

Compansion: From Research Prototype to Practical Integration

Kathleen F. McCoy

*CIS Department, University of Delaware
Newark, DE 19716
mccoy@cis.udel.edu*

Christopher A. Pennington

*Applied Science and Engineering Laboratories
University of Delaware and duPont Hospital for Children
1600 Rockland Road, Po Box 269
Wilmington, Delaware 19899
penningt@asel.udel.edu*

Arlene Luberoff Badman

*Prentke Romich Company
1022 Heyl Road, Wooster, OH 44691
alb@prentrom.com*

(Received 17 March 1997; Revised 24 October 1997)

Abstract

Augmentative and Alternative Communication (AAC) is the field of study concerned with providing devices and techniques to augment the communicative ability of a person whose disability makes it difficult to speak or otherwise communicate in an understandable fashion. For several years, we have been applying natural language processing techniques to the field of AAC in order to develop intelligent communication aids that attempt to provide linguistically “correct” output while increasing communication rate. Previous effort has resulted in a research prototype called Compansion that expands telegraphic input. In this paper we describe that research prototype and introduce the Intelligent Parser Generator (IPG). IPG is intended to be a practical embodiment of the research prototype aimed at a group of users who have cognitive impairments that affect their linguistic ability. We describe both the theoretical underpinnings of Compansion and the practical considerations in developing a usable system for this population of users.

1 Introduction

Augmentative and Alternative Communication (AAC) is the field of study concerned with providing devices or techniques to augment the communicative ability of a person whose disability makes it difficult to speak or otherwise communicate in an understandable fashion. Whichever device is used, the communication rate

of the person using AAC is likely to be extremely slow, and using the aid will require a great deal of cognitive and physical effort. In addition, the listener will often be required to expend effort to understand the person using AAC. Some AAC users, in an effort to speed interactive communication, may use telegraphic language. Telegraphic language is very brief and concise; it generally conveys the content without the use of function words and word endings found in standard English. While telegraphic language will usually be “functional” and get the point across, its use often has adverse side effects. For instance, it may give communication partners the impression that the user is less intelligent because of their use of non-standard language.

For a number of years, we have been concerned with a particular type of communication aid that transforms telegraphic input into well-formed English sentences. This project was first motivated by considering linguistically mature users who would like their device to output well-formed sentences, but who often settle for telegraphic language because of physical and time constraints. For these users we envision a system that expands their telegraphic input yet does not interfere with their control over the dialogue. A research prototype of such a system has been developed which uses a Natural Language Processing (NLP) technique called *Compansion* (because it takes a COMPressed message and through expANSION, converts it into a well-formed sentence)(McCoy et al.1989b), (Demasco and McCoy1992), (McCoy et al.1994a). The reasoning used in this system will be described in detail later.

In continuing our investigation on how *Compansion* might be incorporated into a viable AAC device, we began to focus on a different population of users who might greatly benefit from the technique. Such a user would be one who not only has a physical disability which causes them to require an AAC device, but also has cognitive impairments which affect their linguistic ability. A system that expands telegraphic utterances for this population would not only provide more appropriate output, but might also be viewed as a language intervention tool that could provide feedback of well-formed sentences.

This new development effort involves (1) the Applied Science and Engineering Laboratories (ASEL) of the University of Delaware and the duPont Hospital for Children, and (2) the Prentke Romich Company (PRC is a well-known manufacturer of communication aids). We describe our ongoing effort to develop an intelligent language aid that would provide *Compansion*-like output in a practical communication system. The project combines the experience gained through the development of *Compansion* and other Natural Language Processing technology at ASEL, with the interface, access methods, and practical experience provided by PRC. In this paper we first describe the research prototype, *Compansion*, which provides the NLP theory for this research. The unique aspect of *Compansion* is its semantic reasoning which augments that used by current NLP systems. We point out difficulties with incorporating a technique such as *Compansion* in a standard AAC device, and indicate how our current effort is overcoming these difficulties. Next we introduce our new effort by first describing characteristics of our target population and indicating how these factors have affected implementation choices. We describe how the characteristics of our target population have allowed the pro-

cessing in our new system to be simplified and how the lexical information necessary for NLP reasoning has been handled. In addition we describe issues involved in tailoring the interface and the system functioning for this particular population of users.

2 The Companion Project Overview

The Companion project has been a long term effort to develop sophisticated processing methods that might lead to higher communication rates without requiring a great deal of cognitive effort on the part of the user (Demasco et al.1989), (McCoy et al.1989a), (McCoy et al.1989b), (McCoy et al.1990), (Demasco and McCoy1992), (McCoy et al.1994a).

While there has been some use of NLP techniques in augmentative communication prior to the Companion project, its use has been fairly limited. For instance, some syntactic knowledge has been used in the context of word prediction and language tutoring (Swiffin, Arnott, and Newell1987), (VanDyke, McCoy, and Demasco1992), (McCoy, Pennington, and Suri1996), (Newell et al.1992), (Wright et al.1992). Also, several systems at the University of Dundee such as PROSE (Waller, Broumley, and Newell1992), CHAT (Alm, Newell, and Arnott1987), and TALKS-BACK (Waller, Alm, and Newell1990), (Waller, Broumley, and Newell1992) use semantic (and pragmatic) knowledge to allow the user to access context-appropriate chunks of meaningful language with minimal effort. The major research emphasis with these systems has been the development of schemes which use NLP knowledge to access prestored pieces of text which are appropriate in the current conversation. In the Companion, however NLP techniques are used to attempt to process the user's spontaneous language constructions.

The Companion project was conceived as a rate enhancement technique which would be used in conjunction with a word-based language set. That is, the interface to the system would allow the user to select full words of input. The interface itself was not a focus of the work. Thus whether these words were selected from an electronic word board which the user navigated to find their desired word, or whether the word was actually selected via a sequence of icons which were transformed into words, is immaterial to Companion. The assumption made about the input interface of Companion was that each word of input would take a (basically) constant amount of time (regardless of how many characters were in the word). We call this constant amount of time a *keystroke*. Word endings (e.g., +s plural or +ed past tense) would require an additional keystroke to select.

Thus the focus of the research was on a "black box" which took the words input by the user and expanded them into full sentences to be output via an output interface (e.g., print or a speech synthesizer). See Figure 1. Companion potentially increases the communication rate by requiring fewer words to be selected (since it requires just the content words of the desired utterance to be input) and by eliminating the need for selecting morphological endings. Here we describe the processing in the prototype research system.

The processing is broken into three phases:



Fig. 1. Companion prototype concentrated on processing and assumed I/O interfaces.

1. A “word order” parser is used to group input words into sentence-sized chunks and to indicate each word’s part of speech (e.g., noun, verb). Modifiers (e.g., compound possessives, adjectives, and adverbs) are attached to the word they are most likely modifying at this stage of processing.
2. A “semantic” parser reasons about the main content words of each sentence-sized chunk and produces a semantic representation of the sentence.
3. A translator/generator takes the output of the semantic parser and generates an English sentence which reflects that meaning.

Consider the following example handled by the system:

Input: think red hammer break John

Output: I think that the red hammer was broken by John.

In this instance, the word order parser would note that **think** is a verb which may take either a sentential complement or an NP as its complement, and so attempts to find one of these constructions to the right of **think**. **Red** is an adjective which is “attached” to the noun following it as a modifier. Once this is done, the words to the right of **think** can be seen as a noun-verb-noun pattern. Because this pattern may constitute a sentence, it is sent off to the semantic parser for processing. The semantic representation of the embedded sentence is returned back to the word order parser. The word order parser then continues processing the top-level sentence, calling the semantic parser a second time to find the top-level semantic representation. Once this is found, the translator/generator is called to generate the sentence shown. The sentence generator used by the system is the Functional Unification Formalism (FUF) system (Elhadad1991). In the next two sections we elaborate on the word order parser and the semantic parser.

3 The Word Order Parser

The word order parser takes advantage of the syntactic regularity in telegraphic speech. It is based on a simple transition network grammar which encodes allowable (telegraphic) sentence patterns and a lexicon which indicates potential parts of speech for each word. The parser attempts to fit the given words into an acceptable pattern thus identifying each word’s part-of-speech, deciding how modifiers should be attached, and identifying embedded sentences. Consider the following:

Modifiers – We assume a “right attachment” of adjectives and a “right or left attachment” of adverbs. E.g., in the input `mary put book big table`, the parser would associate the adjective `big` with `table` (and not with `book`).

Compound Nouns – Two nouns following each other in the input can mean (a) The nouns serve in different thematic case roles. (b) The nouns should be conjoined. (c) The nouns should be interpreted as one possessing the other. Case (a) only occurs after the main verb of the sentence because that is the only place there can be two roles. Thus this case can be handled via sentence-level syntactic rules which expect two nouns to follow a verb such as `put` in the example above. In contrast, (b) and (c) require some semantic preference rules such as “like” things are conjoined (e.g., `john mary` would be assumed to mean `john and mary`), and animates possess inanimates (e.g., `john hat` would be assumed to mean `john’s hat`).

Embedded Sentences – As was shown in the example earlier, some verbs (e.g., `think`, `believe`) may take sentential complements. The word order parser identifies the complement and sends it off to the semantic parser as a single unit.

Word Sense Disambiguation – Through allowable sentence patterns, a word which could be either a noun or a verb (e.g., `watch`) may be disambiguated by its place in the pattern. In some cases, this is not enough so both alternatives would be sent for further processing.

4 The Semantic Parser

The semantic parser takes a set of words (tagged as nouns and verbs) from the word order parser and attempts to fit these items into a well-formed semantic structure. The semantic parser does “shallow” understanding in an attempt to build a valid semantic representation out of the lexical items it is given: consider the processing of the embedded sentence earlier (i.e., `hammer`, `break`, `john`). The parser tries to decide which noun in the input is most likely to be the agent. In this example, the parser must recognize that since `break` is the verb, `John` is likely to be the agent with `hammer` as the theme. In order to do this, the parser must have sufficient information about the semantic roles that various words can play, and information about the semantic expectations of particular verbs. The details of this information is explained below.

Traditionally semantic processors have relied on syntactic information in order to disambiguate the intended meaning of an utterance. Recognizing that input from the user is often not syntactically well-formed, some researchers (e.g., (Granger1983), (Fass and Wilks1983), (Weischedel and Sondheimer1983), (Jensen et al.1983), (Carbonell and Hayes1983), (Milne1986)) have attempted to augment this traditional approach to handle certain kinds of ill-formedness. These methods, however, work on the assumption that the user’s input will be primarily well-formed, but may contain some ill-formedness. Still other work has concentrated on ill-formed input of particular kinds which occur within a severely restricted semantic domain: e.g., naval messages (Marsh and Sager1982), (Marsh1983), (Marsh1984). McRoy

(McRoy1991) takes an approach similar to ours towards the problem of word sense disambiguation in that she recognizes several sources of knowledge that must be taken into account in semantic processing.

4.1 *Semantic Representation*

The output of the parser is a representation of sentence meaning relying on case theory (Fillmore1968), (Fillmore1977). Thematic case theory specifies that there is a small set of roles which noun phrases in a sentence may play with respect to the verb. While other people have specified different thematic cases, we rely on a set of thematic cases which have proven adequate for our purposes. AGEXP (AGent/EXPeriencer) is the object doing the action. For us, the AGEXP does not necessarily imply intentionality such as in predicate adjective sentences (e.g., “John” is the AGEXP in “John is happy”). THEME is the object being acted upon, while INSTR is the object or tool etc. used in performing the action of the verb. GOAL can be thought of as a receiver, which is not to be confused with BENEF, the beneficiary of the action. For example, in “John gave a book to Mary for Jane”, “Mary” is the GOAL while “Jane” is the BENEF. We also have a LOC case which captures the location in which the situation is taking place (this case may be further decomposed into TO-LOC, FROM-LOC, and AT-LOC), and TIME which captures time information (this case may also be further decomposed).

The focus of this paper is not on this particular logical form representation since we have attempted to choose a standard representation (see e.g., (Allen1995), (Palmer1984), (Hirst1987)). Rather, our focus is on the lexical knowledge and processing strategies necessary to derive such a representation given a set of uninflected content words. It is this problem to which we now turn.

4.2 *Knowledge Needed*

Traditional semantic interpretation components use a set of rules which act on a parse tree and generate the kind of representation described above. Typically these rules rely on a set of selectional restrictions which are based on the syntactic category (e.g., SUBJ, OBJ) of a component and the semantic type of the word involved. For example (Allen1995) describes a set of rules for the verb **break**. Typical rules indicate that if the SUBJ of **break** is animate, then it is the AGENT (e.g., as in **John broke the window with a rock**); the non-animate object is the THEME; the object of a with-preposition is the INSTRUMENT. Since we do not have the syntactic knowledge about SUBJs and objects available, we rely on some more robust preferences which are implicit in the example above.

First, our solution requires two knowledge hierarchies which are commonly used in standard natural language processing systems. These hierarchies capture information that can be associated with words. The first is the object hierarchy which embodies a classification of objects. The object hierarchy can be derived from WordNet (Miller1990), (Miller1995). Deriving this lexical knowledge is the focus of (Zickus1995) and (Zickus et al.1995). There is also a verb hierarchy whose top-

level nodes (material, relational, verbal and mental) are based on systemic grammar (Halliday1985). The purpose of this hierarchy is to capture semantic generalities between verbs such as what types of objects may fill the thematic cases.

The novel aspect of our semantic parser lies in the way that we indicate semantic preferences for the verbs. The parser uses a set of heuristic preferences to decide the best semantic interpretation for the given input words. These preferences are of two types: semantic case preferences (which can easily be attached to the verb hierarchy), and idiosyncratic case constraints (which are orthogonal to the hierarchical verb organization and thus must be attached to individual words). Both kinds of preferences refer to classes in the object hierarchy.

4.2.1 Semantic Case Preferences

In our system, associated with each verb are a set of preferences which indicate possible ways of filling out a thematic case frame with the input words (called an interpretation) to be rated against each other. We have three different kinds of semantic case preferences: Case Filler Preferences, Case Importance Preferences, and Higher-Order Case Preferences. Preferences ratings fall on a 1-4 scale. 4 signifies that the binding is exceptionally appropriate. 1 signifies that the binding is only appropriate in special cases.

Case Filler Preferences. The case filler preferences are most similar to what has previously been used in semantic representation in that they are used to indicate preferred fillers of a particular verb thematic case. Our case filler preferences are motivated by Preference Semantics (Wilks1975). Here, the kinds of objects that could fill a particular role are indicated, along with an indication of how good each possible filler type is. For example, the preference for filling the BENEFICIARY thematic case for most verbs is: ((human 3) (organization 2) (animate 2)). This basically should be read as, given the choice, a human should fill this role, but an organization or an animate object is also reasonable. No other types of objects may fill the BENEF case.

Case Importance Preferences. While the above preference tells us what kind of object may fill a particular thematic case role, it does not tell us anything about which cases are more important to fill for a particular verb. This is captured by the Case Importance Preferences. For example, with material verbs such as `hit`, it seems much more likely that the role of THEME will be filled than that of BENEF. To represent this, a higher value (3) is given as the preference of filling the THEME case, while a lower value (1) is given as the preference for filling the BENEF case.

Having this preference specified allows the system to handle input such as: [`John hit Mary`]. Many things can be the THEME of `hit`, including most physical objects. So physical objects would be given a “3” rating for filling the THEME role. On the other hand, humans are nearly the only kinds of things that can play the role of BENEF (for any class of verb) and these are given a “4” for this case filler preference. Taking nothing into account but the case *filler* preferences, the system

would prefer *Mary* filling the BENEf role (as in “John hit for *Mary*.”). The case *importance* preferences allow the “John hit *Mary*” reading to be preferred.

Higher-Order Case Preferences. While the above heuristics have proven to be quite useful, there are situations in which they fall short, since the scope of the heuristic is limited to a single thematic case role. However, the heuristic value of an interpretation can be affected by a combination of case bindings. Higher-order case preferences heuristics are intended to account for interactions between thematic cases and their fillers. These preferences must be applied to a full interpretation on the input words.

To see how multiple cases can interact, consider the following: if a non-human animate (e.g., *dog*) is the AGEXP of a material process, it is quite unlikely that an INSTRument is being used (e.g., *dogs* do not typically *eat with spoons*). Note, however, that *dogs* can *eat* and *people* can *eat with spoons*. This preference is captured in the parser by a rule which subtracts from the overall goodness of an interpretation when these conditions are found.

4.2.2 Idiosyncratic Case Constraints

The idiosyncratic case constraints are not used to heuristically compare interpretations, but are used to constrain what interpretations are considered. They differ from the semantic case preferences in that they are more definitive, and that they are placed directly on the verbs themselves and are not inherited down the verb hierarchy.

These constraints seem to be related to verb transitivity and are intended to capture the notion that some thematic cases on a verb are mandatory and others forbidden. These are idiosyncratic in nature because they do not seem to have any relationship to the general semantics of the verb hierarchy. Two semantically similar types may have different idiosyncratic features. For example, the parser encodes the words *eat* and *swallow* as semantically equivalent. However, *swallow* cannot typically have an instrument, while *eat* may.

4.3 Parser Logic

Essentially, the parser begins with an empty frame where all of the roles are represented, but unfilled. Based on information associated with the main verb (identified in the word order parser phase of the system) the parser removes roles from the empty frame that are forbidden for the verb. In addition, the semantic case preferences for the main verb are added to the resulting frame. The parser next considers all of the objects in the input string. Using the information from the case filler preferences, it recognizes what objects can play what roles with respect to the verb.

Every possible interpretation that holds to the specifications of the case filler preferences is tried. At this point, those interpretations which leave a mandatory role uninstantiated will be discarded. Next, each remaining interpretation receives

a heuristic rating based on the case importance and case filler preference values. A simple process is used to combine the scores into the entire interpretation's heuristic score. Higher-order heuristics may act on the resulting number yielding the final score. The interpretation with the highest heuristic value is considered the best guess of what the user intended.

4.4 Processing Example

Consider the input: [John break window hammer]. The word order parser would pass these words as a package into the semantic parser along with an indication that **break** is the main verb and the other words in the sentence are nouns. From the information associated with the verb **break**, we know the GOAL case is forbidden and the THEME and AGEXP cases are mandatory. Next, the parser locates the verb in the verb hierarchy to find the preferences.

The case importance and the case filler information from the verb **break** is listed below. The first entry indicates that 4 points are awarded if the AGEXP case is filled (a case importance preference), and the rest of that entry represents the case filler preference information. 3 points are awarded if AGEXP is filled by a communicator (either a human or an organization), 2 points are awarded for another animate object (e.g., a dog), and 2 points are awarded if this case is filled by an ergative object (e.g., a truck). The other cases are specified in the same way.

Case Name	Case Importance Preference	Case Filler Preferences
AGEXP	4	3 Communicator 2 Animate 2 Ergative-Object
THEME	3	4 Fragile 3 Object
INSTR	2	3 T-Box (hammer) 2 Tool (spoon) 1 Physical
BENEF	1	3 Human 2 Organization 2 Animate
LOC	1	3 Place (Boston)
TIME	1	4 Timed (yesterday)

At this point we have accumulated top-down information about the case frame interpretation by considering the verb. Next, in a bottom-up fashion, different interpretations will be considered as the nouns are fit into the case frame. The table below shows what point values will be awarded when each object is assigned each of its possible roles. For instance, John would be given a value of 3 for filling the AGEXP role, 2 for THEME, etc.

Word	Points for Case
John	AGEXP 3, THEME 2, INSTR 1, BENEF 1, GOAL <Forbidden>
window	THEME 4, INSTR 1
hammer	THEME 2, INSTR 3

With this information, the parser fits the objects into the cases in every possible way abiding by the mandatory and forbidden cases. The processing also requires that only one object fill any given role. Under these constraints, we generate two possibilities. The competing semantic representations are shown below and are analogous to “John broke the window with the hammer” and “John broke the hammer with the window” respectively. Notice that the first interpretation is given a much higher heuristic value and will thus be preferred by the system. As evidence of the flexibility of the system, note that if “window” had not been in the input stream the interpretation equivalent to “John broke the hammer.” would be preferred. The complex preferences do not apply to this example.

((63 DECL	((48 DECL
(VERB (LEX BREAK))	(VERB (LEX BREAK))
(AGEXP (LEX JOHN))	(AGEXP (LEX JOHN))
(THEME (LEX WINDOW))	(THEME (LEX HAMMER))
(INSTR (LEX HAMMER))	(INSTR (LEX WINDOW))
(TENSE PAST))	(TENSE PAST))

4.5 Additional Rate Enhancements

Default Verbs. In telegraphic input it is not unusual for certain verbs to be left out. In such cases the system has some ability to infer the verb. If, at the beginning of processing no main verb is found, the system considers two possible relational verbs: **have** and **be**. A preliminary test is performed to see if either or both of these are reasonable for the given input. If so, processing continues as if the verb were given by the user. With the input: [John paper] the parser will return a semantic parse consistent with: “John has the paper.” With the input: [John tired] the parser will return a semantic parse consistent with: “John is tired.”

Relational verbs are a desirable choice for the default because they are very frequent, and these verbs highlight the objects rather than the verb. It is possible to configure the system to use other verbs as defaults if desired.

Default AGEXP. The system also can infer the AGEXP of the sentence if no reasonable candidate is given in the input string. For example, with the input `tired`, the system will infer that the user is the AGEXP (e.g., “I am tired.”).

Question Words. The user may indicate a question is desired and the appropriate yes/no question words will be added. E.g., “Mary go store?” might generate “Did Mary go to the store?”. Questions interact with the AGEXP inferences in that when a question is involved the AGEXP defaults to “you” for questions. E.g., “tired?” produces “Are you tired?”.

5 Practical Problems with Companion

The Companion prototype is able to generate full sentences from a set of uninflected words input by a user of a word-based communication system. Its implementation demonstrated the feasibility of an NLP-based approach to AAC techniques. While the results of the system have been encouraging, Companion is faced with several practical problems which must be dealt with before the Companion technique can be incorporated into a viable communication device. Some of the problems are discussed here.

Unlimited vocabulary. Companion requires a large amount of information to be associated with each word. While we have been working on techniques to provide at least some of this information automatically (Zickus1995),(Zickus et al.1995) much information must be hand coded on the basis of intuition. As a result, it is currently nearly impossible to handle the problem of unrestricted vocabulary, especially if we consider that there are no automatic methods for deriving the necessary information from either on-line lexical resources or from corpus-based processing.

Complex Independent Subsystems. One of the original motivating applications for Companion was as a writing tool. Because the system processing was designed to handle complex written prose, the modules were each written to work independently and with little help from each other. For example, the semantic parser acts without regard for word order information since word order does not necessarily provide reliable information in prose (where constructions such as passives are much more common than in conversational settings). Because of this, the semantic processor must rely on complicated lexical knowledge that is difficult to acquire.

User Input Assumptions. The system made some assumptions about the input produced by the user. For example, the assumption is that the input would reflect the basic word order of the desired output if possible¹, that the default AGEXP

¹ This assumption is actually operative in the generation phase of the processing and is not discussed in this paper. Our informal experiments have indicated that this assumption may not be valid for users with poor English ability or cognitive impairments like aphasia.

should be I or you, and that relational verbs are likely to be left out. While some of these assumptions have been partially validated through experimental testing (McCoy et al.1994b), (Vanderheyden et al.1994), it is reasonable to expect that some of these assumptions may be dependent on the specific population using the device.

Telegraphic Input Assumptions. Related to the above, the system makes some assumptions about which words will be left out of the telegraphic input. In particular, the assumption is that function words may be left out, but content words will be included. It remains a question whether or not the system's rules are robust enough in practice to recover full sentences in every instance.

Interface. An aspect that has not been dealt with effectively is the particular interface through which the user interacts with the system. While the research prototype system has a particular word-based interface associated with it, research efforts have dealt with the front-end component as a separate "black box" that provides the system with words. We have not tackled the many issues involved with developing a front-end appropriate for a specific population of users. Notice that because the processing required by users is fairly involved (i.e., they must not only select words but must also accept/reject the expanded sentence produced by the system), the interface requirements are quite complex. Experience with the specific population using the device is required to develop an appropriate interface.

Machine Requirements. The research effort on the Companions system to date has been done on powerful Unix workstations in Lisp. Thus the system would probably not run well on current portable microcomputers as would be appropriate for use in real-world situations.

6 Opportunity for Companions: The Intelligent Parser Generator

In order to implement the Companions technique in a viable communication device, the above difficulties had to be addressed.² In the remainder of this paper we describe a joint effort between the University of Delaware and the Prentke Romich Company that attempts to overcome many of these difficulties by focusing on a particular population of users. We describe characteristics of our target population, and then indicate how we set about developing a viable communication device that incorporates Companions. While we found some characteristics of the target population have allowed us to greatly simplify the Companions-like processing, other aspects of the system (e.g., the user interface) must be carefully considered. We highlight our methodology for developing a system tailored to a particular population.

² Indeed these difficulties also make a formal evaluation of the research prototype problematic as an interface and full vocabulary coverage are crucial components for real-world evaluations.

6.1 Target Population

In considering a target population we looked for a group of users who would likely produce telegraphic input, would benefit from the expansion of that input into full sentences, and would reduce the burden of some of the other difficulties outlined above. We chose to consider a young population of users who have cognitive impairments that affect their expressive language ability. Whether a child with cognitive impairments is verbal or nonverbal, their expressive language difficulties may include the following (Kumin1994), (Roth and Casset-James1989): (1) short telegraphic utterances; (2) sentences consisting of concrete vocabulary (particularly nouns); (3) morphological and syntactical difficulties such as inappropriate use of verb tenses, plurals, and pronouns; (4) word additions, omissions, or substitutions; and (5) incorrect word order. While such children may have the ability to functionally communicate their needs and wants, intervention to assist them in their language production should be beneficial both from a social and an educational perspective.

In developing a device geared toward this population, several issues must be dealt with. These include:

lexical access – what is an appropriate method for providing such a user with access to the lexical items that they wish to communicate?

processing questions – how can the Companion-technique be implemented to efficiently run on current microcomputer hardware?

verification of user input assumptions – what kind of input will this population produce and what expansions are reasonable? E.g., how syntactically complicated must the input/output be?

user interface issues – what kind of interface is necessary for a user with cognitive impairments to be able to access the system. The user must not only have access to the lexical items, but must also be able to sift through the expansions provided by the system and have the system output their desired selection.

6.2 Lexical Access: Communic-Ease MAP[®]

PRC has expertise in providing lexical access to the population under study. The speech output communication aids that PRC designs for commercial use incorporate an encoding technique called semantic compaction, commercially known as Minspeak[®] (a contraction of the phrase “minimum effort speech”) (Baker1982), (Baker1984). The purpose behind Minspeak[®] is to reduce the cognitive demand as well as the number of physical activations required to generate effective flexible communication. It uses a language set (i.e., a set of selectable items) consisting of a relatively small set of icons that are rich in meaning and associations. These icons can be combined to represent a vocabulary item such as a word, phrase, or sentence, so that only two or three activations are needed to retrieve an item. This small set of icons thus allow access to a large vocabulary which is stored in the device. Since

they are rich in meaning, icons designed for Minspeak[®] can be combined in a large number of distinct sequences to represent a core lexicon easily.

The Minspeak[®] language set and processing was first utilized with PRC's Touch TalkerTM and Light TalkerTM communication aids (which united different physical interfaces with the icon encoding). With these Minspeak[®] systems, if icons on the overlay remain in fixed positions, once learned, they allow the individual using the system to find them quickly and automatically. This automatic processing was facilitated by the design of prestored vocabulary programs known as Minspeak[®] Application Programs (MAPsTM). In these programs a large vocabulary is prestored in a well-organized fashion using a logical, paradigmatic structure that greatly facilitates learning and effective communication.

One of these MAPsTM, Communic-EaseTM, contains basic vocabulary appropriate for a user chronologically 10 or more years of age with a language age of 5-6 years. Communic-EaseTM has proven to be an effective interface for users in our target population providing access to approximately 580 single words divided into 38 general categories. Most of these words are coded as 2-icon sequences. The first icon in the sequence (the category icon) establishes the word category. For example, the <SKULL> icon indicates a body part word, the <MASKS> icon indicates a feeling word, and the <APPLE> icon indicates a food word. The second icon denotes the specific word. For example, <MASK> followed by <SUN> produces the word "happy"; <APPLE> followed by <APPLE> produces the word "eat".

In addition to the words which are accessed via the icon sequences, Communic-EaseTM contains some morphology and allows the addition of endings to regular tense verbs and regular noun plurals. However, to accomplish this, additional keystrokes are required. It is also possible to spell words that are not included in the core vocabulary. In practice, however, users with either slow access methods or poor language ability tend to produce telegraphic messages consisting of key word sequences.

Notice that the Communic-EaseTM MAP provides a limited vocabulary of the words commonly needed by this population of users. By using this as a lexical access method, the majority of the vocabulary words are known in advance. Thus, in addition to providing the user with a core vocabulary, the Communic-Ease MAP limits the lexical items a user can select. As a result, the system need not store lexical information about an unlimited set of words. The knowledge needed for the words can be coded in advance.

We turn to the problems of identifying the specific input/output requirements of the device.

6.3 Design Methodology: User Centered Design

Our methodology in this collaborative effort is to design a system that is geared toward the specific user population. Thus, we have set out to validate our assumptions about the user input and output requirements and to tune the user interface to the population. Our system functionality has been determined by a collection of transcripts from Communic-EaseTM users. We have collected both raw keystroke

data (so that we can establish the range of input we expect from the population) and keystroke data from videotaped sessions where interpretations of the keystroke data are provided by a communication partner. This data allows us to ensure the output from the system is in fact appropriate.

Collection of such data has allowed us to:

- validate expected sentence structures
- validate the expectation of limited vocabulary
- validate input assumptions

In addition, we plan to validate our interface requirements on the basis of iterative user testing. The interface will be developed so that it can be customized to the specific needs of particular users.

7 Prototype Development

7.1 Envisioned System

The envisioned system combines the PRC LiberatorTM system (which provides both a physical interface and low level processing) that runs a modified Communic-Ease MAPTM (which provides a standard vocabulary and its access method) and an intelligent parser/generator (which provides the Compansion-like processing). The input from the user will be through the LiberatorTM keyboard (most of whose keys contain the icons which are transformed into words via the Communic-Ease MAP). The user will receive feedback through an Interface Display. One part of the display will show the transformed sentences which the user may select to be “spoken” by the system. A simplified block diagram of the system is shown in Figure 2.

The LiberatorTM Overlay/Keyboard accepts user input via a variety of methods (e.g., direct selection), and can also limit user choices via Icon Prediction. With Icon Prediction only icons that are part of a valid sequence are selectable. The user selects icon sequences that are transduced into words or commands according to the Communic-Ease MAPTM. In normal operation, icon labels and the transduced words are sent to the interface display to give the user feedback (words may also be spoken incrementally).

In the proposed system, these components are supplemented with an intelligent parser/generator (IPG) that is currently under development at ASEL. IPG is responsible for generating well-formed sentences from the user’s selected words and is a simplified version of Compansion. In IPG the various modules of Compansion have been collapsed due to the reduced processing needed to handle the simple sentence structures produced by this population (as found in our data collection). The parser/generator is based on a transition network grammar which relies on both syntactic and semantic tests. The parser generator works by incrementally parsing the input along parallel lines where each parse maintains an expanded version of the input. The partial parallel parses maintained by IPG also provide further constraints on the Icon Prediction process. For example, if the user selected “I have red,” the system might only allow icon sequences for words that can be described by a color (e.g., shoe, face).

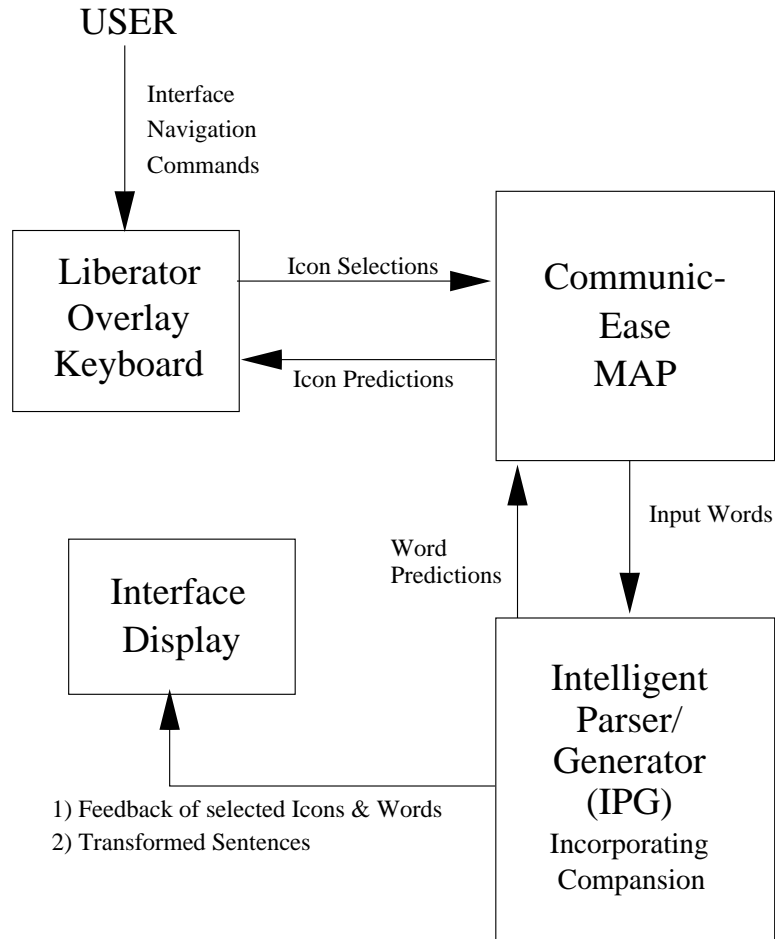


Fig. 2. Block Diagram of Envisioned System (speech and print output is not shown).

Our transcript study has revealed that the simpler sentence structures employed by this population requires less complicated processing on the part of the system in order to transform the telegraphic sentences into full sentences. The particular transformation rules encoded in the system have been motivated by our study of current Communic-EaseTM users. This processing has been incorporated into a set of transformation “rules”. One major advantage of describing transformations as a set of rules is that it is relatively easy to parameterize the system to affect its overall strategy. For example, a clinician or teacher could disable any of the transformation rules depending on the particular user’s abilities or educational goals. For some rules it will also be possible to specify information about how the rule should be applied. For example, preferences for determiner inferencing could be adjusted (e.g., prefer “**the**” over “**a/an**”).

In some situations IPG may find multiple interpretations of the user’s input. These interpretations will be ordered and presented to the user for selection.

7.2 Interface Issues

Beyond the basic operation described above, there are a number of interface issues that need to be resolved before a completed product is developed. These issues are being explored in early system prototypes with iterative user testing. Because it is likely that different users will have different requirements (especially if the system is used by a larger population than just the target population), our methodology is to develop the interface and system function with a series of parameters that can be set to customize the system's behaviors. This will allow the system to be tuned to the needs of particular users.

The first issue of concern is how the intended population can best interact with the system when multiple sentences are generated from an input. As mentioned above, IPG can order output sentences according to a variety of rules. However, if the user has the cognitive ability to select their desired sentence, it will be important to determine the best way to (1) present multiple options to the user, and (2) allow users to select from a list of possible choices. A number of possibilities exist including providing a list on the display screen, offering each sentence one at a time with some user-activated key to request the next choice, and providing the list in an auditory fashion. Options such as these will be explored during prototype development. We discuss how some of these options might be presented in the section on Presentation Layout below.

A second major interface issue revolves around incremental versus non-incremental processing. In incremental processing, the system would attempt to transform input on a word-by-word basis. For example, if the user selected the word "cat," the system might expand it to "The cat" immediately before the next selection is made. Presumably if this was not the intended expansion the user could indicate his/her displeasure before the next selection was made. In contrast, non-incremental processing would wait for the entire sentence to be entered and then produce an output sentence(s). For example, the entire string "cat hungry" would be input before being transformed into a sentences such as "The cat is hungry" or "A cat is hungry".

While incremental processing may appear to have advantages for the system (since it would cut down on the number of possibilities the system must consider) and for the user (since it would produce fewer possibilities in the end), it is likely that it will prove too cognitively taxing for the target population. In particular, it is unlikely that such users will be able to keep the sentence that they intend in their head (in order to select it word-by-word), select the appropriate lexical items, and be able to discriminate whether or not the system's expansion is correct, all at the same time. Because of the cognitive load involved in incremental processing, our initial prototype is being developed for non-incremental processing. This decision could likely be different given a higher functioning population of users.

7.2.1 Editing Functionality

Another issue that must be addressed is the editing permitted by the system. (This is even more crucial when incremental processing is considered.) The editing capabilities of the system will be parameterized to fit the needs of the user. For example, the system might allow deletions only at the end of the string the user has selected. On the other hand, higher functioning users might choose “full editing” capabilities that would allow additions/deletions from the middle of the currently selected string (as an example).

Tied in with editing issues are concerns raised when the user is permitted to spell any word. Spelling, of course, introduces the possibility of unknown words (and misspellings). Note that unknown words are a serious difficulty for the intelligent aspects of the system that require part-of-speech and some semantic information on the words. As mentioned before, the LiberatorTM system provides Icon Prediction. When Icon Prediction is used, the user is encouraged to select only valid sequences (because only these key sequences are allowed to be selected). Icon Prediction has proven very useful for users, especially when they are still learning the appropriate icon sequences for their desired vocabulary. One method of handling misspellings is to force only valid words by expanding “icon prediction” into the spelling mode (using, of course, a fairly substantial dictionary). The intuition is that the system would only allow sequences of letters that matched some element of the dictionary. This would preemptively restrict any misspellings that did not result in a word in the dictionary. However, it would not prevent the user from typing inappropriate words – i.e., a word that is actually in the dictionary but not the word intended by the user. Thus, if Icon Prediction is used in spelling mode, the system must have the ability to process inappropriately used words.

If Icon Prediction is not used in spelling mode, then the system must be able to handle misspellings. One method for doing so would be to assign some default part-of-speech (e.g., noun) and very general semantic information to these words. The effectiveness of this solution must be tested with users. Of course, the original input would be made available as an output choice when either no expanded sentences were generated or when the “heuristic score” for each generated sentence was below a preset confidence level. This behavior can also be set as a default parameter so that the original input is always one of the choices presented to the user.

7.2.2 Presentation Layout

The new system must accommodate a list of generated sentences from which the user can choose the desired sentence. The current LiberatorTM LCD display contains a “buffer” that shows each current icon/key selection (Icon Buffer) and a second “buffer” for message construction and editing (Text Buffer). In the example below, <MASKS> is an icon that has “emotions” as one of its semantic associations. <SUN> is often used to indicate a positive concept. Together they represent the Minspeak[®] encoding for the word *happy*.

I happy	Text Buffer
<MASKS> <SUN>	Icon Buffer

With the integration of IPG we will replace the LiberatorTMLCD display with a larger display that will add a third buffer that will be used to show generated sentences (Gen Buffer). The display below illustrates one of a number of possible configuration layouts.

I happy	Text Buffer
I am happy I was happy I happy	Gen Buffer
<MASKS> <SUN>	Icon Buffer

Each possible layout is determined by setting a group of parameters associated with presentation options. Some of the more important parameters include the size of the Gen Buffer, scrolling behavior, highlighting options, and whether or not the Text Buffer is replaced by the current choice. Audio feedback can also be provided since users who are pre-literate or have visual difficulties may benefit from having each of the potential sentences spoken on a private audio channel. This is often referred to as audio scanning.

In the simplest scenario the display collapses the Text Buffer and the Gen Buffer (size=0, replacement=T). When the input sequence is complete, the contents of the Text Buffer is replaced by the most highly rated expanded sentence. The user could then cycle through the other possible expanded sentences one at a time until their desired sentence is found.

I am happy	Text Buffer
	Icon Buffer

We anticipate that this may be a useful configuration for users who are not comfortable with selecting from among a group of possible alternatives. Part of the evaluation process will include determinations of this sort.

7.3 Development Methodology

The prototype system combines the PRC's LiberatorTM platform and Communic-Ease MAPTM with ASEL's current-generation intelligent parser/generator. In the implementation the LiberatorTM will function primarily as the user's keyboard and a tablet-based portable computer will contain the parser/ generator. The portable

computer will also augment the LiberatorTM's LCD display and provide easy modification of the user interface. The two systems will be connected via an RS-232 or IR link. This strategy allows for rapid initial prototype development.

The intelligent parser/generator includes three major software components. The parser/generator module is written in C++. The system lexicon will include all of the words contained within the Communic-Ease MAPTM along with a variety of words that users may spell. This knowledge base will contain semantic knowledge such as noun categories and properties as well as morphological properties such as word endings. Finally, syntactic knowledge is captured in system grammars that are based on an Augmented Transition Network formalism. The network grammar uses both syntactic and semantic properties of the input words.

Our project methodology is to develop and test the robustness and usability of the system in phases. The parser has been developed in C++ and is being refined and tested as other parts of the project progress. A core grammar has been created and is being revised and enhanced to handle a larger variety of structures. Current lexicon efforts involve expanding the number of entries beyond the basic Communic-Ease vocabulary and adding the necessary semantic knowledge. The first version of the Windows-based user interface has recently been completed and is now being evaluated internally.

The present system prototype consists of a Liberator attached to a Pentium-based desktop PC running the user interface with Windows NT and a software-based text-to-speech synthesizer. When the final details of the user interface and the other software components are worked out, the completed prototype will be implemented on a tablet-based PC and will be field tested with actual augmentative communicators. Because the user interface has been constructed with flexibility in mind (i.e., through parameterization), it will also be evaluated through iterative testing. This will help us determine the most appropriate interface configuration (or possibly set of configurations) for the targeted user population; however, we would still retain the ability to make minor adjustments to suit the needs of a particular user.

Several evaluations of the completed prototype system are planned. For instance, a theoretical evaluation of the grammar coverage is ongoing. As has been stated, we have collected key selections from current users of the Communic-Ease MAPTM. In some situations, we also have an interpretation of those keystrokes provided by the communication partner in a videotaped session. These video sessions have been transcribed and aligned with the keystroke data. While some of this data is being used to develop the grammar, we have set aside a portion of it to be used for testing purposes. This test data will allow us to test the system's grammar in several ways. First, the robustness of the grammar can be tested by determining the number of completed input utterances found in the collected data that can be handled by the grammar. Second, the appropriateness of the grammar can be tested by determining how often the grammar's output matches the interpretation provided by the communication partner in the video sessions. Because we have much more keystroke data than transcribed video data, we also plan a test of grammar

appropriateness by comparing the output of the grammar with that generated by a human faced with the same sequence of words.

In addition to the theoretical grammar testing described above, we also plan an informal evaluation of the usability of the system. We plan to iteratively refine the interface by doing usability studies of our prototype with current users of the Communic-Ease MAP^{TM3}.

8 Conclusions

This paper describes a research effort to bring NLP techniques to practical AAC devices. Important features of the effort include a multidisciplinary team with technical expertise in various areas including NLP and clinical expertise with the target population. This effort focuses on a particular user population which enables us to constrain the system processing sufficiently to make the NLP application feasible. Our effort involves designing the system around the specific needs and abilities of the particular population.

While the Compansion system prototype suggests that NLP techniques can usefully be incorporated into an AAC system, much remains to be done. Here we have tried to point out how specific characteristics of the user population must be considered. Careful selection and study of the characteristics of the user population is crucial. Several aspects of the system must be precisely tuned to ensure system reliability and usability.

9 Acknowledgments

This work has been supported by a Small Business Research Program Phase I Grant from the Department of Health and Human Services Public Health Service, and a Rehabilitation Engineering Research Center Grant from the National Institute on Disability and Rehabilitation Research of the U.S. Department of Education (#H133E30010). Additional support has been provided by the Nemours Foundation.

The authors would like to thank Patrick Demasco, David Hershberger, and Clifford Kushler for their collaboration on the project. In addition we thank John Gray for his discussions and implementation of many of the C++ aspects of the system, and Marjeta Cedilnik for her work on the grammar (and transformation rules). We would also like to thank the paper reviewers for their many thorough and useful comments.

References

- Allen, James. 1995. *Natural Language Understanding, Second Edition*. Benjamin/Cummings, CA.

³ One aspect these studies may shed light on is whether or not users in the population under study can in fact select their desired sentence when a list of possibilities is presented to them.

- Alm, Norman, Alan Newell, and John Arnott. 1987. A communication aid which models conversational patterns. In Richard Steele and William Gerrey, editors, *Proceedings of the Tenth Annual Conference on Rehabilitation Technology*, pages 127–129, Washington, DC, June. RESNA.
- Baker, B. 1982. Minspeak. *Byte*, page 186ff, September.
- Baker, B. 1984. Semantic compaction for sub-sentence vocabulary units compared to other encoding and prediction systems. In *Proceedings of the 10th Conference on Rehabilitation Technology*, pages 118–120, San Jose CA. RESNA.
- Carbonell, Jaime G. and Philip J. Hayes. 1983. Recovery strategies for parsing extragrammatical language. *American Journal of Computational Linguistics*, 9(3-4):123–146.
- Demasco, P., K. McCoy, Y. Gong, C. Pennington, and C. Rowe. 1989. Towards more intelligent AAC interfaces: The use of natural language processing. In *Proceedings of the 12th Annual Conference*, New Orleans, Louisiana, June. RESNA.
- Demasco, Patrick W. and Kathleen F. McCoy. 1992. Generating text from compressed input: An intelligent interface for people with severe motor impairments. *Communications of the ACM*, 35(5):68–78, May.
- Elhadad, Michael. 1991. FUF: The universal unifier user manual version 5.0. Technical report, Columbia University, Computer Science Department.
- Fass, Dan and Yorick Wilks. 1983. Preference semantics, ill-formedness, and metaphor. *American Journal of Computational Linguistics*, 9(3-4):178–187.
- Fillmore, C. J. 1968. The case for case. In E. Bach and R. Harms, editors, *Universals in Linguistic Theory*, pages 1–90, New York. Holt, Rinehart, and Winston.
- Fillmore, C. J. 1977. The case for case reopened. In P. Cole and J. M. Sadock, editors, *Syntax and Semantics VIII: Grammatical Relations*, pages 59–81, New York. Academic Press.
- Granger, Richard H. 1983. The NOMAD system: Expectation-based detection and correction of errors during understanding of syntactically and semantically ill-formed text. *American Journal of Computational Linguistics*, 9(3-4):188–196.
- Halliday, M. A. K. 1985. *An Introduction to Functional Grammar*. Edward Arnold, London England.
- Hirst, Graeme. 1987. *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge.
- Jensen, K., G. E. Heidorn, L. A. Miller, and Y. Ravin. 1983. Parse fitting and prose fixing: getting a hold on ill-formedness. *American Journal of Computational Linguistics*, 9(3-4):147–160.
- Kumin, L. 1994. *Communication Skills in Children with Down Syndrome: A Guide for Parents*. Woodbine House, Rockville, MD.
- Marsh, E. 1983. Utilizing domain-specific information for processing compact text. In *Proceedings of the Conference on Applied Natural Language Processing*, pages 99–103. ACL.
- Marsh, E. 1984. A computational analysis of complex noun phrases in Navy messages. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, pages 505–508, Stanford University, Ca., July. Coling84.
- Marsh, E. and N. Sager. 1982. Analysis and processing of compact text. In *Proceedings of the 9th International Conference on Computational Linguistics*, pages 201–206. Coling82, July.
- McCoy, K., P. Demasco, Y. Gong, C. Pennington, and C. Rowe. 1989a. A semantic parser for understanding ill-formed input. In *Proceedings of the 12th Annual Conference*, pages 145–146, New Orleans, Louisiana, June. RESNA.
- McCoy, K., P. Demasco, Y. Gong, C. Pennington, and C. Rowe. 1989b. Toward a communication device which generates sentences. In *Proceedings of the 12th Annual Conference*, New Orleans, Louisiana, June. RESNA.

- McCoy, K., P. Demasco, M. Jones, and C. Pennington. 1990. Applying natural language processing techniques to augmentative communication systems. In *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki, Finland, August. COLING.
- McCoy, Kathleen F., Patrick W. Demasco, Mark A. Jones, Christopher A. Pennington, Peter B. Vanderheyden, and Wendy M. Zickus. 1994a. A communication tool for people with disabilities: Lexical semantics for filling in the pieces. In *Proceedings of the First Annual ACM Conference on Assistive Technologies (ASSETS94)*, pages 107–114, Marina del Ray, CA.
- McCoy, Kathleen F., Wendy M. McKnitt, Christopher A. Pennington, Denise M. Peischl, Peter B. Vanderheyden, and Patrick W. Demasco. 1994b. AAC-user therapist interactions: Preliminary linguistic observations and implications for Companion. In Mary Binion, editor, *Proceedings of the RESNA '94 Annual Conference*, pages 129–131, Arlington, VA. RESNA Press.
- McCoy, Kathleen F., Christopher A. Pennington, and Linda Z. Suri. 1996. English error correction: A syntactic user model based on principled mal-rule scoring. In *Proceedings of User Modeling '96*, Kailua-Kona, HI.
- McRoy, Susan. 1991. Using multiple knowledge sources for word sense discrimination. *Computational Linguistics*, 17(3), September.
- Miller, G. A. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4).
- Miller, G. A. 1995. A lexical database for English. *Communications of the ACM*, pages 39–41, November.
- Milne, Robert. 1986. Resolving lexical ambiguity in a deterministic parser. *Computational Linguistics Journal*, 12(1):1–12.
- Newell, Alan F., John L. Arnott, William Beattie, and Bernadette Brophy. 1992. Effect of "PAL" word prediction system on the quality and quantity of text generation. *Augmentative and Alternative Communication*, pages 304–311, December.
- Palmer, Martha S. 1984. *Driving Semantics for a Limited Domain*. Ph.D. thesis, University of Edinburgh. Chapter 2: Previous Computational Approaches to Semantic Analysis.
- Roth, F. P. and E. Casset-James. 1989. The language assessment process: Clinical implications for individuals with severe speech impairments. *Augmentative and Alternative Communication*, 5:165–172.
- Swiffin, A. L., J. L. Arnott, and A. F. Newell. 1987. The use of syntax in a predictive communication aid for the physically impaired. In Richard Steele and William Gerrey, editors, *Proceedings of the Tenth Annual Conference on Rehabilitation Technology*, pages 124–126, Washington, DC, June. RESNA.
- Vanderheyden, Peter B., Christopher A. Pennington, Denise M. Peischl, Wendy M. McKnitt, Kathleen F. McCoy, Patrick W. Demasco, Hans van Balkom, and Harry Kamphuis. 1994. Developing AAC systems that model intelligent partner interactions: Methodological considerations. In Mary Binion, editor, *Proceedings of the RESNA '94 Annual Conference*, pages 126–128, Arlington, VA. RESNA Press.
- VanDyke, Julie, Kathleen McCoy, and Patrick Demasco. 1992. Using syntactic knowledge for word prediction. Presented at ISAAC-92. Abstract appears in *Augmentative and Alternative Communication*, 8.
- Waller, A., N. Alm, and A. Newell. 1990. Aided communication using semantically linked text modules. In J. J. Presperin, editor, *Proceedings of the 13th Annual RESNA Conference*, pages 177–178, Washington, D.C., June. RESNA.
- Waller, A., L. Broumley, and A. Newell. 1992. Incorporating conversational narratives in an AAC device. Presented at ISAAC-92. Abstract appears in *Augmentative and Alternative Communication*, 8.

- Weischedel, Ralph M. and Norman K. Sondheimer. 1983. Meta-rules as a basis for processing ill-formed input. *American Journal of Computational Linguistics*, 9(3-4):161–177.
- Wilks, Y. 1975. An intelligent analyzer and understander of English. *Communications of the ACM*, 18(5):264–274.
- Wright, Andrew, William Beattie, Lynda Booth, William Ricketts, and John Arnott. 1992. An integrated predictive wordprocessing and spelling correction system. In Jessica Presperin, editor, *Proceedings of the RESNA International '92 Conference*, pages 369–370, Washington, DC, June. RESNA.
- Zickus, Wendy M. 1995. A software engineering approach to developing an object-oriented lexical access database and semantic reasoning module. Technical report 95-13, Department of Computer and Information Sciences, University of Delaware, Newark, DE.
- Zickus, Wendy M., Kathleen F. McCoy, Patrick W. Demasco, and Christopher A. Pennington. 1995. A lexical database for intelligent AAC systems. In Anthony Langton, editor, *Proceedings of the RESNA '95 Annual Conference*, pages 124–126, Arlington, VA. RESNA Press.