An Online Tool for L2 Vocabulary Collection and Learning

By

Paula Molloy

A thesis submitted in partial fulfilment of the requirements

for the Degree of B.A.

2005

Trinity College Dublin.

Supervisor: Dr. Breffni O'Rourke.
Declarations

I declare that this final year project, in whole or in part, has not been submitted as an exercise for a degree in any other university.

I further declare that, except where reference is given, the material contained in this thesis is my own work.

____________________
Paula Molloy
Abstract

An Online Tool for L2 Vocabulary Collection and Learning

In this project, I will describe current pedagogical theories on the processes of second language vocabulary acquisition, and on the techniques and strategies which can be most effectively employed by students when learning second language vocabulary. I will describe several currently existing software tools for the learning of second language vocabulary, and discuss the design criteria which should be implemented when designing such tools. I will then present the Lexical Organizer program in its original state, and go on to describe the improvements and additional functionalities which I shall implement in the course of the project. Finally, I will suggest some directions for future work using the Lexical Organizer program.
Acknowledgements

I am most grateful to Dr. Breffni O’Rourke, Centre for Language and Computer Studies, who supervised this work and designed the Lexical Organizer program. His encouragement and guidance have been very important to me.

I am also indebted to Alexandre Borgia, who wrote the original version of the Lexical Organizer, and whose help and patience were invaluable in overcoming initial difficulties with the program.

I would like to take this opportunity to thank all those who have contributed in any way towards the preparation of this project, in particular, my family and closest friends.
## List of Figures

| Figure 1.1 | Levelt's Blueprint for the Speaker: | 5 |
| Figure 2.1 | Workspace showing hierarchy of folders: | 26 |
| Figure 2.2 | Grouping entries: | 26 |
| Figure 2.3 | Linking entries: | 26 |
| Figure 3.1 | Before copy function is used: | 34 |
| Figure 3.2 | Copy function: | 34 |
| Figure 3.3 | After copy function is used: | 34 |
| Figure 3.4 | Update all copies function: | 35 |
| Figure 3.5 | Copy and remove original function: | 35 |
| Figure 3.6 | Before copy and remove original function is used: | 35 |
| Figure 3.7 | After copy and remove original function is used: | 35 |
| Figure 3.8 | Before copy and rename: | 36 |
| Figure 3.9 | After copy and rename: | 36 |
| Figure 3.10 | Copy to same folder: | 36 |
| Figure 3.11 | Contents of original entry: | 36 |
| Figure 3.12 | Contents of copy: | 36 |
| Figure 3.13 | Test definitions function: | 40 |
| Figure 3.14 | Showing clue: | 40 |
| Figure 3.15 | Translating from L1 to L2: | 40 |
| Figure 3.16 | Showing correct answer to previous question: | 40 |
| Figure 4.1 | Suggested layout for Print Word List function: | 44 |
| Figure 4.2 | Start of Test: | 45 |
| Figure 4.3 | Test Solution: | 45 |
# Table of Contents

Declaration: II  
Abstract: III  
Acknowledgements: IV  
List of Figures: V

## Chapter 1: Vocabulary and the Foreign/Second Language Learner

1.1 The Mental Lexicon 1  
1.1.1 What is a word? 1  
1.1.2 What does it mean to know a word? 2  
1.1.3 Models of the L1 mental lexicon 3  
1.1.4 L1 mental lexicon development 6  
1.1.5 A comparison of the L1 and L2 mental lexicons 7  
1.2 Vocabulary Learning and Teaching Strategies 9  
1.2.1 What makes an L2 word difficult or easy to learn? 9  
1.2.2 Implicit Learning 11  
1.2.3 Explicit Learning 13  
1.2.4 Discovery Strategies 14  
1.2.5 Consolidation Strategies 16

## Chapter 2: Software Review

2.1 CALL and Vocabulary Learning 18  
2.1.1 Design principles 18  
2.1.2 The evolution of vocabulary instruction programs 19  
2.2 Existing Software 21  
2.2.1 Vocabulary Trainer 21  
2.2.2 Word Pilot 22  
2.3 Current Version of Lexical Organizer 25

## Chapter 3: Additional Functionality

3.1 An Overview of the Design of the Lexical Organizer 29  
3.2 Improving the Flexibility of the Lexical Organizer 30  
3.2.1 Copy Entries 31
Chapter 1
Vocabulary and the Foreign/Second Language Learner

1.1 The Mental Lexicon

1.1.1 What is a word?

Words are the vital components of all languages. Without them, linguistic communication would be impossible. Every known language consists of words, and even speakers of languages without an orthographic system have no difficulty in isolating individual words from a stream of speech (Singleton 1999). Yet it has been surprisingly difficult for linguists to reach a consensus in defining what exactly a word is. Various approaches have been taken in attempting to define the word, with varying degrees of success.

An orthographic approach which defined a word as a sequence of letters bounded by blank spaces was ineffective for unwritten languages, or those whose orthographic system did not use word-boundaries.

A phonetic approach suggested that words are identifiable in terms of how they sound in speech. However, individual words cannot easily be isolated in a speech stream. Indeed, this difficulty has contributed to language evolution, with words taking on phonemes from others with which they frequently appear in collocation. For example, the English word ‘a newt’ was originally ‘an ewt’ (Singleton, 2000).

The semantic approach defines words as being the basic units of meaning. This definition is problematic as it does not account for compound words. For example, is ‘traffic light’ one word or two? Also, morphological components, such as ‘-ed’ added to the end of a word, clearly bring semantic information of their own to a word.

To date, the most effective and uncontroversial definition of a word has been made using the grammatical approach. Here, a word is considered to be positionally mobile...
in a sentence, but internally stable – that is to say, the order of morphemes within the word is constant. However, the mobility of many grammatical words is considerably limited. For example, ‘the’ must appear before the noun which it modifies.

This highlights the fact that there are two different categories of words, each with different properties. These categories are:

- **lexical** words, also called full or content words. These are the words which retain semantic meaning even when they appear out of context. This class of words is open and constantly changing. New words can be invented and added to the lexicon, while other words can fall out of use and disappear.

- **grammatical** words, or empty or form words. These are the words which have little or no semantic meaning independent of the context in which they appear. They play largely grammatical roles in a sentence. This class of words is closed. It is extremely rare for a new grammatical word to be introduced in the lifetime of a speaker.

### 1.1.2 What does it mean to know a word?

The complexity of words and their role in language is proven by the difficulty involved in finding an adequate definition of what a word is. This is because every word comprises a wide range of information, and full knowledge of a word requires a command of each of these different areas.

To know a word, a speaker must know its orthographic and phonetic form (so that it can be recognised and produced in both speech and writing), its syntactic behaviour (so that its different functions can be recognised and used), its various semantic features and the contexts in which they are applicable (so that it can be understood and used correctly), the collocations in which it regularly occurs and how they impact upon its meaning, and its derivatives and inflected forms. Relevant sociolinguistic information, such as the register in which a given word is used, must also be learned.
When someone is learning a foreign or second language, it is naturally impossible for them to learn all of this information immediately. Their knowledge of a word develops with more exposure to a given word, in different contexts which reveal different aspects of its usage. As this information is added to their stored mental representation of the word, their ability to use the word correctly also develops. Initially, it is not unusual for a new word to be known only passively – that is to say, it can be understood when encountered in a text, but cannot be used spontaneously by the learner. As the understanding of the word deepens, the mental representation becomes active, and the learner can access it to use the word productively as well as receptively.

1.1.3 Models of the L1 mental lexicon

It has long been suggested that language is too complex for children to learn simply through imitation and repetition, and that humans must have an inbuilt propensity for language learning. This is the ‘poverty of the stimulus’ argument, which states that our brains are specially evolved to deal with language processing and acquisition. The mental lexicon is an important part of this innate language system.

The mental lexicon is still not fully understood, and many different theories and models have been proposed to explain its organisation. We do know that it must contain comprehensive representations of all the information described in the previous section for each word known by the speaker, and it must also contain connections between these words.

One attempt to describe the workings of the mental lexicon is Morton’s (1969) logogen model. This model attempts to explain the fact that a word will be recognised more quickly when seen in context, and that an unexpected word will be recognised more slowly. For example, readers often make errors because of presuppositions based on the surrounding context. It states that, on receiving semantic information, certain neural units or logogens are activated or ‘fired’, and the words connected to them become readily available.
Marslen-Wilson’s (1989) cohort model is also based on the idea that input activates neurones in order to make words available. In this model, a set of word-detectors analyse spoken input. All words beginning with the first sounds processed are activated, and as the analysis progresses mismatches are removed until only the correct word is left. The point at which only one word remains is called the ‘recognition point’ or ‘uniqueness point’. Semantic information is also taken into account in determining whether or not a word could be correct.

The above models are direct models of lexical processing, where individual words are activated immediately upon receiving input. More modern theories have developed indirect models, where a series of steps are necessary to locate the correct lexical entry.

One of the first of these was Forster’s (1976) search model. Here, several peripheral access files are initially available, one containing phonological information, one orthographic, and one syntactico-semantic. Depending on the type of input, the appropriate access file is searched and when a match is found, a pointer from this match activates the complete form of the word in the main lexicon. Here, a full array of information is available. This model is efficient in that it allows lexical entries to be accessed quickly from several different input types, but only needs the complete entry to be stored once. However, its complete separation of processing components seems unnecessary.

Possibly the most comprehensive model of language processing is Levelt’s (1995) blueprint, which covers not only the mental lexicon itself but also its interaction with both language comprehension and production systems. In this system, two types of knowledge are used. These are:

- **Declarative knowledge** or world knowledge, including general information about the world, about specific situations and about how discourse works. Also included under this heading is lexical knowledge. The lexicon contains lemmas, holding the semantic and syntactic information for a word as well as a pointer towards the relevant lexeme, which give precise morphological information about the word to enable phonetic encoding and decoding.
- **Procedural knowledge** or knowledge of the steps to be taken in order to create or decode a message. The procedural components of the blueprint include the conceptualiser (which generates and monitors messages), the formulator (which takes the pre-verbal message produced by the conceptualiser and attaches syntactic and phonological components), the articulator (which takes the formulator’s phonetic plan and turns it into overt speech), the audition component (which analyses the incoming speech sound into phonetic segments) and finally the parser (which decodes the phonetic representation into semantic and syntactic units).

![Levelt's Blueprint for the Speaker](image)

The conceptualiser uses available encyclopaedic and situational knowledge to construct a pre-verbal message. The formulator consults the lemmas in the lexicon to find the closest semantic match for this message. It then takes the syntactic
information from this lemma to create a structure for the final message, before consulting the related lexeme, where it obtains the correct phonological encoding for the words. The articulator then takes the completed plan and executes it as speech. This speech is monitored by the auditory side of the system, which carries out similar operations in reverse order.

The major flaw with Levelt’s blueprint is that it simply describes the lexicon in terms of how it is used by the production and comprehension components of the language processing system. It fails to show how new elements can be added to the lexicon, or how the information stored there can be changed. We know that the mental lexicon is more dynamic than that which is portrayed here.

1.1.4 L1 mental lexicon development

Although children may be born with an innate propensity for language learning, they still have no vocabulary knowledge with which to fill their mental lexicon. This learning process is facilitated by the manner in which adults speak to young children – they talk slowly and clearly, make much use of repetition, and focus solely on the ‘here and now’ and ostensive definition (pointing at the object being referred to), so that it is easier for the child to isolate a word in the speech stream and connect it with its meaning.

Even newborn babies can discriminate between particular speech sounds, such as voiced and voiceless consonants. Before they ever produce their first words, babies are developing concepts of the world around them (Singleton, 2000) – for example, it has been observed that they look in the same direction as their care-giver (this is called ‘shared reference’).

From the age of one month onwards, babies start producing speech sounds. These develop from very basic *cooing* and *babbling* noises, in which vowel and consonant-like sounds are strung together in a repetitive manner. This process develops their phonological knowledge, and they lose the ability to distinguish speech sounds that are not phonemically important in their native tongue. Babies also develop discourse knowledge at an early stage, when they engage in turn-taking (imitating conversation)
with their primary carer. Gradually, in a process called babbling drift, the child’s productions become more like language, with the introduction of prosody. One-word utterances appear at about 1 year old, followed by two-word utterances six months later.

From the point of view of the construction of the mental lexicon, it is this period which is most interesting. The rate of vocabulary acquisition is very slow up to the point where 30 words are known. During this phase, the child is figuring out what words are, how they can be used, and what entities and actions each word relates to. The child’s use of known vocabulary is still inconsistent. Words are used in a fluid way, with over- and under-extension both common, showing that meanings are not yet firmly attached to individual words, and that word categories are not yet concrete. Initially, only easily imageable words with high utility are learned.

At about 2 years of age, there is a ‘vocabulary explosion’. Suddenly, the rate of vocabulary acquisition shoots up. New words can be learnt after a single exposure. However, nouns are still the most common items to be learnt, as they are the most concrete and easily imageable.

It is at this point that the mental lexicon undergoes a considerable change. It must be organised in a structured fashion to enable the child to store and access all of this new vocabulary quickly. Part-of-speech categories are introduced and a hierarchical structure is established. (Meara, 2002). Word-association tests show that, in the early stages of language learning, children make syntagmatic, or thematic, links between words. After their mental lexicon is revised, they make paradigmatic links based on the semantic and syntactic connections they have formed between words.

1.1.5 A comparison of the L1 and L2 mental lexicons

While the process of building a native language mental lexicon must start from very basic beginnings, the adult L2 learner can begin at a much higher level, having already mastered basic language concepts. However, many similarities can be observed, and the basic process is still that of isolating individual words from a given input and connecting the word form with its meaning.
A new phonological system must be constructed for the L2. By now, the learner has internalised the principle of phonemic differences, but still must determine which sounds act as phonemes in the new language. Where these correspond with L1 phonemes, the learner can simply re-use this knowledge. However, cross-linguistic interference can impede a learner from making necessary distinctions in the L2 which do not exist in the L1.

By the same token, the mature L2 learner does not have the same difficulties in developing concepts. Many word meanings, for both concrete and abstract terms, will be re-usable in the new language, or will only need slight adjustments. Occasionally, problems will arise from the transfer of inappropriate semantic information from a word to its L2 translation. Also, it is possible that a new concept which is not articulated by the native language will have to be mastered. But in general, existing knowledge facilitates the acquisition of new vocabulary.

Word association tests carried out on L2 learners as part of the Birkbeck Vocabulary Project seem to suggest that the L1 and L2 mental lexicons develop along the same lines. As we saw in the previous section, the finalised L1 mental lexicon uses semantic and syntactic features to link words, and mature native speakers of a language tend to make paradigmatic associations between words. In contrast, L2 learners are more inclined to make syntagmatic associations, linking phonologically similar words rather than semantically related ones (Carter, 1998). This would seem to imply that phonological connections take precedence in the L2 mental lexicon. However, on closer examination, it seems likely that the difference in fact stems from the learner's inexperience with the language. Initially, the L2 lexicon is very simple, but as the learner becomes more proficient with the language, it develops to become more like that of an adult native speaker. This theory is borne out by the fact that more advanced L2 learners are more likely to give semantically motivated responses to these tests.

This leads to the question of how integrated the L2 lexicon is with that of the L1. Some evidence exists to suggest that they must be independent to some degree. In the case of language loss due to brain damage, it is not uncommon for one language to be
recovered before another. For example, Singleton (2000) describes the case of a native speaker of Swiss German who recovered French first, then German, but never actually recovered his native language.

Despite this, the majority of the available evidence suggests that at least some integration must exist. The fact that an L2 learner’s reaction time to a word in one language is related to the frequency of its cognate in their other language shows that some information must be shared across the two language systems. Also, the fact that cognates and loan words are more easily learned than other words in a language implies that it must be possible to either transfer or share phonological and semantic information between the two lexicons.

One theory as to how this sharing of information could work is that pairs of exact translation equivalents are stored in a ‘compound’ manner, where the two lexical forms share the same meaning. Alternatively, it is possible that L2 forms which are semantically related to L1 words are stored as variants of that L1 word.

1.2 Vocabulary Learning and Teaching Strategies

1.2.1 What makes an L2 word difficult or easy to learn?

As we have seen, a large amount of information must be learned before a new L2 vocabulary item can truly become part of the L2 mental lexicon. It is reasonable to suppose that the more often an item is encountered, the more likely it is to be properly learned. Therefore the most frequently encountered words in a language could be said to be the easiest to learn. However, many intralexical factors (relating to word form and meaning) must also be considered.

As explained in Laufer (1997), one of the most important aspects in determining the learnability of a word is its pronounceability. A phonologically difficult word will be more difficult to perceive and to say, and therefore the learner will meet difficulty in trying to recall it. For example, when L1 and L2 phonological systems are different, the learner can have problems distinguishing between two L2 phonemes. As a result,
minimal pairs can be confused. Similar problems can arise when stress patterns differ between two languages.

Words in languages in which the orthographical system corresponds strongly to pronunciation are easier to learn. For example, the silent letters and other apparent irregularities in written English cause difficulties for learners.

The morphological system of a language can impact upon how hard or easy its vocabulary is to learn. If its inflectional system is complex, with irregular plural and gender forms, the workload for the learner is increased and so the vocabulary is more difficult to learn. The same applies to derivations. Here, learning can be made easier if a student breaks a word down into its morphemes. However, this is not always reliable – there can be irregularities in the ways in which morphemes can combine to form words.

The similarity of lexical forms, or synformity, also affects learners’ ability to learn new words. Lexical items that look or sound alike can be confused, even if their semantic meanings are very different. Problems occur when two or more lexical forms share an identical number of syllables, identical stress patterns and the same part of speech. In these cases, segments in the same syllabic position are usually confused.

Semantic features are also significant. A commonly held belief is that abstract words are harder to learn than concrete words. While this may be true in the case of L1 acquisition, most L2 learners have already mastered these abstract concepts in their native language and have no difficulty transferring them to a second language. In reality, the problem arises when a new concept is met, one that does not exist in the learner’s L1. More commonly, difficulties occur when a similar, but not identical, concept is expressed in the new language. The learner mistakenly transfers all of the semantic information from the L1 equivalent to the new vocabulary item, and as a result uses the word incorrectly in some situations.
Specificity brings problems for L2 learners, who often overgeneralise and use superordinates. They can also be unaware of register restrictions, and use inappropriate language in certain situations.

Finally, when words have multiple meanings, this can make them more difficult to learn. Learners can have difficulty in distinguishing the different meanings of a word, and in using these meanings correctly. It is very common for a learner to latch onto one meaning of a word and persist in using it, even when the context clearly shows it to be incorrect.

All of these factors are independent of word frequency or context and yet have a very real impact on how quickly learners are able to acquire new words. It is important that language teachers recognise these problem areas and help their students develop strategies to tackle them.

### 1.2.2 Implicit Learning

The vast majority of the words known by the average native speaker of a language have not been explicitly or intentionally learned in a formal setting. Rather, they have been picked up through exposure to the language, or implicitly learned. Clearly, implicit learning is a very important part of any language-learning process, but at present the process is not fully understood.

Miller (1956) posits a theory to explain this unconscious learning process. His claim is that our short-term memory automatically analyses the speech stream and abstracts patterns from it. These patterns are called *chunks*. When a particular sequence occurs regularly, the system recognises it and chunks it automatically, thus cutting down on analysis time and increasing language fluency. These chunks are generally words, but can also be morphemes or even idiomatic phrases. Several pre-formed chunks can be brought together to form a larger structure until eventually an entire sentence is constructed.

Melton (1963) further developed upon the chunking theory. He noticed that a stimulus which is frequently repeated in short term memory becomes easier to learn in
long term memory, and is therefore recognised more easily the next time it is encountered. As regards L2 vocabulary learning, this implies that a learner will recognise frequent L2 sequences more easily than rare ones.

The *power law of practice*, described by Anderson (1982), shows that initially, an L2 learner’s performance will improve with practice, but that the rate of improvement will eventually decrease. Once a learner has some experience of a new language, he will build a sizeable number of pre-formed chunks and so his performance will be faster. Eventually, he will have long-term memory representations of all the small, commonly used chunks in the language, so he will start to form larger, less frequently-seen chunks. These are less useful as they occur more rarely, and so the learning process begins to slow down.

An understanding of this chunking process can aid vocabulary learning for second language learners. One important implication of this system is that repetition of L2 words will help with retention of these words – if they have been seen often enough in the short-term memory, they will be automatically chunked and form part of the long-term memory.

It is also clear that vocabulary will be easier to learn if parts of the learner’s L1 chunking system can be transferred to the L2. This will depend on how similar the allowable arrangement of phonemes is between the two languages. If phonemes frequently occur in the same order in the two languages, many of the pre-existing L1 chunks will be applicable to the L2 words, aiding pronunciation and memorisation. By the same token, phonotactic regularity within the L2 will help the learner.

One area which often receives little attention in classroom teaching, but which is clearly facilitated by the way in which our phonetic memory works, is that of idiomatic language. Idioms and fixed expressions comprise a large part of any language and a command of them is essential to real fluency. If a learner is exposed to these phrases often enough, he will build chunks to represent them and their use will be internalised.
Even grammars can be learned incidentally, as shown by Miller & Chomsky (1958). In their experiment, subjects were shown a series of strings of letters with a hidden pattern. Without explicitly learning or knowing the rules, the subjects were able to identify strings which did not conform to the grammar.

### 1.2.3 Explicit Learning

Implicit learning is most effective when a learner is completely surrounded by input in the target language. Many encounters are usually necessary in order for a word to be learnt. For foreign language learners who have little exposure to their target language, explicit learning is a more viable option. Particularly in the early stages of language learning, where inferring word meaning from context is very difficult, most language teachers prefer to adopt more explicit teaching strategies.

These can consist of both directed implicit learning, and of more strongly explicit techniques (Ellis, 1997). In directed implicit learning, learners are given a set of guidelines in order to assist them in correctly inferring word meanings from context. They are shown how to effectively process the information provided to them in a text. This reduces the likelihood of their simply skipping over unknown words, and of ascribing incorrect meanings to words which may become fossilised if left uncorrected. Because the learner is actively involved in decoding the meanings of words, learning can become more efficient.

One such strategy was proposed by Nation (2001). Learners are advised to first identify the part of speech of the new word, and then look in the surrounding context for words modifying or interacting with this word. They should also try to identify the root of the word and any affixes contained in it. This knowledge should then be used to guess the meaning of the word. It is important that the guessed meaning is checked against some reference, to avoid incorrect guesses becoming fossilised.

While these ‘weak explicit strategies’ (Carter, 1998) are useful for advanced learners, beginners need a more prescriptive approach. One of the most popular is the use of word lists, where new L2 words are paired with an L1 equivalent. The pairing is learnt by heart. Although this approach is not favoured by modern, communicative
styles of language teaching, it has been shown to have extremely good results for beginner L2 learners. A survey of Japanese students (Schmitt, 1997) showed that learners themselves believe that word lists are a highly effective vocabulary learning method. However, the method was less popular with more mature, advanced learners.

With explicit learning, it is important to create a conscious link between the form and the meaning of the new word. The effort involved in creating this link makes the word more memorable, and focuses attention on both semantic and formal features that may have otherwise gone unnoticed. Mnemonic strategies are commonly used for this purpose.

The basis of such strategies is to take an L2 word and associate it with an L1 word which sounds like some part of it. This L1 word can be used as a ‘keyword’ (Carter, 1998) to access the translation of the target word. For example, a student learning the Bali word ‘pintu’ (meaning door) may notice that it contains the English keyword ‘pin’. The student can then create a sentence linking the two words, or a mental picture of them interacting (for example, they could visualise a door with a pin stuck in it).

This technique can be very useful in the early stages of L2 vocabulary learning, but it does have some drawbacks. It is less effective in vocabulary production than in comprehension, because the keyword is only an approximation of the L2 word. This can result in problems with spelling and pronunciation. Also, not every L2 lexical item will have a concrete, easily-pictured keyword associated with it. However, it can be a more effective approach than the simple use of word lists because it links form, meaning and structure and forces the student to consider several different aspects of the vocabulary item.

1.2.4 Discovery Strategies

Encountering new words is obviously a vital step in any attempt to increase L2 vocabulary knowledge. These new words can be learned implicitly through reading books or newspapers, or through watching television programs, in the target language,
or they can be introduced deliberately by the teacher for explicit learning purposes. In this case, it is important for the teacher to choose suitable vocabulary for his/her students.

One publication which aims to help teachers in their selection of vocabulary for learning in the early stages of the acquisition of English is West’s GSL – A General Service List of English Words (1953). It consists of a list of about 2000 words which have been chosen according to certain criteria. The words must have universality (that is, they must be recognised in all English-speaking countries), utility (they must not be restricted to one specific domain but must facilitate discussion of a wide range of subjects), and usefulness (in terms of definition value). The GSL list of words claims to cover 80% of the words in any given text. This quick return on learning is highly motivational for students.

Once the word form has been encountered, its meaning must be made clear. Unless the word is presented together with semantic information, the learner must use determination strategies to discover its meaning. Such strategies can include the inference techniques already described, in which the learner uses clues from the surrounding context to figure out meaning.

Alternatively, a learner’s L1 knowledge may prove useful in determining the meaning of target language words. For example, cognates may exist between the two languages. These are words which are descended from a common parent word, such as ‘hound’ in English and ‘Hund’ in German. Loan words which are incorporated into one language from another can also be useful. For example, Finnish has borrowed the Indo-European word ‘numero’. However, students can be reluctant to assume that a cognate will have the same meanings in the target language as it does in their native tongue.

The use of reference materials can also be instrumental in providing semantic information for a new word form. Although monolingual dictionaries provide more accurate and more detailed information, second language learners generally prefer to use bilingual dictionaries to look up new words.
Social strategies are also useful in discovering word meanings. Students can ask fellow students or teachers for a direct translation, a paraphrase or an L2 synonym of the new word. While the responses are usually quickly and easily understood, the lack of learner input into determining the word meaning can make it less memorable.

1.2.5 Consolidation Strategies

Once the new vocabulary item has been introduced and explained, the learner must ensure that it becomes a permanent part of their mental lexicon. Consolidation strategies are used to make the word easier to recognise and to recall when needed.

One way in which to ensure that a newly-learned word does not fade from memory is to use it regularly in a productive way. Through group work with other students, or interaction with native speakers, L2 learners get a chance to use their knowledge. The more often they encounter a given word in context, the more information they will learn about its collocations, syntactic restrictions and different meanings (Schmitt, 1997).

Mnemonic strategies such as the keyword technique already described also help to consolidate a learner’s knowledge of a word. By linking the new word to a memorable image, or inserting it into a sentence, it becomes easier to retrieve.

Other popular memory strategies allow semantically related words to be linked, for example in semantic grids where words with similar meanings are defined in relation to one another, so that the subtle differences in their meanings can be made clear. However, they can pose difficulties for learners. It is more normal for a student to learn the core meaning of a new word first, and only later acquire the extended senses of the word. Also, these grids are completely devoid of context with which to illustrate the differences in meanings of the words – everything is described explicitly in terms of semantic features, and these descriptions can be more complex than the vocabulary being learned. Finally, they lead to confusion, with students ascribing the features of one word to another member of the semantic grid.
Free-recall studies show that people naturally organise vocabulary into semantically-defined groups. For example, when given a list of randomly-ordered words, all the animals are recalled together, then all the place names, etc. This grouping can be used to assist vocabulary memorisation. Bellezza (1983) shows that, by arranging words in appropriate groups spatially on a page, recall is increased.

Other memory strategies also focus on deep processing of the word in order to stimulate the mind. These include close studies of spelling and pronunciation, as well as the affixes and root of a word. Analysing idioms and proverbs into their component words can make them easier to learn, by forcing the student to think about their origins and their literal meanings.

*Cognitive strategies* use mechanical means to aid vocabulary learning, rather than relying so heavily on mental processing. They include written and verbal repetition, and the use of word lists and vocabulary notebooks as study aids. These can be used for initial exposure to a word as well as facilitating progressive learning, as further information concerning a word can be added at any point.

*Metacognitive strategies* are concerned with evaluating the learning process as a whole. Students should analyse the effectiveness of their chosen vocabulary-learning strategy and be presented with alternatives. For example, research shows that it is important to regularly review new vocabulary. Russell (1979) suggests revision 10 minutes after initial exposure, then 24 hours later, 1 week later, 1 month later and finally 6 months after the word was first learned. Strategies like these will enable students to get the best possible use out of their learning time.
Chapter 2
Software Review

2.1 CALL and Vocabulary Learning

2.1.1 Design principles

We have already seen that vocabulary learning is a complex and ongoing task, involving the addition of a wide variety of linguistic information to the learner’s growing L2 mental lexicon. The learner can avail of many different learning strategies to facilitate this task, the most effective of which take a deep processing approach so that semantic and syntactic features are learned in addition to word form. Any good L2 vocabulary-learning tool should make use of this information.

L2 learners face a sizeable task when trying to build their L2 mental lexicon. The average adult native speaker of English, for example, knows 17000 base words, which have been learnt at an average rate of two or three per day over the lifespan. Even if the L2 learner were exposed to the same amount and quality of input as the native speaker, amassing a similarly large mental lexicon would clearly be a long and slow process. The use of mnemonic techniques or word lists can be helpful, but they focus entirely on word form and neglect the more complex aspects such as collocations and sense relations. New target language words should ideally be seen in context in order for their full range of meanings and usages to become clear.

It is in this area that computers can most successfully be used. Learners can be encouraged to play an active role in their own learning, by researching the target words which they want to add to their lexicon. When using a web-based learning tool, for example, students can easily search the Internet to find real-life contexts for a new word and thus become familiar with the communicative environments in which that word appears.
Learner autonomy can be further encouraged by maximising the interactivity of vocabulary-learning programs. According to Goodfellow (1994), the system should ideally allow the user to locate a target word of their choice in a discourse context, and to create a clear, manipulable representation of this word in a decontextualised list. The system should next offer an opportunity for the learner to process the word, by allowing him/her to discover and enter information about its meaning, grammatical functions and common usage. Finally, the learner should be given the opportunity to practice using their vocabulary, giving meaningful feedback on their errors.

It is important that vocabulary-learning programs promote a deep learning approach, and do not simply focus on form. They should encourage the learner to reflect on the structure of the target language and to make connections between words with common features. By consciously focusing on the semantic, formal and collocational similarities between words, the learner will be better able to form appropriate links in his/her mental lexicon. A vocabulary-learning tool which allows learners to organise words into various linguistic categories will therefore be more effective than one which presents a simple word-pair list. If related words are grouped together, they will be more easily memorised and retrieved.

Finally, the ideal vocabulary-learning program would enable the users to identify their particular learning approach and determine its effectiveness. By tracking the learner’s actions on their mouse or keyboard, the program could analyse the way in which he/she uses the available resources and suggest more efficient or helpful strategies which could be adopted. In this way, the learner would be encouraged to reflect on his/her learning techniques.

### 2.1.2 The evolution of vocabulary instruction programs

As computer technology has advanced, it has become possible for more of these criteria to be implemented by the creators of vocabulary-learning programs. Goodfellow (1995) describes the evolution of such programs in terms of three stages: first generation software, second generation software, and state of the art programs.
By this definition, **first generation computer-assisted vocabulary learning programs** are those that were created before the introduction of Graphic User Interfaces (GUIs). These programs focused mainly on testing vocabulary items already learned by the user, rather than introducing new words. Feedback and acceptable user input were greatly limited, due to the fact that strictly pattern-matching techniques were generally used.

Early examples of such programs include computerised versions of word games, such as crosswords. Later developments, such as Wordchip (Didascalia, 1993) allowed learners to create their own word lists by selecting lexical items from large databases. Such programs were useful insofar as they made effective use of repetition and review, but they failed to highlight the links between words in the mental lexicon, instead presenting each word in an independent, static way. Other programs encouraged learners to use mnemonic techniques to remember words, for example by providing pictures and short sample sentences associated with each new item.

In general, these early attempts at vocabulary-learning tools failed to provide much contextual information to the user. Interactivity was extremely limited, and few opportunities for deep lexical processing were available.

**Second generation programs** were able to make use of improvements in word processors and GUIs, enabling users to manipulate text more easily. These programs made better use of context, and were able to allow the learner a greater degree of autonomy and control over their learning. For example, programs were developed allowing users to select vocabulary items from texts, store them in files and re-use them in different contexts. Lexnet (Swartz 1990) illustrated the semantic links between words and encouraged learners to duplicate them.

During this phase, a variety of different lexical tools were developed. These included on-line dictionaries, which helped learners to autonomously discover new word meanings. For example, Dicologique (Hayet, 1993) made use of semantic hierarchies in organising vocabulary, and defined words in terms of synonyms and hyponyms. In this way, new words were learned as part of the learner’s existing lexicon, and their relationships with existing entries were made clear.
The development of concordancing tools such as Microconcord (John 1991) emphasised the role of the learner as linguistic researcher. The user examines data showing the target word in context and infers information about the word’s different meanings in different collocations, its main forms, derivatives, and discourse functions. The active role taken by the learner ensures that a deep level of processing takes place and vocabulary items are more easily memorized and recalled.

The latest technological developments have resulted in state of the art computer-assisted vocabulary learning tools, which support a wide range of vocabulary-learning processes. These programs combine context-rich resources, lexical tools and testing with feedback, in order to support learner autonomy as well as providing an instructional role. An example of such a program is Lexica (Goodfellow 1994), which allows learners to categorise their chosen words according to their semantic or formal relations to other items. The system tests vocabulary knowledge by generating cloze tests based on the information entered by the user.

2.2 Existing Software

Only a small number of CALL programs specifically dedicated to vocabulary learning exist. It is more common for wider grammar and comprehension programs to include a vocabulary section, which generally consists of nothing more than a word list. Examples of such systems include LUISA (which presents lists of words for which the user must give the Italian translation), and Français Interactif (which gives the user lists of vocabulary divided into different topics, together with an audio recording of their pronunciation). Systems such as these do nothing to promote learner autonomy, have a very low level of interactivity, and fail to encourage deep processing of the vocabulary items. At best, words are grouped together thematically. Semantic, syntactic and formal relationships between lexical items are not identified and exploited for learning purposes.

2.2.1 Vocabulary Trainer

Of the vocabulary-specific CALL programs available, VTrain 4.0 / Vocabulary Trainer (described in the CALICO Review by Olaf Böhlke, 2002) is a typical
example. VTrain 4.0 can be downloaded from the Vocabulary Trainer website, with licences available for both personal use and use by academic institutions. This is essentially a drill-and-practice program, based on the theory that by repeatedly practicing a particular vocabulary item, a student can automatise it, leaving him/her free to focus attention on higher levels of processing.

This program can be used for virtually any language, as its extensive authoring abilities allow users to create their own entries. The basic concept is that of a flashcard system. The user can choose any number of cards from a master set of flashcards, and these cards are then stored in the user’s personal cardfile box. The user then works through this cardfile, being tested on the flashcards one by one. Each flashcard consists of a question, which may be augmented by images, sounds or videos. The user enters his/her answer to this question, and is told whether it is right or wrong. Feedback is fairly limited and cannot guide the student by informing him/her which part of the answer is incorrect. Only when the student’s response exactly matches the answer stored on the ‘reverse’ of the flashcard is it accepted.

The system keeps track of the learner’s current vocabulary knowledge. It creates charts showing what percentage of the items in the current cardfile is known. Also, within each cardfile are five boxes representing different levels of proficiency. Each time a question is correctly answered, that flashcard is ‘promoted’ to the next level. In this way, the student can concentrate on those items which are less well known at any point.

The VTrain program contains many good design features. It allows the student to choose exactly which vocabulary items to learn in any particular session. This autonomy is further enhanced by the possibility of the user creating his/her own flashcards, thereby adding new vocabulary items to their lexicon. The questions on these flashcards can make use of mnemonic techniques such as audio or visual clues. By allowing the user to organise vocabulary into different cardfiles, the system allows related words to be grouped together into categories for a particular learning task. This helps the user to focus on the various relationships that exist between lexical items.
However, the flashcard system by nature does not provide a context for the new words, instead presenting them on an individual basis. As a result, the student will not learn the full range of meanings and usages of a word, so their vocabulary knowledge will be incomplete. Also, the limited and unhelpful feedback provided when the user enters an incorrect answer can be frustrating, and does nothing to help with the identification of problem areas.

### 2.2.2 Word Pilot

Word Pilot 2000 Premium Edition, reviewed in the CALICO review by Jill Turnbull (2001), is an example of a different approach to vocabulary learning technology. Designed to help students “become more effective writers and speakers of English” (www.compulag.com), it is a commercially available program which may be purchased both by individuals and by academic institutions. Here, the focus is on self-directed language investigation. The user is provided with tools such as a concordancer and Internet language resources, and must use them in order to collect and analyse L2 data.

The program provides libraries or specialised corpora, comprising a selection of texts from a specific genre. It also provides several word lists, including frequently used words and commonly confused words. The user can download extra libraries and word lists from the distributor’s website, or can create his/her own personalised ones.

When the concordancer is given a word from one of the lists, and a library through which to search, it produces lists showing the most frequent collocations of that word in the library. These lists can be modified to show one-word to four-word collocates, depending on the needs of the user. In this way, a student can identify fixed expressions involving the chosen word. The concordancer can also display one-line or larger contexts for the word, allowing the student to discover the different meanings a word might have, the similarities and differences in meaning between semantically related words, and in which contexts the different meanings are likely to appear. By comparing the concordance results for a given word in different corpora, the student can identify those genres and registers in which the word is most frequent, and the influence that genre has on meaning.
Once the user has discovered the syntactic and semantic properties of a chosen vocabulary item, it is possible to annotate that word. Dictionary definitions or sample contexts can be linked to the items in the word lists for future reference.

Word Pilot can also generate tests based on these concordance results. For example, it can take a full sentence containing a chosen word from a library, delete the word, and ask that the user chooses the correct answer from a list of commonly confused words to fill the blank. The program does not give feedback on the student’s response beyond indicating whether or not it was correct. This is partly because the principle behind the Word Pilot program is that the student should take responsibility for his/her own learning, and should use the concordancing tool to find out why a particular answer was incorrect.

Learner autonomy is clearly a very important part of this program. Virtually no language instruction is provided, although it is possible to use web links to access more pedagogical sites. Instead, the student must deduce the semantic features and syntactic properties of each word through analysis of the concordance results. This inevitably leads to deep learning on the part of the student, who must take advantage of clues provided from context as well as from word form in order to determine word meaning. Vocabulary learning will be more complete than if simple word lists were used, and the relationships between words will be more easily recognised.

However, once the student has discovered this information, the program does not provide an opportunity for further processing. The structure of the word lists is such that it is impossible to group related words together in a meaningful way. Even where words from a similar category appear grouped together, it is impossible to tell where one group ends and another begins. Furthermore, when the user adds a new word to the list, rather than being allowed to include it in an existing group, the word simply appears at the end of the list. As a result, the user cannot take advantage of the relationships he/she has discovered between vocabulary items when consulting the word lists.

The lack of meaningful feedback provided by the program can also be quite frustrating and demotivating. Since it only accepts a single correct answer for test
questions, and failing to explain why others are incorrect, the system does not help the user to identify problem areas in their vocabulary knowledge.

2.3 Current Version of Lexical Organizer

The program with which I will be working in the later stages of this project is the Lexical Organizer, designed in the Centre for Language and Communication Studies, Trinity College Dublin (O’Rourke 1998). This is a web-based vocabulary learning tool, which allows users to store and organise lexical items dynamically.

The Lexical Organizer consists of a workspace on which lexical items can be created, stored and manipulated. These items appear as coloured tiles on the workspace. By double-clicking on a tile, the user can view and edit annotations to the lexical item. Information about the meaning and usage of the word can be stored, along with authentic or user-created examples of the word in context. This information can include a direct L1 translation of the item, an L2 synonym, a paraphrase, the word’s grammatical category, the register in which it is most commonly used, a list of its frequent collocations, or anything else which the user deems relevant and useful. These Entry tiles can be moved around the workspace, to group related items together in order to help the user visualise the associations between them, or to form sentences.

Hierarchical categorisation is also possible. The user may store related lexical Entries in Folders, which also appear as tiles on the workspace. When double-clicked, these Folders open as another workspace, with all the same functions as the first. A Folder can contain more folders as well as lexical Entries, so that the hierarchy can be arbitrarily deep (Fig 2.1).

Within a workspace, further visual representations of the relationships between words are possible. Entries can be assembled together into Groups. Here, a container is drawn around the member entries. Entries can be added to or removed from Groups at any time, and can be moved around within the container. Links may also be drawn between Entries, to show more complex relationships between them. These links can be simple lines, or can include directional arrows. In addition, the colours of the tiles can be changed to illustrate yet another type of relationship between words.
Currently, the Lexical Organizer is in a basic state and has limited functionality. Even at this point, however, it meets many of the design criteria specified at the beginning of this chapter, and can be useful to the L2 vocabulary learner in many ways.

This program requires the learner to be autonomous and to take control of his/her own learning. Firstly, the user must decide on which words are to be added to the lexicon. The layout of the Entry annotation window encourages the student to discover and
enter information about the word’s meaning, grammatical function and usage, thereby forcing processing on several levels and rendering the word more memorable. Because authentic examples may be stored in the Entry annotation, the student is encouraged to find real communicative examples of the word in context, and will thereby become familiar with its different meanings and collocations. Allowing the user to also include his/her own examples encourages him/her to start using the vocabulary item productively rather than simply receptively. At this point, however, the program is not fully interactive. While the user is free to create and organise lexical Entries, there is still no testing function and no opportunity for the user to receive feedback from the program.

Deep learning is further facilitated by the hierarchical categorisation feature of the Lexical Organizer. By encouraging the user to group words together according to categories, the program forces reflection on the structure of the L2 lexicon. The student is free to construct the lexicon in a way that he/she finds meaningful. While this may not, and probably will not, be an accurate reflection of their mental lexicon, it is nonetheless useful to consciously focus on the various relationships between words. In addition, the flexible, dynamic nature of this hierarchy reflects the ever-changing mental lexicon. As the student discovers more information about a lexical item, it can be moved from one Group to another.

This program also contains a valuable feature not found in many vocabulary-learning tools. A user can make his/her lexicon public, so that it can be viewed by all other users of the system. A guest viewer of a lexicon will not be able to make changes to its structure, or to add or delete Entries. However, by looking at the different ways in which other students have organised their lexicons, a learner will notice different connections between words and will find new meanings of which they were not previously aware. Being able to view another learner’s sample sentences will also provide more opportunities to see given words in context.

As well as being a purely vocabulary-learning tool, the Lexical Organizer has the potential to help student writers plan their work. For example, a student could create a Folder to represent a piece of work, with sub-Folders representing each section or paragraph within it. These sub-Folders would contain Entries for the key words or
ideas to be expressed within the section. These could be moved around and re-organised until the most satisfactory arrangement was reached. Clearly, this type of application has great potential as a collaborative exercise. A group of learners at the same workstation, or indeed individual students making use of the Public Organizer feature, could work together discussing the plan and structure of their text.

Clearly, the Lexical Organizer is already an extremely useful vocabulary-learning tool. However, additional functionalities are necessary for it to achieve its full potential. In particular, it is important that the program provides some feedback or testing function, so that students can track their progress and evaluate the effectiveness of their learning strategies.
Chapter 3
Additional Functionality

3.1 An Overview of the Design of the Lexical Organizer

As described in the previous chapter, the Lexical Organizer is a web-based tool for the storage and learning of vocabulary. The user sees only the online web-page, which displays the Lexical Organizer as an applet. It is through this applet that the program is launched and the user is able to log on to the system. However, the underlying architectural structure of the Organizer is somewhat more complex than this simple user interface would suggest. The source code of the program is divided into two sections: the Delphi code which is concerned with the background administrative workings of the program, used by the Organizer Administrator; and the Java code (see Appendix B) which describes the webpage and Organizer applet which are visible to the user.

The basis of the Lexical Organizer is a Microsoft Office Access Database, in which all of the information relating to each individual organizer’s hierarchies and vocabulary is stored. This database consists of various tables containing information such as the names, co-ordinates, colour and location in the organizer hierarchy of each Entry and Folder item in each Organizer. It also stores information about the various Groups and Links which exist within the Organizers, as well as the semantic and grammatical information which has been annotated to member Entries. Within the database, a unique ID number is assigned to each Organizer, Folder, Entry, Group and Link which exists in the overall Lexical Organizer program. In this way, a large number of different users can create individual Organizers and populate them with their own personal Folder hierarchies and Entries, without interfering with those of other users.

The information stored in this database is accessible only through the Lexical Organizer Server, which is defined in the Delphi section of the source code, and is available to the Java code which comprises the main part of the Organizer program.
Also defined in the Delphi section of the code is the Lexical Organizer Configuration Panel, which the Organizer Administrator uses to connect the Lexical Organizer Server to the Database, and to manage user accounts (see Appendix A). In the Configuration Panel, the Organizer Administrator can connect to the Server and create, delete and rename Organizers, as well as set and change login passwords.

These administrative sections are the only durable parts of the Lexical Organizer. The applet displayed on the web-page does not store any lexical or structural information of its own. Instead, it simply connects to the Server and reads the information from the database. When a user logs on to the system and opens his/her Organizer, the applet calls on the Java code which then uses the Server to access the database and read the information relevant to the current user. Using this data, a new version of the Organizer is created and displayed online. As this visual representation is manipulated and expanded, the changes are recorded in the database so that the next time the user logs on, the new version of his/her Organizer that is constructed will contain all of the correct, updated information.

It is in this way that so much information can seemingly be stored in one small webpage. Any number of users may be logged on at any one time, each accessing the database independently via the website. Each user may manipulate his/her own personal Organizer, and can also view those which other users have made public by logging on as a guest. However, all of this data is simply stored in a single database, which is accessed independently by each user who is connected to the Server via the webpage.

3.2 Improving the Flexibility of the Lexical Organizer

As described in the previous chapter, one of the main strengths of the Lexical Organizer program is its flexibility. The user is completely free to structure the lexicon in whatever way seems most appropriate to him/her based on the relationships between target vocabulary items, with the result that the Organizer reflects the various links between the words. However, in the original version of the Organizer, this flexibility is somewhat limited. While more information can be annotated to an Entry at any time, and it can be Grouped with or Linked to other Entries within the same
Folder to illustrate the various relationships it holds with them, an Entry must remain
in the Folder in which it was originally created. It cannot be moved around the Folder
hierarchy, or belong to more than one Folder at the same time.

3.2.1 Copy Entries

This feature allows the user to make a copy of any Entry in the Lexical Organizer, by
simply right clicking on the Entry tile and choosing Copy To from the pop-up menu.
A dialogue box appears on the screen, containing a list of all the Folders in the user’s
Organizer. The user can select the Folder to which he/she wants to copy the selected
Entry by clicking on its name in this list, or by typing it into the text field provided.
When the user presses enter or clicks on the Select button, a copy of the selected
Entry is made and appears in the chosen Folder. This copy has the same name and
annotated information as the original Entry, but is now an entirely separate entity.
Any changes which are made to the name, position, colour or stored information of
one copy will not affect the other, unless the “Update Copies” feature, described
below, is used when adding information to the Entry.

When making a copy of an Entry, the user can choose to remove the original Entry
from the Organizer, in effect simply moving the Entry to a new Folder. This can be
done by simply ticking the “Delete original” checkbox in the “Copy To” dialogue
box. This feature is useful in the event that the user discovers that he/she has simply
miscategorised a word and wishes to completely remove it from its current location.

In addition, by ticking the “Blank copy only” checkbox in the “Copy To” dialogue
box, the user can create an empty copy of the selected Entry. This results in an Entry
with the same name as the original, but with no annotated information, appearing in
the specified Folder. In this way, a copy of the Entry appearing in one Folder or
category could be used to store information about one aspect of its meaning and
usage, while another copy could appear in a different Folder and focus on a different
aspect. This context-based manner of storing and presenting the various semantic and
syntactic features of a vocabulary item avoids any suggestion that one of a word’s
meanings is more important than the others. The user will simply learn that different
features of the word belong to different situations and contexts, as illustrated by the
different categories in the Lexical Organizer. By allowing the user to separate out the different meanings and contexts of a particular vocabulary item, this feature helps the learner to avoid the typical mistake, described in Chapter 1, of focusing on just one meaning of a word and ignoring the others, even when the surrounding context shows that meaning to be incorrect.

Alternatively, in situations where a word has the same meaning regardless of context but can still be categorised in different ways within the user-defined hierarchy, the user can simply use the default settings of the “Copy To” feature together with the “Update All Copies” feature. When making a change to the information stored in an Entry, simply pressing the “Update All Copies” button makes sure that all the copies of that Entry throughout the Organizer also receive this new information. However, as this feature makes all the copies of an Entry identical, it should not be used for situations in which one copy needs to have information that another copy does not.

Another potentially useful application of the “Copy To” feature is the possibility of creating copies of an Entry within the same Folder as the original, which can be done by simply choosing the current Folder from the list which appears in the “Copy To” dialogue box.

The Copy feature greatly enhances the flexibility of the Lexical Organizer so that it more accurately reflects the dynamic nature of the learner’s L2 mental lexicon, which is continually being restructured and reorganised to accommodate new information and to forge new connections between lexical items. As a learner discovers new information about the meaning and usage of a target vocabulary item, he/she can now re-categorise it quickly and easily. An Entry is no longer confined to a single Folder but can exist in many different categories and in different relationships to other words.

### 3.2.2 Practical Examples

The advantages to the learner of this organisational flexibility are best illustrated through the use of practical examples.
In these examples, the user is creating a Folder to store semantically related words around the theme of “La maison” (the house). Within this main Folder, various sub-folders are created to represent the different rooms of the house. Each sub-Folder contains vocabulary relevant to that room.

(a) The user creates a Folder to represent “La cuisine” (the kitchen), and adds an Entry “serviette” (towel) within the Folder. She then realises that this Entry also belongs in the “La salle de bains” (the bathroom) Folder. The student then uses the “Copy To” function to create an exact copy of the “serviette” Entry in the “La salle de bains” Folder. This example is illustrated in Figs 3.1 to 3.3.

(b) Now, the student decides to add more information to the “serviette” Entry, which currently simply contains a direct translation of the word. As the word has an identical meaning and usage in both contexts, the user can avail of the “Update all copies” function when adding this information, so that simply changing one of the copies is enough to ensure that both receive the new information. This example is illustrated in Fig 3.4.

(c) Next, the user adds an Entry “lavabo” (sink) to the Folder “La cuisine”. Later, she discovers that this word for sink is actually more suitable for a bathroom. The correct word for a kitchen sink is “évier”. Therefore, she uses the “Copy To” function to move the Entry to the “La salle de bains” Folder, and ticks the “Delete original” checkbox so that the Entry is removed from the “La cuisine” Folder. This example is illustrated in Figs 3.5 to 3.7.

(d) The user then creates an Entry “couteau” (knife) in the “La cuisine” Folder, and annotates information such as a translation, the gender and part of speech of the word, and a sample sentence. She then creates an exact copy of this Entry in the same Folder, containing all the same information. She renames this Entry “couteau à beurre”, and updates its meaning to “butter knife”, keeping all the other notes and examples intact. The
original Entry remains unchanged. This example is illustrated in Figs 3.8 to 3.12.

![Fig 3.1: Before copy function is used](image1)

![Fig 3.2: Copy function](image2)

![Fig 3.3: After copy function is used](image3)
An Online Tool for L2 Vocabulary Collection & Learning

Chapter 3.

Fig 3.4: Update all copies function

Fig 3.5: Copy and remove original

Fig 3.6: Before copy and remove function is used

Fig 3.7: After copy and remove function is used
Fig 3.8: Before copy and rename  
Fig 3.9: After copy and rename

Fig 3.10: Copy to same folder

Fig 3.11: Contents of original entry  
Fig 3.12: Contents of copy
Finaly, the user goes to the Folder “Les Fruits” (fruit) elsewhere in the Organizer, where the names of different types of fruit are stored. Here, she creates an Entry “cassis” (blackcurrant). While searching for information on this word, she notices that it has a second meaning, “a dip in a road”. She uses the “Copy To” function to send an empty copy of the Entry “cassis” to the Folder “La circulation” (traffic). This blank copy simply shares its name with the original Entry, so the user can fill it with information and examples which are relevant to this second context at a later point.

As we can see from these examples, as well as having the pedagogical benefits described earlier in this chapter, the new Copy function makes the Lexical Organizer considerably more user-friendly by cutting down on inauthentic labour and reducing the overall workload for the user. As it is now so easy to copy or move an Entry to a new Folder, the student is encouraged to discover and record different meanings and contexts for each new vocabulary item he/she encounters.

3.3 Improving the Interactivity of the Lexical Organizer

As we have seen in Chapter 2, the Lexical Organizer is already a reasonably interactive program, according to Goodfellow’s (1994) definition of interactivity. The users are required to select target words of their choice and to create Entries representing those words. They are then encouraged to process the words on several levels, by inputting semantic, grammatical and contextual information into the Entry, and by organising the Entries into related categories or Folders. However, the Lexical Organizer as yet provides no real output or feedback to help the users with their vocabulary learning. While it provides an excellent method of organising and storing new L2 vocabulary items, the program still needs to encourage the user to regularly review these items so that they can become a permanent part of his/her mental lexicon.
3.3.1 Test Definitions

This feature represents a first step towards a more fully interactive Lexical Organizer. It allows users to test themselves on vocabulary items of their choice, and offers limited feedback on their performance. By right-clicking on any tile displayed on the Lexical Organizer workspace, and choosing the Test Definitions option from the pop-up menu, the user can launch the test program for that item. A dialogue box appears on the screen, presenting options for the form that the test will take, and text fields in which the user can read the question and submit the answer (Fig 3.13).

If the user has opted to test him/herself on a Folder, the dialogue box allows him/her to choose between simply testing the Entries that appear on that Folder’s immediate workspace, and also testing those Entries that appear in its sub-Folders. If the user ticks the “Test All Sub-Folders” checkbox, as he/she proceeds through the test, new dialogue boxes will appear to represent the tests for the new Folders. These dialogue boxes have the name of the folder being tested as their title, so if the user wishes to skip a particular folder, he/she can quickly identify it and can simply close the dialogue box. The test will proceed, displaying the next Entry or Folder to be tested. Otherwise, if the user does wish to take the proposed test, he/she can simply work through the questions as they are presented. When the test for a particular Folder is completed, its dialogue box will disappear and be replaced by the next one.

In the case that the user has simply selected a single Entry to be tested on, the same dialogue box appears but the “Test All Sub-Folders” checkbox is deactivated.

Within the test dialogue box, the user can also choose to accept the default setting, in which they are presented with the word to be translated in the L2, and asked to give their native language translation of it. Alternatively, by ticking the “Translate from L1 to L2” checkbox, the user can choose to be given the word in their native language and enter the L2 translation. In both of these settings, the L2 version of the word is taken as being the title of the Entry, while the native language translation is taken as being the information given in the “Meaning” section of the annotated notes to that Entry. Once this selection has been made, the user presses the “Next Question” button to start the test (Fig 3.15).
The test begins by displaying the first word to be translated in the text field labelled “Translate”. The user may now type his/her answer into the text field labelled “Answer”, or click the “Show Clue” button to ask for a hint. The clue will be displayed in the text field labelled “Clue”, and consists of the information stored in the “Notes” section of the annotation to the Entry being tested. If no notes are available, the “Clue” text field simply displays the message “There are no clues for this question” (Fig 3.14).

Once the user has typed his/her response into the “Answer” text field, he/she may submit the answer by pressing the “Submit” button. If the answer is correct, the “Right Answer” text field in the dialogue box displays the message “Correct!” . If the user’s answer is incorrect, however, the “Right Answer” text field displays both the question and the answer together. The test program automatically moves on to the next question in the Folder, if it is available, and displays it in the “Translate” box. The “Answer” and “Clue” text fields are automatically cleared, to avoid confusion (Fig 3.16).

In the event that an Entry is empty, the test function simply skips over it and moves on to the next Entry, as it would be impossible to test knowledge of a word for which no translation is given.

This program relies on a simple pattern-matching technique to determine whether or not an answer is correct. If the user’s response differs in any way from the response stored in the Entry’s annotated notes, it is deemed to be incorrect. Clearly, this method is extremely limited as there are almost always a wide variety of answers which could be acceptable. It is for this reason that the test function does not include a score-keeping feature, as students may find it frustrating to be marked down for extremely superficial differences between their answer and that stored in the computer. It is also the motivation for the fact that the program simply displays both the question and the answer in the “Right Answer” field rather than leaving the user to make repeated guesses until he/she hits the exact formulation that the computer is looking for. Instead, he/she will be able to recognise that his/her proposed answer was either very close to that given by the computer (perhaps, for example, only
differing in the article used), or indeed that he/she was wrong. With the current design of the test function, while the feedback is severely limited, user frustration is avoided by recognising and accepting these limitations until such a time as they can be eliminated.

3.4 Further Attempted Improvements and Technical Difficulties

3.4.1 Print Entries

I hoped to introduce a “Print” feature into the Lexical Organizer, which would allow the user to select an Entry and print it out in the form of the flashcard. The flashcard would contain the name of the Entry, its meaning and notes, and any examples which had been annotated to it in the Organizer. Each flashcard would be laid out identically, regardless of the quantity of information printed on it. For self-testing purposes, the name of the Entry would be printed on the top of the card, which could be folded over to create a double-sided test card.
The planned design of this feature was that, when the user right-clicked on an Entry and chose “Print” from the pop-up menu, the program would automatically create a text document named after the Entry and containing its corresponding flashcard. I intended to make use of the Java IO Library, which allows Java programs to create and write information to text files. However, when I compiled and attempted to run this code, it was unsuccessful. No error message appeared, so I can only conclude that the problem was caused by security settings on the computer being used to run the program, or possibly by interference from the IO stream being used by the program for communication with the Server.

If this Print facility could be successfully implemented, the user would effectively be able to create a hard copy of his/her Lexical Organizer, which could be used to review vocabulary items on a regular basis, even when access to a computer or to the Internet is not possible. We saw in Chapter 1 that frequent review of new words is a vital part of the learning process and greatly increases the likelihood of a vocabulary item becoming a stable part of the L2 mental lexicon. The availability of a compact, portable, yet detailed representation of these new items would make it easier for students to fit this revision into their study schedule.

### 3.4.2 Automatic Test Function

Another feature which I initially intended to introduce into the Lexical Organizer program was an automatic test function. The concept behind this feature was that of assigning a “birthday” to each new Entry in the Lexical Organizer. The Entry would then be automatically reviewed, either through testing or simply being brought to the user’s attention, at regular intervals following this “birthday”. For example, based on (CITATION)’s recommendations for the review of new vocabulary items, the Entry should be automatically tested at periods 24 hours, one week, one month, six months and one year after it is initially created.

I attempted to introduce this function by adding a CreationDate feature to all new WordEntries. This CreationDate, which would take the form of a GeorgianCalendar object (from the Java Text Library), would store the date on which the Entry was
created. Based on this date, five review days corresponding to those suggested by (CITATION) above were created. A final GeorgianCalendar object would store the date upon which the Entry was last tested. This date would be initialised as the CreationDate, and would be automatically updated each time the Entry was tested by the user. I then introduced a method which, when the user selected “Review Vocabulary” from the pop-up menu, searched the Organizer for Entries which had a review date corresponding to or preceding the current date, and whose LastTested date also preceded the review date in question – that is, it had not been tested since the most recent review date had passed. Any Entries matching these criteria were then presented to the user to be tested.

Unfortunately, my attempt to implement this function failed due to the dynamic way in which the Lexical Organizer program is designed. As each Organizer is created anew each time the user logs on, this means that each WordEntry is also newly created on each occasion. As a result, the CreationDate object was continually being initialised to the current date, so the review dates could never come to pass. The only solution to this technical problem is to introduce a new column into the Entries table in the Access database on which the Lexical Organizer program is based. This column should contain the creation date of the Entry, from which the review dates can be calculated. As it will be stored in the database, it will not be affected by the opening and closing of the user’s Organizer. Successfully implementing this change will require making some changes to the Delphi code, which describes the relationship between the Server and the database.

If this functionality were to be added to the Lexical Organizer, the user would be given an opportunity to review all of his/her target vocabulary items at appropriate intervals. This would ensure that no items would be forgotten about, regardless of their position in the Organizer hierarchy, thereby greatly increasing the user’s chances of successfully committing all of his/her target words to memory.
Chapter 4
Conclusions

4.1 Future Work: Additional Functionalities

While the Lexical Organizer is at this stage a useful vocabulary learning tool, with a variety of pedagogically motivated functionalities which will greatly assist users in organizing both their target vocabulary and their time effectively for learning purposes, there is still scope for further improvement to its design and implementation. The following are but some of the many functionalities which could be added to the Lexical Organizer at some point in the future.

4.1.1 Print

As explained in the previous chapter, the introduction of a Print feature to the Lexical Organizer would be extremely beneficial to the user. For example, should the function already described be successfully implemented, the possibility of a student being able to print his/her Entries out in flashcard form to be used for revision or self-testing activities would help the learner to make the most efficient use of his/her time.

This Print Entry feature could easily be extended to allow a student to print out all of the component Entries of any given Folder, or even of his/her entire Lexical Organizer, as a set of flashcards. If the solution side of the flashcard included information about the parent Folder of an Entry, for example, the student would be able to perform self-testing activities such as those described by Carter (1998), in which words related to a selection of different semantic categories are presented jumbled together on a page, and must be sorted into their separate categories by the learner. These games are based on Rudska’s (1991) theory that visual representations of the links between words can be stimulating and lead to better recall of those words.
In addition, it would be useful for a student to be able to print the contents of his/her Organizer, or of a Folder within the Organizer, in the form of a word list. As we saw in Chapter 1, word lists can be extremely helpful for second language vocabulary learning, particularly for beginner-level students. These word lists could visually represent the hierarchical structure of the Lexical Organizer by indenting across the page so that words at the same level of the hierarchy will be indented to the same degree, for example. They could also retain the relationships indicated by the colours of the Entry tiles in the Organizer, perhaps by printing out the words in the colour that the corresponding Entry appears in on the screen. In this way, the word lists would be more informative and visually stimulating, and therefore more pedagogically useful, than traditional lists in which target vocabulary items appear in vertical columns on a page. An example of a possible layout for such a word list is given in Figure 4.1.

![La maison](image)

---

**La maison**

- *la maison*  
  *the house*
- *porte*  
  *door*
- *fenetre*  
  *window*
- *pieces*  
  *rooms*

**salle de bains**

- *bain*  
  *bath*
- *douche*  
  *shower*
- *serviette*  
  *towel*
- *se laver*  
  *to wash oneself*

**chambre**

- *lit*  
  *bed*
- *armoire*  
  *wardrobe*
- *dormir*  
  *to sleep*

**cuisine**

- *serviette*  
  *towel*
- *couteau*  
  *knife*
- *couteau de buerre*  
  *butter knife*
- *frigo*  
  *fridge*
- *faire les vaisselles*  
  *to do the washing up*
- *cuisiner*  
  *to cook*

*Fig 4.1: Suggested layout for Print Word List function*
It may also be possible for these word lists to in some way visually represent the various Group and Link relationships which may exist between Entries in the Organizer. For example, perhaps a box could be drawn around Grouped words. The more information from the structure of the Organizer can be incorporated into the word lists, the richer the visual representation will be and the more effective the word lists will become in aiding retention and recall of the target vocabulary items.

4.1.2 Test

The Lexical Organizer program could be further enhanced by improvement and expansion of the Test function, which is currently quite basic. A successful implementation of the “birthday”-based automatic test feature described in Chapter 3 would ensure that the student would get the maximum benefit from his/her Organizer, as every word would be reviewed at regular, theoretically-motivated intervals.

Another possibility for a future Test function would be to encode some game-like activities such as those described in the previous section. For example, Entry tiles from two or more different Folders could appear jumbled together on a workspace, and have to be divided into Groups corresponding to their separate semantic characteristics (which would correspond to their membership of different Folders). Such activities would take full advantage of the Lexical Organizer’s highly visual manner of representing relationships between words. An example of what this activity would look like is given in Figures 4.2 and 4.3.
Hints for this test could be given by providing the correct number of empty Group containers, so that the student will know how many different semantic categories there are in the test. Different difficulty levels could also be set, for example by allowing the user to open the annotated information to the Entry tiles at the easy level, at making this impossible at the more difficult level.

The simplicity and extreme flexibility of the Lexical Organizer’s visual interface are clearly some of its main strengths, and this Test function would make good use of these unique qualities to provide a functionality which is not available in other, more traditional vocabulary learning programs.

4.1.3 Copy

While the Lexical Organizer now has quite an advanced set of Copy features which allow the user to copy or move Entries from and to any Folder in his/her Organizer, resulting in an extremely high degree of flexibility in the structure of the lexicon, further enhancements to this feature could be introduced.

Currently, only Entry items can be moved around the Organizer hierarchy. It is not yet possible to copy or move an entire Folder in one step. This feature could be usefully added to the program, as situations may arise in which a user discovers that he/she has miscategorised a Folder after filling it with a large number of Entries. In the current version of the Organizer, moving each of these Entries individually to a newly-created Folder in the correct location could be a relatively time-consuming and laborious task. Ideally, a Copy Folder function could reduce this workload to a single step.

It could also be useful to consider adding a Copy function allowing guests visiting a public Organizer to copy an Entry which they see there into their own personal Organizer. In this way, a student who, while viewing another user’s Organizer, discovers some vocabulary items of which he/she was previously unaware would be able to add these items to his/her own Organizer quickly and easily. The guest would still not be allowed to alter the public Organizer, and could make any changes he/she
considered necessary to the copy of the Entry located in his/her own Organizer. This feature would decrease the workload involved, and might encourage a guest to benefit more from his/her viewing of a public Organizer by eliminating the laborious process of transcribing the information from one Entry to another, as well as removing the possibility of errors appearing in the copied Entry.

4.2 Suggested Uses of the Lexical Organizer

We have already seen how the Lexical Organizer differs greatly from previously existing vocabulary learning tools and programs, and seen some practical examples of its various features being used. So far, these examples have been based on the idea of an individual user creating and using his/her own personal Organizer for the purposes of storing, categorising and reviewing new L2 vocabulary that he/she has encountered, in class or elsewhere. However, the Lexical Organizer can easily lend itself to different types of usage, such as those described below.

4.2.1 Collaborative Learning

As well as being used purely for vocabulary storage and learning purposes, the Lexical Organizer can be used as a workspace for planning written compositions in the user’s second language. A Folder representing the overall task could be created, containing sub-Folders to represent each of the sections or paragraphs within the compositions, which in turn could contain lexical Entries signifying the keywords for these sections. In this way, the user could easily make a structural plan for his/her composition, moving Entries from one Folder to another until the most suitable arrangement has been reached. While this brainstorming and planning process can be carried out by students individually, it can be even more productive, stimulating and therefore conducive to language learning, when a group of students collaborate on a project.

This collaboration could take place face to face among learners sharing a single workstation and working together on a single Organizer. As the students discuss the topic of the composition, plan a structure for the text, and rearrange Entries (lexical items) and Folders (structural features) on the screen, they will communicate their
thoughts, opinions and suggestions to each other. Provided this communication takes place in the second language, they will learn not only relevant vocabulary for the composition at hand, but also how to express themselves to others through their target language. The user-friendly, simple nature of the Lexical Organizer will ensure that all communication between students will be authentic and task-focused, and little effort will be wasted on coming to terms with the technology rather than with the language.

If this face-to-face communicative situation is not possible, the Lexical Organizer can equally be used for long-distance, asynchronous collaboration. The Organizer is a web-based tool which can be accessed by any number of users anywhere, at any time. Students can also choose to make their personal Organizer public, so that it can be viewed by guests. As a result, there are several different ways in which on-line collaboration on a composition task could take place.

One possibility is for a group of students, who are unable to arrange a face-to-face meeting, to create an Organizer in which to collaboratively plan their composition. Each student will be given the name and password of this Organizer, so that he/she will be able to access it. When a student opens the Organizer, he/she will be able to see the combined efforts of those students who have already made their contributions. He/she will be able to add Entries, annotate any Entries already present, and move these Entries to different Folders (representing different parts of the text). A student will be able to visit the Organizer as often as he/she wishes, to see how the plan is progressing. The final result will be the combination of the contributions of all the students in the group.

Alternatively, each member of the group could create a Folder to represent the composition within his/her own private Organizer. Within this Folder, he/she could develop his/her own plan for the text. Each student would then make his/her own Organizer public. In this way, each student would be able to view the plans constructed by the other group members, but would only be able to alter his/her own plan. Students could therefore create individual plans, but take ideas from the proposed structures and vocabulary items of other students. The advantage of this mode of operation is that each student gets the benefit of fully planning the
composition for him/herself, but is still able to learn from the suggestions of others, and can expand his/her vocabulary by taking previously unknown words that appear in the Organizers of others.

### 4.2.2 Research Possibilities

Another potentially interesting area in which the Lexical Organizer could be used is that of research. Because the Organizer is an entirely student-driven program, unlike other more traditional vocabulary learning tools, it could offer insights into the learning strategies favoured by students of second languages.

With the Lexical Organizer, the student is entirely free to structure his/her lexicon in whatever way seems most appropriate to him/her, and to categorise lexical entries according to whatever criteria seem the most useful for learning purposes. The user is given no instructions as to what relationships to record between words, what features are most important, or what type of information should be annotated to the Entries.

This offers a sharp contrast to most other vocabulary-based language learning programs in existence. As we saw in Chapter 3, the traditional approach was to provide the user with a list of vocabulary to be learned, together with an L1 translation, and then test him/her on this list. The lexical items in these word lists were often semantically unrelated, perhaps being chosen to illustrate a particular grammatical point or to facilitate the reading of a particular passage. In more recent times, the flexibility of vocabulary learning programs has increased, with users often being allowed to choose which items to focus on from a list, or from a text. Some programs, for example Vocabulary Trainer, allow the user to organize their chosen words into grammatical or semantic categories. However, the Lexical Organizer has a unique way of representing these categorizations and relationships visually, which greatly increases the user’s freedom to illustrate the various relationships which he/she perceives as existing between lexical entries.

As a result, the Lexical Organizer has the potential to allow us to discover more about the manner in which learners perceive the second language lexicon. By studying the way in which authentic language learners use and structure their Organizers, and the
type of relationships they establish between Entries within Folders, researchers could reach a better understanding of the vocabulary learning strategies of second language learners. This could help them to develop new teaching methods, presenting vocabulary to students in more meaningful ways and exploiting the semantic or grammatical connections to which learners typically attach the most importance.

### 4.3 The Use of Computers in Vocabulary Learning

In this project, I have focused on the particular example of the Lexical Organizer program to illustrate some of the ways in which computers can be used to aid second language vocabulary learning. However, there is clearly no single ideal form for such a program to take. As we saw in Chapter 1, a wide variety of different learning strategies can be adopted by different students, with no single approach suiting all learners. Indeed, even an individual learner will progress through several different approaches as his/her proficiency in the target language develops. For example, beginners in a second language can benefit from explicit, directed learning, whereas more advanced students need to be given an opportunity to develop their implicit learning skills.

As the thorough learning of new vocabulary items is clearly a complex task, it is important that the designers of vocabulary learning software are aware of, and make every effort to benefit from, current theories on the most effective methods of teaching and learning vocabulary.

A good vocabulary learning program should aim to facilitate the use of as many of these different strategies as possible, and should include both discovery and consolidation strategies. That is to say, as well as providing students with help in committing new vocabulary items to memory, the program should allow them the opportunity to discover for themselves the words which they wish to learn, as research (Hatch & Brown, 1995) has shown that lexical items in which the reader has a particular interest, and a specific reason for wanting to learn, are more easily memorised and assimilated into the L2 mental lexicon. The use of web-based vocabulary learning tools offers students the possibility of quickly and easily locating authentic target language material on the Internet, in which they can locate unknown
lexical items to be learnt. A self-contained vocabulary learning program should ideally try to parallel this vast resource by providing large corpora of second language material, from which students can select items of interest to them, as this will ultimately lead to more effective learning than the simple provision of pre-defined word lists.

Once the student has selected the words which he/she wants to learn, the next step, as described in Chapter 1, is that of determining its meaning. Learners at different levels will wish to use different strategies in order to achieve this. Beginners generally prefer to be given either an L1 translation or an L2 paraphrase of the word, whereas advanced learners prefer to initially attempt to deduce the meaning for themselves from context, and check their answer against reference material (Schmitt 1997). Again, web-based programs have the advantage of allowing the user to make his/her own choice in this area. Beginner students can easily use the many on-line reference tools which are available, while for advanced students the surrounding context in which they initially located the word will help them to make an educated guess as to its meaning. There is also the possibility of searching the Web for more examples of the chosen word in context. Self-contained programs should include a reference tool in which students can find definitions of their chosen words, as well as a concordancing tool allowing them to find more examples in the corpora.

Most importantly, vocabulary learning tools should include some of the consolidation strategies described in Chapter 1, which help a new vocabulary item to become a permanent part of the learner’s L2 mental lexicon. An important concept at this point is that of “deep processing”, meaning that the learner should not focus solely on the form of the word but should also study various features such as the word’s grammatical role, its common collocations, its different semantic features and the various contexts in which they are applicable, its position in the lexicon, and so on. Therefore, the program should ideally allow the user to include as much of this assorted information as possible in his/her representation of a word. For example, the Lexical Organizer does this through both its hierarchical structure and the notes which can be annotated to each entry. Whatever form this encoding of information takes, it is crucial that it is present as without it the student cannot achieve a full knowledge of the vocabulary item.
In addition to this deep processing approach, a vocabulary learning program could benefit from incorporating some of the other consolidation strategies described in the literature. For example, the program could encourage the student to use mnemonic strategies such as the keyword technique or imagery to help him/her to remember a word, by allowing the user to store a visual representation of the word together with its definition.

The program could also make use of metacognitive strategies by allowing the user to identify and analyse his/her own preferred learning strategy by tracking his/her use of the program and highlighting behaviour patterns which might be significant. For example, a particular student may have a habit of copying sample sentences from authentic material rather than coming up with his/her own examples, and as a result might have difficulty using new vocabulary items productively. By drawing a user’s attention to such potential problems, the program would encourage him/her to evaluate the effectiveness of his/her learning techniques and to be aware of different possible approaches. However, this would require quite advanced technology in order to accurately track the student’s moves and analyse them so as to make helpful suggestions.

Finally, as we have seen that regular review of new vocabulary items is a crucial part of the learning process, good vocabulary learning software should include a test mechanism allowing users to quiz themselves on their vocabulary knowledge. The most dynamic and efficient way of doing this is to use the information which the user has entered about the word as the basis for the test for that word, rather than storing a vast amount of material which may potentially never be used by the student. However, this approach generally results in using a straightforward pattern-matching technique to check whether the right answer has been given, which can lead to student frustration if an alternative, equally correct answer is rejected because it does not take exactly the form of the stored answer. Until natural language processing software improves, it will be quite difficult for a computer program to offer meaningful feedback on spontaneously-produced user input.
While it is clearly important for software designers to take current research on effective vocabulary learning into account when creating vocabulary learning programs, it is nonetheless vital to remember that computers still have their limitations. It will not be possible to implement in a piece of software all of the strategies recommended for effective language teaching. For example, as we have seen, computers are as yet no match for a human teacher in terms of correcting student answers to questions and offering helpful feedback on any errors.

Therefore, it is often equally useful to add features which may not necessarily be theoretically motivated but which can easily be implemented on computers. These could include authoring tools allowing the user to create exercises based on his/her target vocabulary, such as sorting a selection of jumbled words into categories based on specified criteria, or reconstructing a sentence from a set of jumbled words. While these activities do not directly correspond to any of the learning strategies discussed in the literature, they do encourage the student to reflect upon the structure of the L2 lexicon, and upon the syntactic structure of L2 sentences, and therefore do have considerable pedagogical value.

In order to adequately cater for the wide range of learning strategies adopted by second language vocabulary learners and teachers, therefore, it is important that designers of vocabulary learning software both learn from current pedagogical theory and fully exploit the potential offered by computers. As I have shown, the Lexical Organizer program is one example of how these two influential factors can be combined usefully for the second language learner, but there is still considerable room for improvement in its implementation. This reflects the situation in the wider field of computer aided second language vocabulary learning – while considerable advancements in available software have been made in recent years, the area still has great potential for future development.
Appendix A

Technical Information

Installation Instructions

In order to successfully install and run the Lexical Organizer, and view and edit all of its Source Code, your computer should be equipped with the following software:

- Internet Explorer
- Microsoft Office 2000
- WinZip
- Java JDK Version 1.5.0 (available free to download at www.java.sun.com)
- Java Runtime Environment Version 1.5.0 (available free to download at: www.java.sun.com)
- JBuilder 9 (available free to download at: www.borland.com/products/downloads/download_jbuilder.html)
- Delphi software (available free to download at: www.borland.com/products/downloads/download_delphi.html)

Once the Lexical Organizer files have been downloaded and unZipped, the Server must be installed.

   a) Open a command prompt and go to the directory in which the Organizer files are located.
   b) Type orgserv /install.

The server is now registered, and the Configuration Panel can be used to complete the installation process. You must now specify a datasource for the server, and connect it to the server.
c) Make a copy of the “empty.mdb” Access Database located in the installation folder and rename it. This file will be the datasource for the server.
d) Open the Lexical Organizer Configuration Panel and log on as an Administrator. Press the “Set Database Connection” button.
e) Select “Microsoft OLE DB Provider for ODBC Drivers” from the Provider panel.
f) Select the "Use connection string" option to specify a datasource. Press the "Build ..." button.
g) When requested for a DSN name, press the "New ..." button. This will open a new dialog:
   1) In the first screen, select the "Microsoft Access Driver".
   2) In the second screen, enter a name for the data source file to be created (any name will do).
   3) In the next screen (entitled "ODBC Microsoft Access Setup") under the "Database" frame, press the "Select ..." button and select your Access Database, copied from the “empty.mdb” file. Press OK to quit the dialog.
h) Select your newly created data source file. Press OK to quit the dialog (the previous selection window may pop-up again, just confirm your choice).
i) The connection is now set up. You can test it using the "Test Connection" button at the bottom right. If it works, you can apply the changes by restarting the LO Server – simply press “Stop” and then “Start” in the Services panel of the Configuration Panel.

The Lexical Organizer is now fully installed. You should be able to manage user accounts with the Configuration Panel, and launch the Organizer applet by opening the Lexicon HTML file located in the Java Client folder.

**Compiling changes to the Java Source Code**

When compiling any changes you have made to the Java Source Code, it is first necessary to specify the classpath of the scrolladesktop.jar package. This can be done by opening a command prompt and typing:
“set CLASSPATH=C:\..\scrollabledesktop.jar” (… = classpath of the package). It is necessary to specify this classpath each time you restart your computer, otherwise the package will not be found and the compiler will produce error messages.

Once you have successfully compiled the Java code in the organizer directory (by going to the organizer directory and typing “javac *.java” at the command prompt), you must create an organizer package. This can be done by backing up to the Java directory at the command prompt and typing:

“jar cf organizer.jar organizer”

If you now copy and paste this organizer.jar file into the Java Client directory, so that the applet in the Lexicon HTML file will be able to access it. Any changes you have made to the code should now appear in the program when you launch the web page.
Appendix B
Java Source Code

In this appendix, I will describe the functions of the main .java files in the Lexical Organizer program. Where no description is given for a file, this means that the file deals with very low-level concepts which will not be of concern to a programmer making future alterations to the program.

**baseEntry.java**: this is an abstract class defining a theoretical baseEntry object, from which more concrete representations of Entries will inherit features. It is a very open, broadly-defined class, which provides an opportunity for different types of implementation to be introduced – for example, a panel allowing sound or graphics to be stored as part of the annotations to an Entry is one possible extension of this class.

**bigCellRenderer.java**

**ColorSet.java**: in this class, an array of colours is created.

**ContainerInterface.java**

**CustomBaseInternalFrame.java**: this is the most specific description of the Folder window which appears when a Folder is opened. It interacts with the FolderManager class to ensure that, when the window is opened and closed, the FolderManager is aware of this.

**EntryGroupMenuItem.java**: this object appears on the EntryPanel pop-up menu, under circumstances defined in the EntryPanel class. This class allows the user to add or remove an Entry to or from a group.

**defaultEntry.java**: this class describes the defaultEntry object, which is a specific type of baseEntry object. It is the panel that appears when the user double-clicks on
an Entry, and includes functions to display and alter the content of the Entry, to add and remove examples to the list, and to update all copies of the Entry.

**EntryPanel.java:** this is one of the most important classes in the Organizer program. It describes Folder and Entry objects, and defines the menu that appears when the user right-clicks on these objects, as well as the “Drag and Drop” methods allowing them to be moved around a workspace. Methods allowing the user to Copy, Remove, Rename, Test, Group and Link Entries and Folders are defined here.

**FolderEntry.java:** this class describes the FolderEntry object, which is a type of EntryPanel. It contains a CustomBaseInternalFrame object, which represents the window that appears when a Folder is opened. It includes a method “getMyFrame()” which calls on the orgClient method “populateFolder()” to add the Folder’s components to the window.

**FolderFrame.java:** this class defines, on a very basic level, the simple window which appears when a Folder is opened. The methods which populate this window with the components of the Folder are defined elsewhere.

**FolderManager.java:** the FolderManager object keeps track of what Folders are open and closed in the Organizer. The class includes methods to identify the Folder in which a given Entry, Link or Group is located.

**FolderPanel.java:** this is a very important class, as it defines what actions can take place within a Folder workspace. It includes the FolderPanel pop-up menu, and methods allowing the user to choose any of the options on the menu. These options include adding an Entry or a Folder, showing and changing the direction on a Link, and creating or removing a Link or a Group. The class also defines more “Drag and Drop” methods allowing Entries to be moved around on the workspace.

**GroupDiagram.java:** this class describes the container which appears around Grouped objects in a Folder. It ensures that all of the member Entries appear within the bounds of the GroupDiagram, and that the diagram is resized appropriately when new Entries are added or removed to the Group.
**GroupMenu.java:** the GroupMenu object appears on the FolderPanel pop-up menu. There is one for each Group existing in the Folder. The class includes features ensuring that the GroupMenu has the same colour as the Group which it represents, and methods allowing the Group to change colour and to be removed.

**LocalObjTransferable.java:** this object is responsible for ensuring that there is no recursion in the Folder hierarchy when Folder objects are created or moved.

**LineDiagram.java:** this class describes the LineDiagram objects which link Entries together within Folders. Instances of this class are created by the “linkEntries()” method in the FolderPanel class, which is itself invoked when the “Link Entries” option is chosen from the EntryPanel pop-up menu. It includes methods allowing the user to show and change the direction of the link, which are also called from methods in the FolderPanel class. It also ensures that links remain correctly displayed, even when linked Entries are moved on the workspace.

**LocalObjFlavor.java**

**LoginFrame.java:** this class describes the dialogue box which appears when the Organizer webpage is first opened, or when the user selects “Login” from the Options menu. It contains a method “connect()” which checks whether the username and password entered are valid (by calling on methods in orgClient.java), and can display a list of public organizers which can be viewed by guests.

**MoveToFrame.java:** this class describes the dialogue box which appears when the “Copy To” option is chosen from the EntryPanel pop-up menu. It includes a function “allFolders()” to find and list all of the existing Folders in the current Lexical Organizer, allows the user to select one, and then invokes the “copyto()” method in the EntryPanel class.

**organizerApplet.java:** this class describes the main Lexical Organizer window which appears when the Applet is launched. It defines the Options menu which appears on the toolbar and contains options such as Login, Logout, Public Organizer,
Change Password and Remove Organizer. The class includes methods to handle each of these options.

**orgClient.java:** this object will connect to the Server and communicate with it as the user opens and changes his/her Organizer, to ensure that the representation of the Organizer onscreen matches up with the information stored about it in the database. The class includes methods which tell the Server what changes the user is trying to implement (for example, removing an entry or adding a group), and receives messages from the Server indicating whether or not this change has been successfully recorded in the database. If it has not, the orgClient object posts a message to the screen informing the user of the error that has occurred.

**PanelDragContainer.java**

**scrollableDesktop.java:** this object is a basic version of the desktop workspaces which will be extended and used by Folders elsewhere in the program.

**serverMsgs.java:** this class defines a serverMsgs object, which is created within the orgClient class to handle messages from the Server. It reads the message from the Server, and then calls on the FolderPanel object to which the message refers and invokes the relevant method. For example, if the Server message relates to a link being created, the serverMsgs object calls the FolderPanel method “linkEntries”.

**TestDefFrame.java:** this class describes the dialogue box which appears when the “Test Definitions” option is chosen from the EntryPanel pop-up menu. It includes a function which finds the next Entry or Folder item contained within the Folder being tested, and proceeds to test the user on it, by inserting the appropriate text into the text fields. This object allows the user to choose whether to see a clue, skip a question, be given the question in their L1 or L2, and be tested on the sub-Folders of the current Folder.

**WordEntry.java:** this class describes the WordEntry object, which is a type of EntryPanel. The WordEntry object contains a DefaultEntry object, which displays the annotated information (meaning, notes and examples) associated with the Entry.
References


Bohike, O. (2002). *VTrain 4.0 (Vocabulary Trainer).* Available (01/05/2005): http://www.calico.org/CALICO_Review/review/vtrain00.htm


O'Rourke, B., & Schwienhorst, K. (1999). Learner databases and virtual worlds: Using computers to create collaborative learning environments.


