate to the word at hand. Stored are high-frequency aspects of words that take considerable time to assemble or aspects that are difficult or impossible to derive by rule. What is computed are aspects that are easy to derive by rule or aspects that are inherently context dependent. The WEAVER++ model captures these trade-offs in a particular way. It implements lexical knowledge symbolically as rules and stored facts but also has an activation process that determines which facts and rules get deployed for particular words.

2. Planning stages in speech production

Following most language production theories, the WEAVER++ model instantiates the assumption that speaking starts with conceptualization (but see Dennett, 1991). Conceptualization processes plan messages, which are conceptual structures to be conveyed to reach the communicative goal. Messages in WEAVER++ make explicit the intended lexical

concepts and their relationships. Lexical concepts are concepts that correspond to the meaning of words. Next, formulation processes take the message, retrieve appropriate words for the lexical concepts, and build syntactic and morphophonological structures, resulting in a phonetic plan for the utterance. Phonetic plans in WEAVER++ make explicit motor programs for the syllables in the utterance. Finally, articulation processes execute the motor programs, which yields overt speech.

Lexical access in WEAVER++ consists of two consecutive stages, namely lemma retrieval and word-form encoding, which are stages of access that are part of the formulation stages of syntactic and morphophonological encoding, respectively (see Caramazza, 1997, for a proposal without lemma retrieval, and a comment on this proposal by Roelofs et al., 1998). In lemma retrieval, a message concept is used to retrieve a lemma from memory, which is a representation of the syntactic properties of a word, crucial for its use in sentences. For example, the lemma of the word *to evade* says that it is a transitive verb and specifies which syntactic environment the word requires. Lemma retrieval makes these properties available for syntactic encoding processes. Furthermore, lemmas contain slots for the specification of abstract morphosyntactic parameters such as aspect (progressive, perfect), tense (past, present), number (singular, plural), and person (first, second, third). In word-form encoding, the lemma and the information in the slots are used to retrieve the morphophonological properties of the word from memory in order to construct an appropriate articulatory program. For example, for first-person present tense *evade* the morpheme <evade> and the segments /ɪ/, /v/, /eɪ/, and /d/ are retrieved and a phonetic plan for [ɪ.'veɪd] is generated.

Figure 7.1 illustrates the stages of access in WEAVER++. Assume a speaker wants to convey the concept TO EVADE and wants to refer to it as an ongoing activity. Expressing this requires mapping the lexical concept EVADE(X,Y) onto the articulatory program for the form *evading*. First, the lemma retriever takes EVADE(X,Y) and delivers the lemma of *evade*. This includes making available the word class and subcategorization features. In order to derive the appropriate word form, progressive [ɪ.'veɪ.dɪŋ] instead of, for example, first-person present tense [ɪ.'veɪd] or past tense [ɪ.'veɪ.dɪd], the lemma retriever has to inspect the message for the relevant temporal information (not shown in Figure 7.1) and has to fill the corresponding morphosyntactic slots. (I ignore for present purposes that in English, the progressive is a two-word form of a verb, consisting of the auxiliary *be* followed by the present participle of the verb. Thus, the slot value “progressive” will have to lead to the retrieval of the lemma of the auxiliary. I refer to Levelt, 1989, and Levelt et al.,

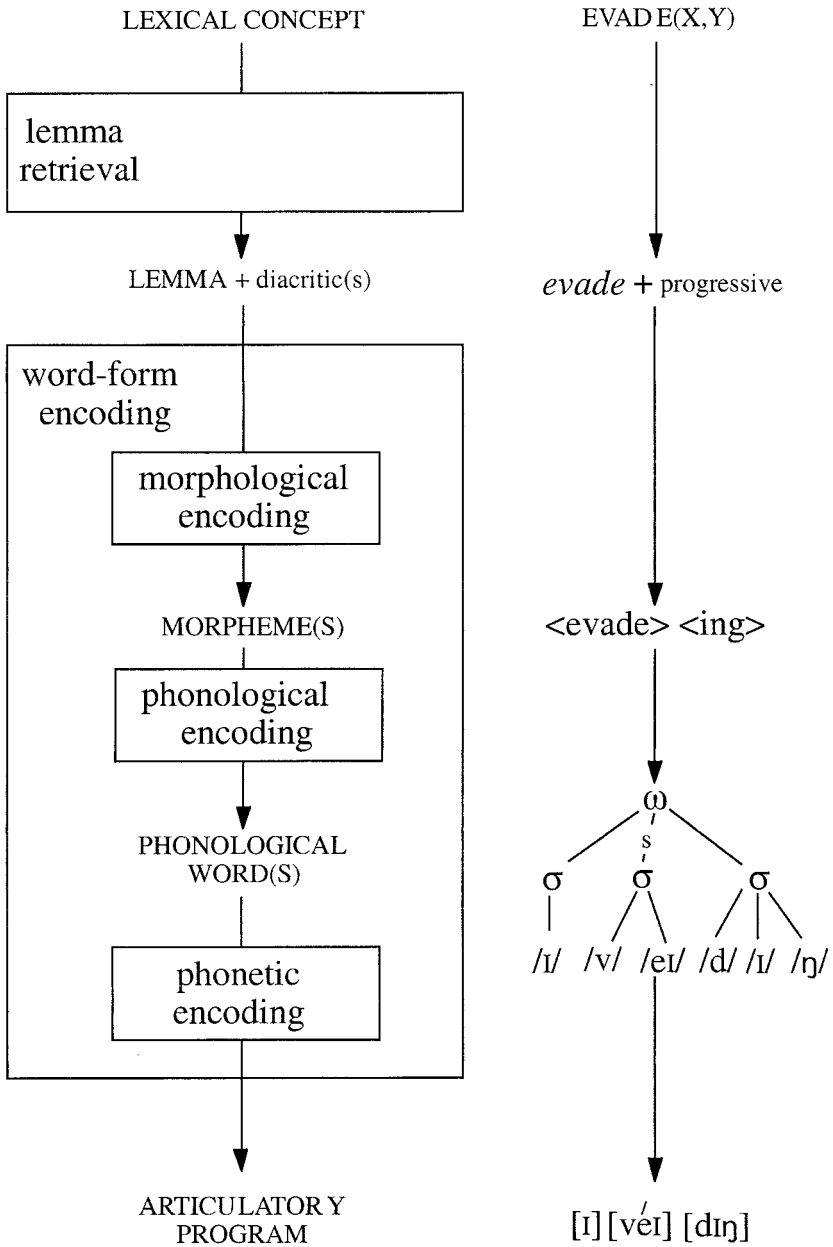


Figure 7.1. Stages of lexical access in WEAVER++.

1999b, for discussion.) The lemma plus slot values are then input to word form encoding. The articulatory program is derived in three major steps, namely morphological, phonological, and phonetic encoding (cf. Dell, 1986; Levelt, 1989). The morphological encoder takes the lemma of *evade* plus the slot value for “progressive” and produces the stem morpheme <evade> and affix <ing>. The phonological encoder takes these morphemes and produces a phonological word representation, which makes explicit the syllables and the stress pattern. This representation describes the progressive form of *evade* as a phonological word (ω) consisting of three syllables, the second carrying word accent(s). The first syllable (σ) has /i/ as nucleus. The second, stressed syllable has /v/ as onset and /eɪ/ as nucleus. The third syllable has /d/ as onset, /ɪ/ as nucleus, and /ŋ/ as coda. Finally, the phonetic encoder takes this phonological word representation, accesses a store of learned motor programs for syllables (Levelt and Wheeldon, 1994), and delivers the corresponding articulatory program for [i.'veɪ.dɪŋ].

3. Storage or computation?

3.1. A case for storage in conceptual preparation and lemma retrieval

Linguistic models often 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 has not. It is a psycholinguistic issue, however, whether the message representation of a lexical concept literally consists of nothing but the concept's features. In a decompositional view on memory representation, *bachelor* would be represented in the message by features like UNMARRIED(X), HUMAN(X), ADULT(X), and MALE(X). By contrast, in a non-decompositional view, *bachelor* would be represented by the abstract representation or “chunk” BACHELOR(X). In that view, the representation BACHELOR(X) would point to and give access to conceptual features such as UNMARRIED(X) in long-term memory but would not contain these features as proper part. Note that chunking involves recoding, which is sometimes overlooked. For example, Jackendoff, 1990, p. 38 assumes that one can have chunks without a loss of information. However, the chunk BACHELOR(X) is assumed to replace a set of conceptual features. A chunk is a memory code that gives access to the memory codes it replaces (like a speed dial button on a telephone, e.g., M8 might stand for the telephone number

3521911), but the chunk does not contain these other codes as proper part. Otherwise, chunks would not have the computational advantages they have. For example, chunks may reduce the load on short-term memory and demands on attention, 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 in short-term memory (or "M8" instead of "3521911"). 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 (see Roelofs, 1997a, for discussion).

In lemma retrieval, a message concept has to be mapped onto the lemma of a word in memory, for which there are two main theoretical options (Levelt, 1989; 1992; Roelofs, 1992; 1997a). On the one hand, the message encoder may compute a message concept for a word using a vocabulary of conceptual primitives, and the lemma retriever may access lemmas on the basis of the temporarily assembled set of primitive concepts (cf. Bierwisch and Schreuder, 1992). On the other hand, the message encoder may include lexical concepts in the message as ready-made wholes and lemmas may be accessed from these wholes (cf. Fodor et al., 1980). For example, Dell and colleagues (Dell, 1986; 1988; Dell and O'Seaghdha, 1992) assume that the mental lexicon is an associative network in which nodes for conceptual features (e.g., IS-TO AVOID(X,Y), IS-TO ESCAPE(X,Y), etc.) are connected to lemmas (e.g., *evade*). The lemma of *evade* is retrieved by activating a set of conceptual features such as IS-TO AVOID(X,Y), IS-TO ESCAPE(X), and so forth. Thus, in this view, message concepts for the retrieval of words are assembled. Although a speaker has stored in memory which conceptual features go with which lemmas, the message encoder computes concepts for word retrieval by activating a set of conceptual features corresponding to the intended thought. By contrast, under the non-decompositional account, the message encoder prepares a message by retrieving lexical concepts as chunks such as EVADE(X,Y). The lemma of *evade*, then, is retrieved on the basis of the chunk EVADE(X,Y) instead of features such as IS-TO AVOID(X,Y), IS-TO ESCAPE(X), and so forth. These features are connected to the chunk EVADE(X,Y) in memory, and support thought and reasoning, but are not involved in the actual retrieval of lemmas. Note that the functional role of lexical concepts (part of conceptual processes) and lemmas (part of syntactic encoding) is very different.

The computation of message concepts for words, as defended by Bierwisch and Schreuder, 1992, confronts models with a number of problems concerning convergence that have not been appropriately dealt with yet

(see Levelt, 1989, and Roelofs, 1992; 1993; 1996a; 1997a, for extensive discussion). Although solutions have been proposed within a decompositional framework for some of the problems (e.g., Bierwisch and Schreuder, 1992; Bowers, 1999; Caramazza, 1997; Zorzi and Vigliocco, 1999), none of the proposals solves the class of problems as a whole. The convergence problems are (1) how to correctly dissect a thought into lexical concepts during message encoding, (2) how to avoid retrieving hyponyms or hyperonyms along with or instead of the intended words, and (3) how to correctly retrieve a single word instead of several words for a synonymous phrase, or vice versa. I discuss these problems in turn.

In preparing concepts for the retrieval of words, the message encoder has to know which sets of conceptual features correspond to the meaning of words, since only "lexical concepts" can be verbalized by single words. If there is no match, the retrieval system would halt and no word would be retrieved. This is the dissection problem in message encoding (Levelt, 1992). For example, the thought MALE PARENT corresponds to a single lexical concept but the thought YOUNG PARENT does not. The dissection problem is nontrivial because word meanings typically evade definition (e.g., Fodor et al., 1980). What further complicates matters are set inclusions. For example, the set of features of a specific word such as *father* contains the features of its hyperonym *parent* as a proper subset (i.e., *father* has all the features of *parent* plus a few extra ones, for example indicating the sex). Which features should be activated in retrieving a lemma and which subset of features to retrieve its hyperonyms? The hyperonymy problem concerns how to avoid retrieving the word *parent* along with or instead of its hyponym (the retrieval target) *father*. Several models suffer from this problem (e.g., Miller and Johnson-Laird, 1976). But the opposite hyponymy problem also haunts models. For example, in a spreading activation network like that proposed by Dell and O'Seaghdha, 1992, the conceptual primitive PARENT(X,Y) is linked to both the lemma node of *parent* and that of *father*. So in activating PARENT(X,Y) to retrieve *parent*, both *parent* and *father* will attain the same level of activation. This problem may perhaps be solved by tuning the strength of the connections between conceptual features and lemmas. The feature PARENT(X,Y) should be strongly connected to *parent* but weakly to *father*. This would certainly seem to be an improvement but it does not solve all problems. The word *father* and the phrase *male parent* are synonymous. The word-to-phrase synonymy problem concerns how to avoid retrieving *father* along with or instead of *male* and *parent* for the phrase *male parent*, or vice versa. Tuning connection strengths is insufficient to solve this problem, because the same primitives are involved in producing the single word and the phrase. Perhaps

this problem may be solved by sequentially activating the conceptual primitives in producing a phrase. This would seem to be an option in theory but it runs into empirical difficulties. There is empirical evidence that suggests that lemmas making up a phrase are planned in parallel (Meyer, 1996).

In summary, conceptually preparing lemma retrieval may involve assembling message concepts out of a vocabulary of primitive concepts. Alternatively, messages may be prepared by retrieving stored lexical concepts. Consequently, lemmas may be retrieved in a conceptually decomposed or non-decomposed fashion. Existing decompositional theories fail to account for convergence in conceptually driven lemma retrieval. This argues for storing rather than computing lexical concepts and for conceptually non-decomposed rather than decomposed lemma retrieval.

3.2. A case for computation in morphophonological encoding

Most existing models assume that word forms are stored in memory as sequences of syllable nodes and that each consonant is stored as an onset or coda (e.g., Dell, 1986; 1988; Shattuck-Hufnagel, 1979). For example, Dell (Dell, 1986; 1988) assumes that the form lexicon is a network that contains nodes for morphemes (e.g., <evade>), syllables (e.g., /i/ and /verd/), segments (marked for syllable position, e.g., /onset v/, /nucleus ei/, and /coda d/), and phonological features (e.g., voiced). Furthermore, there are nodes that specify the abstract CV structure of a word (Dell, 1988). However, by storing syllable structures these models run into difficulties because syllabification is context dependent. Often the syllabification of segments ignores morpheme and word boundaries in that a segment of one morpheme or word is syllabified with an adjacent morpheme or word. This may occur in the production of polymorphemic words or connected speech (cf. Levelt, 1989; Roelofs, 1997c). By rigidly storing words as sequences of syllable nodes and storing each consonant as an onset or coda, these models have a difficult time dealing with the flexibility of syllable membership. As of yet, this problem has not been solved within these models (see Roelofs, 1997b, for extensive discussion).

Consider, for example, the production of the progressive form *evading*. The progressive form is created by adding <ing> to the stem <evade>. The resulting form is syllabified as $(i)_\sigma(vei)_\sigma(dmj)_\sigma$. Thus, juxtaposing *-ing* to the stem changes the syllabification of /d/. This segment occupies a coda position in *evade*, syllabified as $(i)_\sigma(verd)_\sigma$, but an onset position in *evading*. Or consider the production of connected speech. For example, in producing “evade it”, *it* may be adjoined to *evade*. This yields

the new phonological word *evadit*, which is syllabified as $(i)_\sigma(v\epsilon i)_\sigma(dit)_\sigma$. In models that store syllable structures, however, segments are marked for syllable position, so we have /coda d/ for *evade*. Models like Dell's (Dell, 1986; 1988) prohibit selecting this coda node for an onset position, which would be required for the production of *evading* and *evadit*.

The syllabification across morpheme and word boundaries asks for computing rather than storing syllable structures, as argued by Levelt, 1992 and Roelofs, 1997b. The computation of syllable structures has consequences for the design of other aspects of the production system, namely it requires morphologically decomposed form entries for languages such as Dutch. Morphological structure is needed, because several morphemes are separate domains of syllabification (cf. Booij, 1995). In Dutch, this holds for prefixes such as *ver-* and *ont-* and suffixes such as *-achtig*, but not for suffixes such as *-ing* and *-en*. For example, without morphological structure, the /t/ of the Dutch prefix *ont-* of *ontwijken* 'evade' would incorrectly be syllabified with the base *wijken* following the maximal onset principle. Some morphemes such as the plural suffix *-en* are not independent domains of syllabification. For example, the /k/ in *ontwijken* is syllabified with *-en*. Morphological complexity can play a role in form planning without having a synchronic semantic motivation. Morphemes may be separate domains of syllabification independent of semantic transparency. For example, the /r/ of the opaque prefixed verb *verijdelen* 'frustrate' is syllabified with the prefix *ver-* and not with the base as the maximal onset principle would predict. In form planning, morphology appears to operate "by itself" (cf. Aronoff, 1994). In WEAVER++, morphology is "word based" in that no attempt is made to derive the morphemes of a word from its meaning. Which morphemes make up the stem of words is stored by connecting morpheme nodes to lemmas in memory. In planning a derivationally complex word or compound, the stem morphemes are simply accessed from the lemma and used to assemble the word form (see Levelt et al., 1999b, for extensive discussion).

3.3. A case for storage in phonetic encoding

The phenomenon of syllabification across morpheme and word boundaries requires that "phonological syllables" are computed during production rather than stored with words in memory. This corresponds to linguistic models, which derive syllable structures by rule. However, since speakers use only a few hundred different syllables for most of their talking, there is not much use for them in constructing all articulatory programs ("phonetic syllables") from scratch time and again.

Thus, it is plausible to assume that in phonetic encoding, the motor programs are typically not computed but retrieved (for an extensive discussion, see Levelt, 1989; 1992; Levelt et al., 1999b; Levelt and Wheeldon, 1994). Learned motor programs must present a set of retrieval cues to higher-level processes so that the appropriate programs can be accessed. Ideally, these retrieval cues constitute a reasonably small set. Levelt (Levelt, 1992; Levelt and Wheeldon, 1994) has proposed that the retrieval cues for articulatory programs are the phonological syllables that are constructed as part of phonological word representations. Programs that are not stored will be computed on the basis of the information in the phonological word representation. Syllabary access translates an abstract phonological representation into a context-dependent phonetic representation that can guide articulation. For example, the articulatory program for *evade* [i.'veid] comprises motor programs for the syllables [i] and [veid], where [i] and [veid] stand for packages of gestural scores for the articulatory movements to be made. A score specifies the gestures and their temporal relationships (e.g., Browman and Goldstein, 1986; Levelt, 1989; 1992). Scores make explicit articulatory tasks, such as lip protrusion and lowering of the jaw. The details of the movements realizing these scores are left to the articulatory system.

To summarize, I have discussed a number of arguments concerning the relative merits of storage and computation. It makes sense to store high-frequency aspects of words that take considerable time to assemble, such as motor programs, and aspects that are difficult or impossible to derive by rule, such as lexical concepts. And it is plausible to compute those aspects of words that are easy to derive by rule and aspects that are inherently context-dependent, such as syllabifications. In the next section, I discuss in some detail how the trade off between storage and computation has been captured in the *WEAVER++* model.

4. The *WEAVER++* model

Like many models, *WEAVER++* implements the mental lexicon as an associative network of nodes, links, and link labels that is accessed by spreading activation. *WEAVER++* deals with the dissection problem, the word-to-phrase synonymy problem, and other convergence problems by assuming that each lexical concept is represented in the network by an independent node (cf. Collins and Loftus, 1975). For example, the network contains the nodes *EVADE(X,Y)* and *AVOID(X,Y)* connected by a link labeled *IS-TO* (to evade something is to avoid something). The node *EVADE(X,Y)* is connected to the lemma node for *evade*. Conceptually preparing a word consists of selecting lexical concept nodes. Lemmas

are accessed from the lexical concept nodes rather than from assemblies of conceptual feature nodes. The model handles the problem of syllabification across morpheme and word boundaries by assuming that syllable positions are not stored with words in memory (e.g., there are no /onset d/ and /coda d/ nodes), but that syllable positions are assigned on-line by a syllabification process (cf. Levelt, 1992). The assignment of segments to syllable positions takes neighboring morphemes and words into account. Syllable positions of segments are computed for phonological words rather than for lexical ones. Finally, the model implements the assumption that speakers have a phonetic syllabary, a store of motor programs for frequently used syllables, which is accessed on the basis of the phonological syllables constructed as part of phonological word representations (Levelt and Wheeldon, 1994).

Words are not planned by a central agent that overlooks the whole process but by several "production rules" that work in parallel on small parts of the word. Production rules are condition-action pairs. If the condition of a rule is met, the action is performed. The production rules are stored with the nodes and have a limited overview only. Activation of nodes in the network triggers production rules that select lemmas and incrementally build phonetic plans by selecting and connecting form nodes. When the activation of a node exceeds threshold, a production rule verifies the link between the node and the selected nodes one level up in the network. Syntactic production rules select the lemma node linked to the target lexical concept node. For example **IF the lemma is linked to the message concept and the activation of the lemma exceeds threshold THEN select the lemma**. Morphological production rules select the morpheme nodes that appropriately encode a selected lemma node and its tense, aspect, and agreement specification. Phonological production rules select the segments linked to the morpheme nodes and prosodify the segments in order to construct phonological word representations. Prosodification consists of syllabification and stress assignment. Finally, phonetic production rules select the syllable program nodes that appropriately encode the constructed phonological syllables, and access the corresponding syllable programs in the syllabary.

Figure 7.2 illustrates the structure of lexical entries in *WEAVER++*'s associative network. In particular, it shows the memory representation of the word *evade*. A lexical network with nodes and labeled links is connected to a syllabary with learned motor programs (the syllabary is omitted from Figure 7.2). The lexical network consists of three major strata: a conceptual, a syntactic, and a word-form stratum. The conceptual stratum contains concept nodes and labeled conceptual links. Each lexical concept in the language, for example *EVADE*(*X*, *Y*), is represented

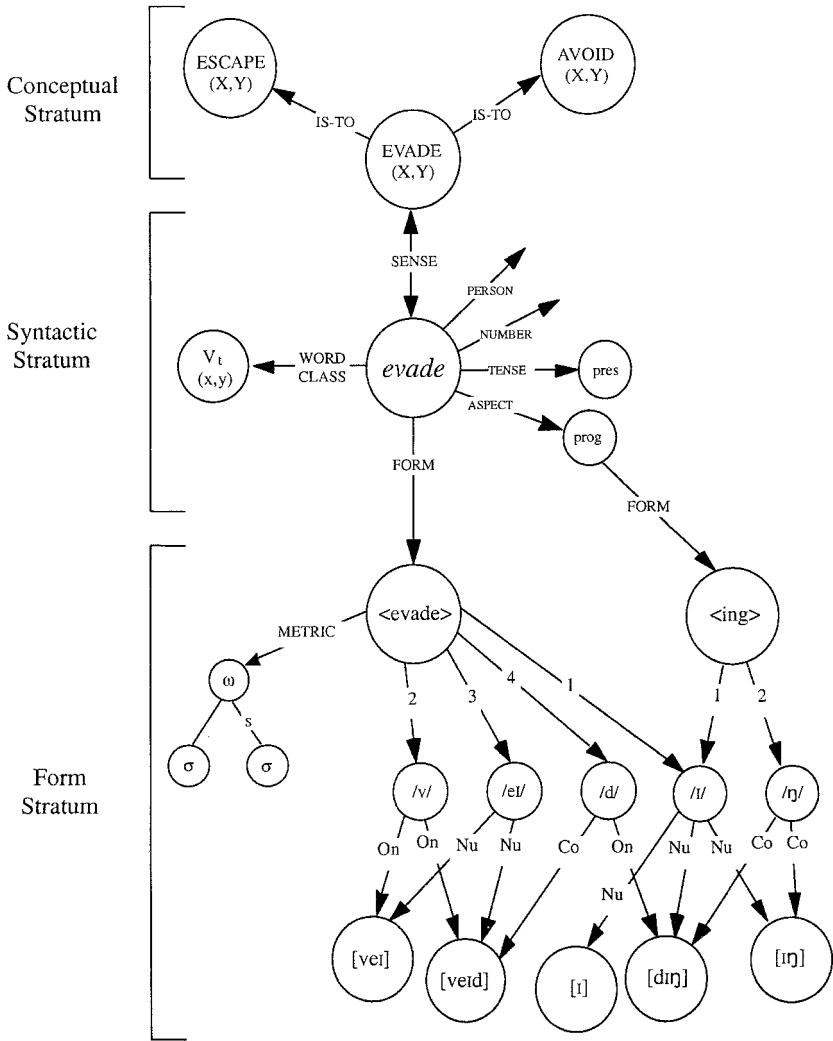


Figure 7.2. Representation of the word "evade" in WEAVER++.

by an independent node. The links specify conceptual relationships. The syntactic stratum contains lemma nodes (*evade*), syntactic property nodes and labeled links (e.g. WORD CLASS: Verb), and slots for the specification of morphosyntactic parameters (e.g., TENSE: present). The word-form stratum contains metrical structure, morpheme, segment, and

syllable program nodes and links. Morpheme nodes are connected to a lemma and its diacritics. The links between morphemes and segments specify the serial position of the segments. The links between segments and syllable program nodes specify possible — as opposed to actual — syllabifications. The word-form stratum is connected to a syllabary, storing ready-made motor programs for high-frequency syllables.

The WEAVER++ model implements the claim that lexical concepts are stored in memory rather than computed on-line and that lemmas are retrieved in a conceptually non-decomposed way. That is, for example, the verb *evade* is retrieved on the basis of the chunk $EVADE(X,Y)$ instead of features such as $IS-TO\ AVOID(X,Y)$, $IS-TO\ ESCAPE(X)$, and so forth. Retrieval starts by enhancing the level of activation of the node of the target lexical concept. Activation then spreads through the network, each node sending a proportion of its activation to its direct neighbors. The most highly activated lemma node is selected. For example, in verbalizing the thought $EVADE$, the activation level of the lexical concept node $EVADE(X,Y)$ is enhanced. Activation spreads through the network, whereby the lemma nodes *evade* and *avoid* will be activated, among other nodes. The *evade* node will be the most highly activated node, because it receives a full proportion of the activation of $EVADE(X,Y)$, whereas *avoid* and other lemma nodes receive only a proportion of a proportion of the activation of $EVADE(X,Y)$. Upon verification of the link between the lemma node of *evade* and $EVADE(X,Y)$, this lemma node will be selected.

The basic theoretical claim implemented in WEAVER++ concerning word-form encoding is that lemmas are mapped onto learned syllable-based articulatory programs by rightward incrementally computing morphological and phonological structures. The latter make explicit phonological syllables. These phonological syllables are then used to retrieve stored motor programs from a phonetic syllabary. Figure 7.2 illustrates the form representation of *evade* in WEAVER++. The non-metrical part of the form network consists of three layers of nodes: morpheme, segment, and syllable program “address” nodes. Morpheme nodes stand for roots and affixes. Morpheme nodes are connected to the lemma and its morphosyntactic parameters. For example, the stem <evade> is connected to the lemma of *evade*. A morpheme node points to the segments that make up its underlying form, and, for some words, to its metrical structure. For storing metrical structures, a principle of economy applies. WEAVER++ assumes that the main accent of Dutch (and English) words is on the first syllable containing a full vowel (which holds for more than 90 percent of the word tokens), unless the lexical form representation indicates otherwise (Levelt et al., 1999b). Thus, for

polysyllabic words that do not have main stress on the first stressable syllable, the metrical structure is stored as part of the lexical entry, but for monosyllabic words and for all other polysyllabic words, it is not. For example, the metrical structure for *evade* [ɪ.'veɪd] is stored, but for *table* ['teɪ.bəl] it is not. Stored metrical structures describe abstract groupings of syllables (σ) into phonological words (ω). Importantly, it is not specified which segments make up the syllables nor is the CV pattern specified. 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 /d/ is the coda of [veɪd] and the onset of [dɪŋ]. Segments also point directly to their phonological features (not shown in Figure 7.2), which allows computing motor programs in case there is no stored program in the syllabary.

Encoding starts when a morpheme node receives activation from a selected lemma. Activation then spreads through the network in a forward fashion. The morphological encoder selects the morpheme nodes that are linked to a selected lemma and its parameters. Thus, <evade> and <ing> are selected for *evade* and the specification "progressive". In general, a lemma and its diacritics (e.g., *evade* + "past"; *eat* + "past") correspond to a stem and its affixes (either decomposed <evade> + <ed> or irregular, non-decomposed <ate>) at the word form level. For every word class there is a paradigm, that is, a set of inflectional forms (regular or irregular) that encode the fixed set of grammatical functions specified by the diacritics. Some diacritics obtain their value by agreement with other parts of the utterance (e.g., person, number: *evades* versus *evade*) during the process of syntactic encoding. The stems at the word-form level may be simple or complex. Complex stems may have been created by derivation in the language (i.e., base morphemes plus one or more affixes, e.g., *exhale*) or compounding (i.e., existing words added together, e.g., *afterthought*).

The phonological encoder selects the segments and, if available, the metrical structures that are linked to the selected morpheme nodes. Next, the segments are input to a prosodification process that associates the segments to the syllable nodes within the metrical structure (for "exception" words) or constructs metrical structures based on segmental information. Thus, when stored, metrical structures are retrieved and woven into the phonetic plan, otherwise they are constructed on the spot. The prosodification proceeds from the segment whose link is labeled first to the one labeled second, and so forth. In the prosodification, syllable positions (onset, nucleus, coda) are assigned to the segments fol-

lowing the syllabification rules of the language. Essentially, each vowel and diphthong is assigned to a different syllable node and consonants are treated as onsets unless phonotactically illegal onset clusters arise. In the encoding of *evade*, the /ɪ/ is made nucleus of the first syllable, and the /v/ onset, the /eɪ/ nucleus, and the /d/ coda of the second syllable. The prosodification process provides for cross-morpheme and cross-word syllabification. In planning polymorphemic words or connected speech, the structures of adjacent morphemes or words may be combined. This leads to new phonological words. For example, *WEAVER++* may syllabify <ing> with <evade> for the progressive form *evading* or it may prosodify the stem <evade> and <it> together for the cliticization *evadit*. Then, following the maximal onset principle in syllabification (e.g., Goldsmith, 1990), /d/ will be made onset of the third syllable instead of coda of the second syllable, yielding $(i)_\sigma(vɛɪ)_\sigma(dɪŋ)_\sigma$ and $(i)_\sigma(vɛɪ)_\sigma(dɪt)_\sigma$. In this way, *WEAVER++* achieves syllabification across morpheme and word boundaries. Note that, again, a storage versus computation trade-off is involved. For <evade> the metrical structure is stored but for <ing> it is not. In prosodifying <ing> with <evade>, the system retrieves the metrical structure for <evade> and adapts it such that it can accommodate <ing> (i.e., a third syllable node σ is added to the structure).

The phonetic encoder selects the syllable program "address" nodes whose labeled links to the segments correspond with the syllable positions assigned to the segments. For example, [veɪd] is selected for the second phonological syllable of *evade*, because the link between [veɪd] and /v/ is labeled onset, between [veɪd] and /eɪ/ nucleus, and between [veɪd] and /d/ coda. Similarly, the phonetic encoder selects [veɪ] and [dɪŋ] for the progressive form *evading* and [veɪ] and [dɪt] for the form *evadit*. Finally, the phonetic encoder addresses the actual syllable programs in the syllabary, thereby making the programs available to the articulators for the control of the articulatory movements (following Levelt, 1992; Levelt and Wheeldon, 1994). Certain adaptations have to be made to the retrieved motor programs. In particular, the encoder uses the metrical representation to set the programs' parameters for loudness, pitch, and duration. Also, the encoder has to take care of the transitions at syllable boundaries. After making the adaptations, the phonetic plan is ready to govern articulation.

WEAVER++ provides for a suspend/resume mechanism that supports incremental generation of word forms. Incremental planning means that encoding processes are triggered by a fragment of their characteristic input (Levelt, 1989). The three form encoding stages (i.e., morphological, phonological, and phonetic encoding) compute aspects of a word form in

parallel from the beginning of the word to its end. For example, syllabification can start on the initial segments of a word without having all of its segments. Only initial segments and, for some words, the metrical structure are needed. When given partial information, computations are completed as far as possible, after which they are put on hold. The computed representation is buffered until the missing segments are available and syllabification can continue. When given further information, the encoding processes continue from where they were left.

5. Storage in conceptual preparation and lemma retrieval

A first test of a model is to see whether it accords with existing data. Below I show that the claim of storage of lexical concepts and non-decomposed retrieval correctly predicts several empirical findings on spoken word production. The findings are from “picture-word interference” experiments. Speakers have to name pictured objects while simultaneously trying to ignore written words superimposed on the pictures. The measurement of interest is the speech onset latency, that is, the difference in time between picture onset and production onset. Elsewhere (in Roelofs, 1992), I have shown by computer simulation that WEAVER++ explains, among other phenomena, the classical semantic effects of picture and word distractors in picture naming, picture categorizing, and word categorizing. Here, I want to concentrate on those aspects of these findings that are particularly relevant for the claim of storage in conceptual preparation and lemma retrieval. In particular, Glaser and Dünghoff, 1984 and Roelofs, 1992; 1993 have shown that naming is facilitated when the distractors are hyponyms of the target. Glaser and Dünghoff showed that in picture categorizing, a semantic facilitation effect is obtained from written distractor words. For example, producing the word *furniture* in response to a pictured chair is speeded up by the distractor word *bed* compared to distractor *fish*. Similarly, in word categorizing a semantic facilitation effect is obtained from distractor pictures. For example, producing *furniture* in response to the word *chair* is facilitated by a pictured bed compared to a pictured fish.

Semantic facilitation is not restricted to distractors that are basic object level names such as *chair*, *bed* and *fish*. I have shown (see Roelofs, 1992) that semantic facilitation is also obtained in naming a pictured object, for example, in naming a chair while the distractor words *furniture*, *bed*, or *throne* are superimposed. In addition to the type of distractor, the experiment manipulated the stimulus onset asynchrony (SOA), that is, the time difference between the presentation of the picture and the

distractor word. The written distractors were presented 100 msec before (referred to as a negative SOA), simultaneously with, or 100 msec after picture onset (called a positive SOA). The finding was a semantic facilitation effect at the earliest SOA of -100 msec and no effect at later SOAs. Figure 7.3 plots the semantic facilitation effect against SOA. The semantic effect was the same for hyperonym (*furniture*), cohyponym (*bed*), and hyponym (*throne*) distractors compared to unrelated distractors. The bars represent means across these three types of distractor word. Semantic facilitation has also been obtained for hyponym distractor verbs in producing verbs rather than nouns, for example when participants had to say *drink* to a drinking person and ignore the distractors *booze* or *whimper* (Roelofs, 1993).

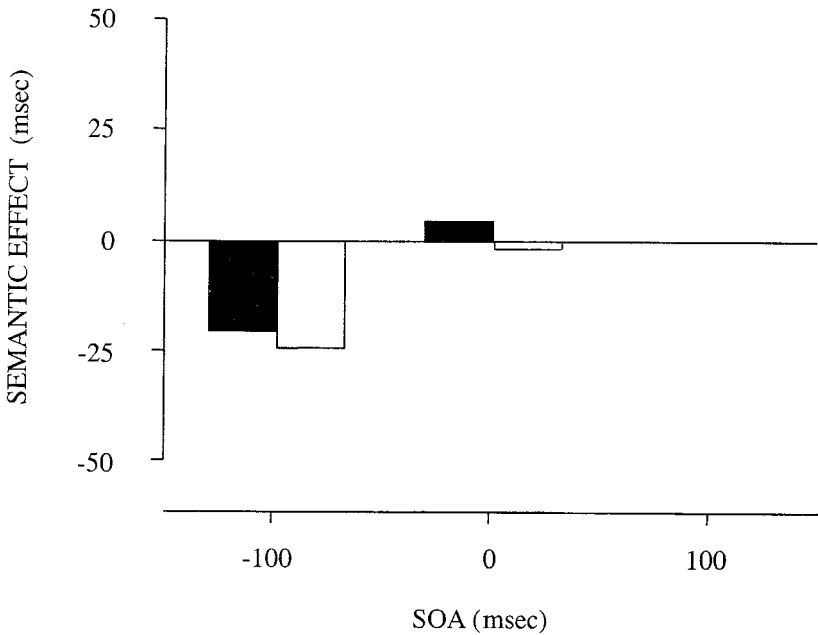


Figure 7.3. The semantic effect for picture naming with written distractor words per SOA: ■ = empirical data (Roelofs, 1992) and □ = predictions by WEAVER++.

Figure 7.3 also shows what WEAVER++ predicts for the noun production experiment. In WEAVER++ a “shortlist” of target lemmas can be defined depending on the task that is set for the retrieval system (e.g., in a categorization task the response set consists of hyperonyms such as *furniture*, *animal*, etc.). Competition is restricted to these shortlisted lemmas. The assumption is that speakers set up the shortlist before an experiment when they receive a booklet with the pictures and names to

be used. Thus, in the model, only potential target responses will compete for selection. In case of picture or word categorization, *furniture* and *animal* are the targets and will compete, but *chair*, *bed*, *fish*, and *dog* will not. The distractor *bed* superimposed on a pictured chair will activate the target *furniture* via the conceptual network, but *bed* will not be a competitor for *furniture* because *bed* is not a permitted response in the experiment. By contrast, *fish* on a pictured chair will activate *animal*, which is a competitor of the target *furniture*. Thus, semantic facilitation is predicted and this is exactly what is empirically obtained (see Figure 7.3). A chi-square measure of fit showed that the predictions of the model do not differ statistically from the real data.

The semantic facilitation effects in picture and word naming and categorizing demonstrate two important points. First, contrary to what has been suggested in the literature (cf. Caramazza, 1997), the explanation of semantic effects (in picture-word interference) does not require semantic decomposition. What is required is that lemmas of related concepts are connected in memory and this may be achieved by semantic decomposition or by a semantic network as implemented in WEAVER++. Second, the finding of semantic facilitation from hyponyms excludes one type of solution to the hyperonymy problem in lemma retrieval. Bierwisch and Schreuder, 1992 have proposed a decompositional model in which the convergence problem is solved by inhibitory links between hyponyms and hyperonyms (i.e., words inhibit their hyperonyms). For example, in producing *chair*, activating the lemma of *chair* leads automatically to inhibition of the lemma of its hyperonym *furniture*. However, the existence of such inhibitory links predicts semantic inhibition from hyponym distractors (e.g., distractor *throne* should inhibit target *chair*), but facilitation is what has been empirically obtained (empirically, distractor *throne* facilitates the production of *chair*, see Figure 7.3). Also, the finding of semantic facilitation in picture and word categorizing refutes such an inhibitory link between words and their hyperonyms (e.g., between *chair* and *furniture*). In general, these semantic facilitation effects pose difficulty to models in which lemma selection is achieved by hard-wired competition among lemmas, as in the models of Starreveld and La Heij, 1996 and Cutting and Ferreira, 1999. Instead, the facilitation argues in favour of dynamic competition in lemma selection (i.e., the shortlist idea).

It is well known from the literature on human memory that when information is frequently accessed, the retrieval time decreases. This also appears to hold for the retrieval from lexical storage. WEAVER++ accounts for such frequency effects in word production. Frequency ef-

facts in the model originate from differences in the speed of applying production rules, which depends on frequency of usage.

Experiments by Jescheniak and Levelt, 1994 have shown that when lemma information such as the grammatical gender of a noun is accessed, a frequency effect is obtained. For example, Dutch participants had to decide on the gender of a picture's name (e.g., they had to decide that the grammatical gender of the Dutch word *tafel* (*table*) is non-neuter), which was done faster for high-frequency words than for low-frequency ones. The effect disappeared with repetition, contrary to a "robust" frequency effect obtained in naming the pictures.

Jescheniak and Levelt provided evidence that the locus of the robust frequency effect is the form level. When participants had to respond to an English probe word by producing its Dutch translation equivalent, the production latency of a low-frequency homophone was determined by the sum frequency of that word and its high-frequency counterpart. For example, participants had to produce the Dutch word *bos* in response to *bunch* (low-frequency reading). The production latencies for these homophones were compared to the latencies for two types of other words. First, there were low-frequency control words whose frequency was matched to that of the low-frequency reading of the homophone. The low-frequency control for *bos* was *hok* (*kennel*). Second, there were high-frequency control words whose frequency was matched to the sum frequency of the low-frequency reading (i.e., *bunch*) and high-frequency reading (i.e., *forest*) of *bos*. The high-frequency control for *bos* was *hoek* (*corner*). Producing the homophones (*bos*) in their low-frequency reading went as fast as producing the high-frequency controls (*hoek*), and it went faster than producing the low-frequency controls (*hok*). Thus, a low-frequency homophone inherits the frequency of its high-frequency counterpart. In WEAVER++, homophones have their form nodes and production rules in common but not the lemma. By sharing form nodes and production rules, a low-frequency homophone inherits the frequency properties of its high-frequency counterpart. This explains the homophone effect observed by Jescheniak and Levelt.

6. Computing morphological structures

In testing for the assembly of morphologically complex forms in production, I have employed the on-line preplanning, or "implicit priming", paradigm developed by Meyer, 1990; 1991. The task falls into the general class of choice-response tasks. Priming and precuing of choice responses has been widely used in studying the advance planning of skilled action. For example, Rosenbaum, 1980 used precuing to control the amount of

preparation in arm movement. He manipulated the uncertainty in the specification of arm direction and extent, and observed that as more information was available to allow preparation, movement initiation time decreased. The implicit priming task differs from precuing in that no explicit cues are given in advance but the cue is implicit in the response set. However, the logic is the same in that both implicit priming and precuing allow for preparation of the action. In the implicit-priming paradigm, speakers first have to learn a set of prompt-response pairs and then have to produce the appropriate response when one of the prompts is shown. The big advantage of using paired associates compared to picture naming is that the responses do not have to refer to depictable entities, which gives more freedom in the selection of experimental materials. Roelofs, 1999 showed that implicit priming using paired associates and picture naming gives equivalent results.

An implicit-priming experiment consists of a number of alternating learning and test phases. Before each block of test trials, participants learn a small set of prompt-response pairs such as *street - bypath, rule - bylaw, show - byplay*. During the subsequent test phase, the speakers are shown per trial one of the first words of the pairs, called the prompts (e.g., *rule*), and they have to produce the corresponding second word of the pair as fast as possible without making mistakes (e.g., when presented with *rule* they have to produce *bylaw*). The order of prompts across trials is random. The production latency, the interval between prompt onset on the computer screen and speech onset, is the main dependent variable. There are homogeneous and heterogeneous response sets. In a homogeneous set, the response words share part of their form and in a heterogeneous set they have nothing in common. For example, the responses share the first syllable (*bypath, bylaw, byplay*) or they are unrelated in form (*bypath, misprint, doorstep*). Heterogeneous sets are created by regrouping pairs from different homogeneous sets. Therefore, each word pair is tested both under the homogeneous and the heterogeneous condition, and all uncontrolled item effects are kept constant across these conditions. Each participant is tested on all sets.

In testing sets with monomorphemic words like *bible*, Meyer, 1990; 1991 observed a preparation effect: The production latencies of the words combined in homogeneous sets where they share initial segments (e.g., /bar/) were smaller than those of the same words combined in heterogeneous sets. The form overlap between the responses in a homogeneous set apparently allows speakers to preplan part of the responses before the beginning of a trial, whereas such preplanning is not possible in heterogeneous sets. Subsequent studies have tested whether the size of the preparation effect depends on the morphological status of

the shared string of segments. WEAVER++ predicts that strings of segments that constitute morphemes should yield larger preparation effects than strings of segments that are not morphemes. For example, the pre-planning effect should be larger for the syllable /bai/ in response sets including complex words like *bypath* than for /bai/ in sets including simple words like *bible*. For monomorphemic words like *bible* consisting of the single morpheme <bible>, sharing the first syllable /bai/ allows for phonological preparation only. In contrast, for polymorphemic words like *bypath* consisting of the morphemes <by> and <path>, not only phonological but also additional morphological preparation should be possible, hence the prediction of a much larger preparation effect. The experimental outcomes confirmed the predictions by WEAVER++. In producing disyllabic simple and complex nouns, a larger facilitatory effect was obtained when a shared initial syllable constituted a morpheme than when it did not (Roelofs, 1996c).

WEAVER++ predicts an effect of morpheme frequency. High-frequency morphemes are retrieved faster from memory than morphemes of low frequency, so the benefit from preparation should be larger for low-frequency morphemes than for high-frequency ones. This prediction has been empirically confirmed. For example, in producing compounds in Dutch (Roelofs, 1996b), the facilitatory effect was larger for response sets sharing a low-frequency morpheme like <bloem> (*flower*) — as in *bloemkool* (*cauliflower*) — than for response sets sharing a high-frequency morpheme like <bloed> (*blood*) — as in *bloedspoor* (*trace of blood*). The compounds were matched for word frequency. Also, in producing particle verbs (Roelofs, 1998), the facilitatory effect was larger, for example, for *veeg* (low frequency) in “veeg op” (“clean up”) than for *geef* (high frequency) in “geef op” (“give up”).

The outcomes of further experiments supported WEAVER++’s claim that forms of morphologically complex words are assembled in a rightward fashion rather than, for example, in parallel (e.g., Stemberger, 1985). In producing complex nouns, no facilitation was obtained for shared noninitial morphemes. For example, no effect was obtained for <rol> in *bijrol* (*supporting role*), *koprol* (*somersault*), *deegrol* (*dough roll*). And in producing prefixed verbs, a facilitatory effect was obtained for the prefix but not for the noninitial base. For example, a facilitatory effect was obtained for the prefix <be> of *behalen* (*to obtain*), but not for the base <halen>.

According to WEAVER++, morphological complexity can play a role in form planning without having a synchronic semantic motivation. Indeed, Roelofs and Baayen, 2001 obtained the effect of morpheme preparation for semantically opaque complex words of Dutch like *bijval* (<bij><val>

applause). In producing simple (*bijbel* <bijbel>) and complex nouns (*bijrol* <bij><rol> and *bijval* <bij><val>), a larger preparation effect was obtained when a shared initial syllable constituted a morpheme than when it did not, replicating Roelofs, 1996c. Importantly, the size of the morphemic effect was identical for semantically transparent complex words (*bijrol*) and opaque complex words (*bijval*), as shown in Figure 7.4. This suggests that morphemes are present in the memory representations of opaque complex words, contrary to what intuition as well as semantic analysis would suggest. The figure also shows the simulation results. Again, there is no statistical difference between model and real data. These findings support WEAVER++'s assumption that form planning is word-based. That is, no attempt is made to derive the morphemes of a word given its meaning. Instead, the morphemes of the stem of derivationally complex words and compounds are stored with the words in memory.

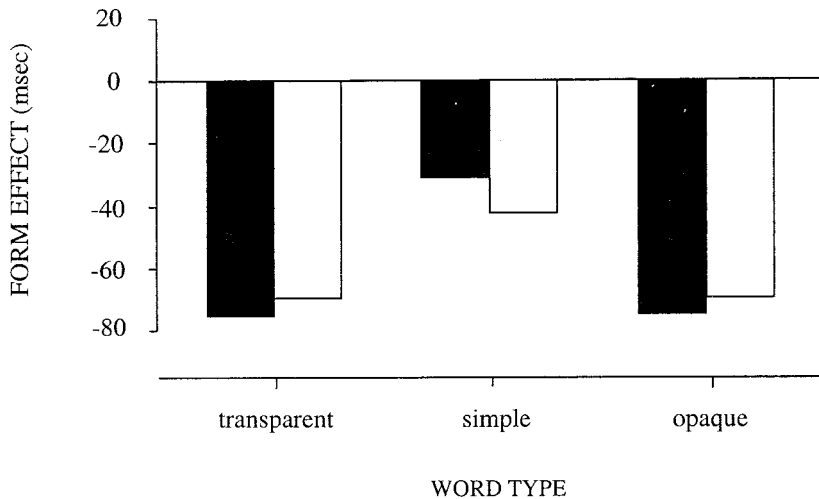


Figure 7.4. Form preparation effect for transparent complex, simple, and opaque complex words: ■ = empirical data (Roelofs and Baayen, 2001) and □ = predictions by WEAVER++.

The studies discussed so far concerned the encoding of complex stems. But how is inflectional encoding achieved? The morphological encoder in WEAVER++ selects the morpheme nodes that are linked to a selected lemma and its morphosyntactic parameters. A lemma and its diacritics may be mapped onto a simple or complex stem and its affixes (e.g., *evade* + “past” is mapped onto the morphemes <evade> and <ed>) or onto an irregular form (e.g., *eat* + “past” is mapped onto the morpheme <ate>).

For regular verbs such as *evade*, linking the stem <evade> to the lemma node of *evade* and the affix <ed> to the diacritic “past” will yield the right result (note that the latter implements a “rule” in that <ed> is selected for a non-progressive form when a lemma’s tense parameter is set to “past”, independent of the lemma involved). However, for irregular verbs such as *eat*, the diacritic “past” must be linked to the allomorph <ate> and selection of <ed> must be blocked (cf. Pinker, 1994).

Whether inflectional encoding is achieved by retrieving all forms from storage (the “single route” view) or by retrieving irregular forms and computing regular ones (the “dual route” view) is a hotly debated issue (for reviews, see Plunkett, 1995; Pinker, 1994). Evidence for two routes comes from dissociations between performance on regular and irregular forms (see e.g., Pinker, 1994). In *WEAVER++* a distinction is made between regular and irregular forms but *both* are produced via an associative network. So the model seems to embrace both sides of the debate. Evidence from the preparation paradigm supports the model. Janssen et al., 2001 observed for Dutch that the effect from preparing a word-initial syllable is much reduced when one of the words in the response set is an inflected verb while the other words are nouns. For example, the effect from sharing the syllable *wa* [wa] in a set including the noun *water* [wa.tər] and the inflected verb *waadde* ([wa.də], the past tense of *waden*, English *wade*) was much less than the effect from sharing *wa* in a set including the nouns *water* and *wapen* (*weapon*). The second syllable /də/ is the past tense suffix of the inflected form *waadde*, whereas the syllable /tər/ has no morphological status in the noun *water*. The reduction of the preparation effect was not observed when the inflected verb in the set was replaced by a homophonous noun (*wade* [wa.də], *shroud*). This suggests that the reduction was not due to the phonological form of the inflected verb itself. There was also no reduction when the past tense form of a strong verb was the odd one out in a set. For example, *kreeg* (past tense of *krijgen*, English *receive*) in a set with *kruin* (*tonsure*) and *krent* (*currant*) did not diminish the preparation effect. This suggests that the reduction is not due to a word class difference (i.e., verb versus noun). Instead, the reduction of the preparation effect must be due to the regular inflectional process itself (see Janssen, 1999; Janssen et al., 2001, for an extensive discussion).

7. Computing phonological structures

WEAVER++ assigns a specific role to metrical structures in syllabification. For words like the trochee *table*, metrical structures are computed on-line, but for words like the iamb *cigar*, metrical structures are stored.

The stored metrical structures specify the stress pattern across syllables, but not the precise CV structure of the syllables as models such as that of Dell, 1988 do. The prosodification process in WEAVER++ associates segments to the syllable nodes within the metrical structure for “exception” words (e.g., *cigar*) or constructs syllable and metrical structures based on segmental information (e.g., for *table*). These claims have been tested in a series of form preparation experiments. I refer to Levelt et al., 1999b; Roelofs, 1997b, and Roelofs and Meyer, 1998 for reviews. In homogeneous sets, the responses shared a number of word-initial segments, whereas in heterogeneous sets they did not. As we saw, earlier research has shown that sharing initial segments reduces production latencies (Meyer, 1990; 1991; Roelofs, 1996c). The crucial manipulation for the current experiments was whether the metrical structure of the words in a set was constant or variable. In metrically constant sets, all responses had the same metrical structure, whereas in metrically variable sets they did not.

A first series of experiments tested (in Dutch) the production of polysyllabic words that do not have main stress on the first stressable syllable such as English *cigar*. According to the model, the metrical structures of these words are not computed but stored in memory. If the responses in a set have different metrical structures, segment-to-frame association cannot take place before the beginning of a trial, and no preparation effect should be obtained. This prediction was tested by comparing the preparation effect for response sets with a constant number of syllables such as {*manier* (*manner*), *matras* (*mattress*), *makreel* (*mackerel*)} to that for sets having a variable number of syllables such as {*majoor* (*major*), *materie* (*matter*), *malaria* (*malaria*)} (respectively, 2, 3, and 4 syllables). In the example, the responses share the first syllable *ma*. Word stress was always on the second syllable. As predicted, a preparation effect was obtained for the metrically constant sets but not for the variable sets. The same predictions were also tested by comparing the preparation effect for response sets with a constant stress pattern such as {*marine* (*navy*), *materie* (*matter*), *malaise* (*depression*), *madonna* (*madonna*)} (all responses having stress on the second syllable) to that for sets having a variable stress pattern such as {*marine* (*navy*), *materie* (*matter*), *manuscript* (*manuscript*), *madelief* (*daisy*)} (first two responses having stress on the second syllable and the last two responses having stress on the third syllable). All response words were trisyllabic. Again, as predicted, a preparation effect was obtained for the constant sets but not for the variable sets. We also tested whether the size of the preparation effect was affected by the constancy versus variability of the CV structure of the response words. We compared the effect of segmental

overlap for response sets having a constant CV structure such as {*bres* (*breach*), *bril* (*glasses*), *brok* (*piece*), *brug* (*bridge*)} (responses all CCVC) to that for sets having a variable CV structure such as {*brij* (*porridge*), *brief* (*letter*), *bron* (*source*), *brand* (*fire*)} (responses respectively, CCVV, CCVVC, CCVC, CCVCC). In the example, the responses share the onset cluster *br*. Facilitation from segmental overlap was obtained for both the constant and the variable sets. The size of the preparation effect was the same for both types of set. These results suggest that the stress pattern across syllables but not the exact CV structure is stored, supporting WEAVER++.

A second series of experiments (again in Dutch) tested the production of monosyllabic words and polysyllabic words whose main stress is on the first syllable like English *table*. According to the model, the metrical structures of these words are computed on-line on the basis of retrieved segments. Consequently, preparation of initial segments should now be possible for both metrically constant and variable sets. This prediction was tested by comparing the preparation effect for response sets with a constant number of syllables such as {*borstel* (*brush*), *botsing* (*crash*), *bochel* (*hump*), *bonje* (*fight*)} (all disyllables stressed on the first syllable) to that for sets having a variable number of syllables such as {*borstel*, *botsing*, *bok* (*goat*), *bom* (*bomb*)} (two disyllables stressed on the first syllable and two monosyllables, respectively). In the example, the responses share the onset and nucleus *bo*. As predicted, a preparation effect was obtained for both the constant and the variable sets, and the size of the effect did not differ between the two types of set. The same result is predicted for varying the number of syllables of polysyllabic words with an unstressable first syllable (i.e., words with a schwa as the first vowel) and stress on the second syllable. This was tested by comparing the preparation effect for such words in response sets with a constant number of syllables such as {*gebit* (*teeth*), *gezin* (*family*), *getal* (*number*), *gewei* (*antlers*)} (all disyllables having stress on the second syllable) to that for sets having a variable number of syllables such as {*geraamte* (*skeleton*), *getuige* (*witness*), *gebit*, *gezin*} (two disyllables stressed on the second syllable and two trisyllables stressed on the second syllable, respectively). As predicted, the same preparation effect was obtained for the constant and the variable sets.

In summary, preparation of word initial segments for words with irregular stress depends on all words in a set having the same stress pattern across syllables but not on having the same CV structure. For words with regular stress, sharing the metrical structure is not necessary for preparation. These findings support WEAVER++.

According to WEAVER++ response preparation critically depends on shared segments rather than shared articulatory movements or phonological features. In support of this, it has been shown empirically that preparation requires that the responses share their initial segments fully and that sharing features only does not allow for preparation (Roelofs, 1999). For example, the initial segment of the words *boat*, *bird*, and *boy* is the same, whereas the initial segment of the words *paint*, *boat*, and *bird* is the same except for one feature, namely voicing (i.e., /b/ is voiced, /p/ is voiceless). However, a preparation effect was only obtained for the sets with full segment overlap, but no effect was observed at all (not even a reduction of latencies) for the sets with feature overlap. The same result has been obtained in comparing sets of disyllabic words sharing the first syllable fully (e.g., *te* in *tennis*, *terrace*, *teddy*) and sets of disyllabic words whose first syllable differs in one feature only (e.g., *devil*, *tennis*, *teddy*). Although syllables were shared except for one feature in the first segment, this feature difference completely blocked preparation. Also, the same results were obtained when words were produced in response to pictured objects, and when place of articulation rather than voicing was manipulated (e.g., /m/ versus /n/). The special status of identity suggests that segments are planning units ("chunks") independent of their features. The findings support segmental models such as WEAVER++, but they argue against models without an explicit representation of segments, such as the model proposed by Dell et al., 1993.

8. Phonetic access and underspecification

After phonological structures have been computed, the phonetic encoder in WEAVER++ accesses stored motor programs for the phonological syllables or, if no such program is stored, computes motor programs from scratch. Phonetic encoding involves mapping an abstract phonological representation of the word onto a concrete feature specification or specification of its articulatory gestures.

As with the access of lemma information and morphemes, the retrieval of syllable programs from storage should depend on frequency of usage. Levelt and Wheeldon, 1994 indeed observed an effect of syllable frequency that is independent of the word-frequency effect discussed earlier. Furthermore, in producing disyllabic words, the syllable-frequency effect was confined to the second syllable. By assuming that a production rule for a syllable program node is applied faster for a high-frequency syllable than for a low-frequency one, the finding of an independent effect of syllable frequency is readily explained by WEAVER++. Further-

more, during the encoding of disyllables, the encoding of the first syllable will have a head-start as a result of the left-to-right prosodification of segments. Thus, according to the model, the second syllable typically sets the pace of the encoding of the whole word form. Therefore, one expects the syllable-frequency effect to be confined to the second syllable, as is observed by Levelt and Wheeldon. However, the syllable frequency effects in the experiments were very small. Furthermore, in some experiments, syllable and segment frequency were correlated. In recent experiments that controlled for a number of possible confounds, effects of syllable and segment frequency were not obtained (Levelt et al., 1999b). Clearly, the speed of accessing motor programs for syllables does not depend very much, if at all, on their frequency. The validity of the syllabary assumption needs to be tested further.

Underspecification concerns the issue of whether certain features are stored with the underlying segments of morphemes in memory. A number of different views have been advanced (e.g., Archangeli, 1988; Kenstowicz, 1994, for review). In one view, only contrastive features are specified underlyingly (e.g., Halle, 1959; Steriade, 1987). For example, voicing is not contrastive among the sonorant segments of Dutch, whereas it is distinctive among obstruents. Then, voicing would not be specified with sonorant segments in memory. The second, stronger view, is based on the phenomenon that, in certain phonological contexts, contrastive features have marked (i.e., "unexpected") and unmarked (i.e., "expected") values (e.g., Archangeli, 1988; Kiparsky, 1982; Pulleyblank, 1983). On this account, only the marked values are stored and the unmarked values are assigned by rule. For example, in Dutch, the segment following /s/ in syllable onset position is never voiced (e.g., /sp/ occurs, but /sb/ does not). Consequently, voicing does not have to be specified for such segments in these contexts, because the feature value (unvoiced) can be assigned by a rule. In a third view, the mental lexicon contains archiphonemes such as the segment /P/ standing for either /b/ or /p/ (Levelt, 1989). The difference between the second and the third view is that archiphonemes can take both marked as well as unmarked values, whereas segments not specified for unmarked feature values can take the unmarked value of a feature only (their specified counterparts contain the marked value already).

Whereas the justification of underspecification in linguistic theory may be evident (i.e., eliminate as much redundancy in the linguistic representations as possible and capture the redundancy in a rule), the significance of underspecification for a model of speech production is much less clear. Archiphonemes and the elimination of noncontrastive or unmarked features place different demands on the number of mem-

ory representations needed to perform the appropriate computations. Whereas the view of unspecified noncontrastive or unmarked features seems to be fully compatible with WEAVER++ as it stands, the view of archiphonemes requires extra representations. For example, in WEAVER++ elimination of noncontrastive or unmarked features may be achieved by having segment nodes point to their contrastive or marked features only. The noncontrastive or unmarked features may then be filled in by rule or by retrieving motor programs from the syllabary. Archiphonemes (as conceived by Levelt, 1989), however, ask for something extra. Consider the memory representation of the Dutch words *speer* (*spear*), *peer* (*pear*), and *beer* (*bear*). Connecting the morpheme nodes <speer>, <peer>, and <beer> to the archiphoneme /P/ will not work, because although the feature value for voicing (here, voiceless) can be assigned by rule to /P/ in case of <speer>, it has to be specified for <peer> and <beer> in order to make explicit the distinction between *peer* and *beer*. Thus, memory has to contain segments such as /b/ and /p/ in addition to an archiphoneme such as /P/ (cf. Spencer, 1991). This raises the question of what is gained by archiphonemes. Why not connecting <speer> to /p/? What is gained by having both segments and their archiphonemes in the network? In short, for a model of speech production, the functional significance of archiphonemes is not so clear.

Of course, the ultimate judgement about underspecification should be based on empirical evidence. Although some evidence from speech errors suggests that there may be archiphonemes involved in speech production (e.g., Stemberger, 1991), I do not think we have the final answer here. Stemberger observed errors such as "...beer binnen" ('... bear inside'), where "...speer binnen" was intended. The key observation here is that, after the /s/ has got lost, the /p/ of <speer> is realized as /b/. The error might be explained (as Stemberger does) by assuming that <speer> connects to /P/ in memory. In the production of *speer*, the /P/ surfaces as /p/, because it is preceded by the /s/. However, without the context of /s/, the phonotactic rule fails to apply. Then, the underlying /P/ is free to surface as either /p/ or /b/. This might explain why, after the /s/ got lost in the error, /b/ is produced instead of /p/. WEAVER++ can only accommodate this finding if archiphonemes are included in the model. Then, when the /s/ gets lost, the syllable programs [ber] and [per] would fit the phonological syllable (Per)_σ. So, there might be a race between [ber] and [per], with the chances of winning being equal. The error *beer* for *speer* might then occur.

9. Conclusions

I have argued that word production can be characterized by a storage versus computation trade-off. The planning of a word calls forth some mixture of retrieval from storage and computation. That mixture not only varies over the ensemble of planning stages but also within stages. I have discussed the distribution of trade-offs in the context of a concrete model of lexical access in spoken word production, the WEAVER++ model. The model falls into the class of hybrid models in that it implements lexical knowledge symbolically as rules and facts but also has an activation process that determines which facts and rules get deployed for particular words. WEAVER++ implements the claim that lemmas are retrieved from stored non-decomposed representations of lexical concepts. Furthermore, morphological and phonological structures are computed on-line, typically followed by the retrieval of stored motor programs for the computed phonological syllables. As concerns phonological feature underspecification, WEAVER++ can handle several theoretical possibilities, so for the model, this is more of an empirical issue rather than a theoretical one.

References

- Archangeli, D. (1988). Aspects of underspecification theory. *Phonology*, 5:183–207.
- Aronoff, M. (1994). *Morphology by itself*. Cambridge, MA: MIT Press.
- Bierwisch, M. and Schreuder, R. (1992). From concepts to lexical items. *Cognition*, 42:23–60.
- Booij, G. E. (1995). *The phonology of Dutch*. Oxford: Clarendon Press.
- Bowers, J. S. (1999). Grossberg and colleagues solved the hyperonym problem over a decade ago. *Behavioral and Brain Sciences*, 22:38–39.
- Browman, C. P. and Goldstein, L. (1986). Towards an articulatory phonology. *Phonology Yearbook*, 3:219–252.
- Caramazza, A. (1997). How many levels of processing are there in lexical access? *Cognitive Neuropsychology*, 14:177–208.
- Collins, A. M. and Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological Review*, 82:407–428.
- Cutting, C. J. and Ferreira, V. S. (1999). Semantic and phonological information flow in the production lexicon. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25:318–344.
- Dell, G. S. (1986). A spreading-activation theory of retrieval in sentence production. *Psychological Review*, 93:283–321.

- Dell, G. S. (1988). The retrieval of phonological forms in production: Tests of predictions from a connectionist model. *Journal of Memory and Language*, 27:124-142.
- Dell, G. S., Juliano, C., and Govindjee, A. (1993). Structure and content in language production: A theory of frame constraints in phonological speech errors. *Cognitive Science*, 17:149-195.
- Dell, G. S. and O'Seaghdha, P. G. (1992). Stages of lexical access in language production. *Cognition*, 42:287-314.
- Dennett, D. C. (1991). *Consciousness explained*. London: Penguin Books.
- Fodor, J. A., Garrett, M. F., Walker, E. C. T., and Parkes, C. H. (1980). Against definitions. *Cognition*, 8:263-367.
- Garrett, M. F. (1975). The analysis of sentence production. In Bower, G. H., editor, *The psychology of learning and motivation*. New York: Academic Press.
- Garrett, M. F. (1980). Levels of processing in sentence production. In Butterworth, B., editor, *Language production (Vol 1). Speech and talk*. London: Academic Press.
- Garrett, M. F. (1988). Processes in language production. In Newmeyer, F. J., editor, *Linguistics: The Cambridge survey*, volume 3. Cambridge, MA: Harvard University Press.
- Glaser, W. R. and Döngelhoff, F.-J. (1984). The time course of picture-word interference. *Journal of Experimental Psychology: Human Perception and Performance*, 10:640-654.
- Goldsmith, J. (1990). *Autosegmental and metrical phonology*. Cambridge, MA: Basil Blackwell.
- Halle, M. (1959). *The sound pattern of Russian*. The Hague: Mouton.
- Hankamer, J. (1989). Morphological parsing and the lexicon. In Marslen-Wilson, W., editor, *Lexical representation and process*. Cambridge, MA: MIT Press.
- Jackendoff, R. (1990). *Semantic structures*. Cambridge, MA: MIT Press.
- Janssen, D. (1999). Producing past and plural inflections. Unpublished doctoral dissertation, University of Nijmegen, The Netherlands.
- Janssen, D., Roelofs, A., and Levelt, W. J. M. (2001). The production of inflected verbs. *Language and Cognitive Processes*. In press.
- Jescheniak, J.-D. and Levelt, W. J. M. (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.
- Kenstowicz, M. (1994). *Phonology in generative grammar*. Oxford: Blackwell.
- Kiparsky, P. (1982). Lexical morphology and phonology. In Yang, I. S., editor, *Linguistics in the morning calm*. Seoul: Hanshin.

- Koskenniemi, K. (1984). A general computational model for word-form recognition and production. In *Proceedings of the 10th International Conference on Computational Linguistics*.
- Levelt, W. J. M. (1989). *Speaking: From intention to articulation*. Cambridge, MA: MIT Press.
- Levelt, W. J. M. (1992). Accessing words in speech production: Stages, processes and representations. *Cognition*, 42:1–22.
- Levelt, W. J. M., Roelofs, A., and Meyer, A. S. (1999a). Multiple perspectives on word production. *Behavioral and Brain Sciences*, 22:61–75.
- Levelt, W. J. M., Roelofs, A., and Meyer, A. S. (1999b). A theory of lexical access in speech production. *Behavioral and Brain Sciences*, 22:1–38.
- Levelt, W. J. M. and Wheeldon, L. (1994). Do speakers have access to a mental syllabary? *Cognition*, 50:239–269.
- Meyer, A. S. (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, A. S. (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, A. S. (1996). Lexical access in phrase and sentence production: Results from picture-word interference experiments. *Journal of Memory and Language*, 35:477–496.
- Meyer, A. S. and Schriefers, H. (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.
- Miller, G. A. and Johnson-Laird, P. N. (1976). *Language and perception*. Cambridge, MA: Harvard University Press.
- Pinker, S. (1994). *The language instinct: The new science of language and mind*. London: The Penguin Press.
- Plunkett, K. (1995). Connectionist approaches to language acquisition. In Fletcher, P. and MacWhinney, B., editors, *The handbook of child language*. Oxford: Blackwell.
- Pulleyblank, D. (1983). *Tone in lexical phonology*. PhD thesis, MIT.
- Roelofs, A. (1992). A spreading-activation theory of lemma retrieval in speaking. *Cognition*, 42:107–142.
- Roelofs, A. (1993). Testing a non-decompositional theory of lemma retrieval in speaking: Retrieval of verbs. *Cognition*, 47:59–87.
- Roelofs, A. (1994). On-line versus off-line priming of word-form encoding in spoken word production. In Ram, A. and Eiselt, K., editors, *Pro-*

ceedings of the Sixteenth Annual Conference of the Cognitive Science Society. Hillsdale, NJ: LEA.

- Roelofs, A. (1996a). Computational models of lemma retrieval. In Dijkstra, T. and De Smedt, K., editors, *Computational psycholinguistics: AI and connectionist models of human language processing*. London: Taylor & Francis.
- Roelofs, A. (1996b). Morpheme frequency in speech production: Testing WEAVER. In Booij, G. E. and van Marle, J., editors, *Yearbook of Morphology 1996*. Dordrecht: Kluwer Academic Publishers.
- Roelofs, A. (1996c). Serial order in planning the production of successive morphemes of a word. *Journal of Memory and Language*, 35:854–876.
- Roelofs, A. (1997a). A case for nondecomposition in conceptually driven word retrieval. *Journal of Psycholinguistic Research*, 26:33–67.
- Roelofs, A. (1997b). Syllabification in speech production: Evaluation of WEAVER. *Language and Cognitive Processes*, 12:657–693.
- Roelofs, A. (1997c). The WEAVER model of word-form encoding in speech production. *Cognition*, 64:249–284.
- Roelofs, A. (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, A. (1999). Phonological segments and features as planning units in speech production. *Language and Cognitive Processes*, 14:173–200.
- Roelofs, A. and Baayen, H. (2001). Morphology by itself in planning the production of spoken words. *Psychonomic Bulletin and Review*. In press.
- Roelofs, A., Meyer, A., and Levelt, W. J. M. (1996). Interaction between semantic and orthographic factors in conceptually driven naming: Comment on Starreveld and La Heij (1995). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22:246–251.
- Roelofs, A. and Meyer, A. S. (1998). Metrical structure in planning the production of spoken words. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24:922–939.
- Roelofs, A., Meyer, A. S., and Levelt, W. J. M. (1998). A case for the lemma-lexeme distinction in models of speaking: Comment on Caramazza and Miozzo (1997). *Cognition*, 69:219–230.
- Rosenbaum, D. A. (1980). Human motor initiation: Specification of arm direction and extent. *Journal of Experimental Psychology: General*, 169:444–474.
- Sandra, D. (1994). The morphology of the mental lexicon: Internal word structure viewed from a psycholinguistic perspective. *Language and Cognitive Processes*, 9:227–269.

- Shattuck-Hufnagel, S. (1979). Speech errors as evidence for a serial-order mechanism in sentence production. In Cooper, W. E. and Walker, E. C. T., editors, *Sentence processing: Psycholinguistic studies presented to Merrill Garrett*. Hillsdale, NJ: Erlbaum.
- Spencer, A. (1991). *Morphological theory*. Cambridge, MA: Blackwell.
- Starreveld, P. A. and La Heij, W. (1996). Time-course analysis of semantic and orthographic context effects in picture naming. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22:896–918.
- Stemberger, J.-P. (1985). An interactive activation model of language production. In Ellis, A. W., editor, *Progress in the psychology of language (Vol. 1)*. London: LEA.
- Stemberger, J.-P. (1991). Radical underspecification in language production. *Phonology*, 8:73–112.
- Steriade, D. (1987). Redundant values. In Bosch, A., Need, B., and Schiller, E., editors, *Papers from the parasession on autosegmental and metrical phonology*. Chicago: Chicago Linguistic Society.
- Zorzi, M. and Vigliocco, G. (1999). Compositional semantics and the lemma dilemma. *Behavioral and Brain Sciences*, 22:60–61.