# Test of a Communication Aid With Stored Text.

Parts of this text were printed in the International

Journal of Rehabilitation Research, 23, June 2000,

pg 139-144 as a short article. Among many other

mistakes the Abstract was left out. Much of this text

is now outdated.

## Introduction

I have constructed and evaluated a text based communication aid with two healthy test subjects using morsecode and synthetic speech. Results were published in Int J. of Rehabilitation Research, June 2000, in a short article. The abstract was left out and is included underneath. Also, many appendices were left out and are included in this text, intended as a contribution to the art of designing communication aids, based on a limited but intense experience and uniquely for those with in depth interest. Appendix 4 is also shown separately on my homepage.

Much literature is mentioned but I realise very well that others read something quite different in those same sources. This is related to one of the realities of designing interactive computer applications, one encounters very different issues that can not all of them be covered in depth and too many people have strong opinions on many of those issues. I apologise if this text could be more readable or if it repeats itself.

I would like to thank my wife Annie and children Armelle and Arnoud for their support. Of the many friends, researchers and caregivers who have shared their time with little or no material compensation I name only a few: Deborah Anthis, Ann Baker, Oscar Estévez Uscànga, Vianney de Jong, W.J.M. Levelt, Cox Schouten, Gerrit van der Veer. Six anonymous referees, Horabail S. Venkatagiri, Jefferey Higginbotham, Marilyn J. Buzolich and John Todman commented on (parts of) manuscripts with helpful criticisms. Test subjects were Greta Deroost, Petra Sleeman, Allaert Troost, Esther Parigger and Teresa McKee. A speech card was made available by Erik Brouwer and Ben van Mastbergen from Het Dorp. The library of University of Amsterdam in Het Bungehuis, faculty of general linguistics, was much appreciated. Minspeak$^{TM}$ is a proprietary name of Prentke Romich Corporation and so is the Liberator 2. EZ-Keys is a proprietary name of Words+.

Amsterdam, June 1998/june 2000

## Abstract

This text considers a communication aid that employs computer technology to construct and voice electronically stored texts. Typing is combined with four rate enhancements:

- upper case characters mean frequent words
- a menu in the left margin helps to quote
- word prediction and phrase prediction help to construct and to select messages.

Phrase prediction uses the first characters of the words of stored phrases and word prediction employs recency marking to repeat words easily and to avoid looking at the screen.

To test, two non-disabled subjects both communicated with the same aid during eighteen hours. The system itself was easy to learn but learning to communicate with it required a serious effort. Access by Morse Code was used to test if cognitive load is not too big. In five conversations with a total length of 251 minutes this resulted in ten characters voiced for thirteen dits and dahs (SD 0.028). In 24 out of 25 data-points, rate enhancements selected more than 10% of the spoken text. Several conceptually simple rate enhancements were combined in realistic communications.

## One Hypothesis and Many Questions

Some subjects are speechless, able to read and cognitively unimpaired. They may suffer from conditions such as amyotrophic lateral sclerosis (ALS) or cerebral palsy and often have other motor impairments. Speech disabled subjects type at rates far less then 200 words per minute (wpm) of fluent speech or about 30 wpm needed to maintain a slow conversation (Foulds, 1980). This rate problem seriously impedes conversation and can be very frustrating.

Computer programs have been constructed that allow users to select stored material with few key-strokes and voice it with synthetic speech (Alm, Arnott and Newell, 1992). To the theoretically inclined, many interrelated questions now arise: Which functions should such software offer, what material should be stored, how stable will it be, what roles might stored material have apart from speeding up conversation, what other technologies do users of such systems need, how can these aids be tested, with whom? To answer such questions a prototype was constructed that combines stored text with four different rate enhancing techniques. An overview of its design is presented, some new techniques are described and part of its testing.

## Design

Satisfactory use has been reported of aids that store large amounts of data and model the pragmatics of "effective social communication" (Todman, Elder and Alm, 1995; Todman and Lewins, 1996). A different approach starts not with future conversations but with actual texts and asks what will users want to do with it?

Four groups of functions can be discerned:

1. Functions to compose and to vocalize every possible text.
2. Functions to rapidly access few texts.
3. Functions to choose from many texts.
4. Functions to organise texts in flexible ways.

Editing, typing and word prediction should address the first, margin menu and redefined upper case characters should address the second group. Phrase prediction and a special kind of word prediction are present simultaneously for the third. It is hoped that paragraphs to store text plus special functions to navigate between paragraphs are adequate for group four. As regards to learning to communicate with an aid, much effort was invested in a clear manual and a clear presentation on the screen. To visualize the prototype figures 1 to 5 are a good starting point.

## Can stored material be tested?

Subtle techniques do not guarantee effective communication. Users of aids with stored material have to remember a significant amount of text, have to reformulate their communicative intent and require an unclear amount of training. It is hard to reduce the cognitive load caused by text prediction and by complex user interfaces. Also, it remains doubtful whether any collection of stored material allows satisfactory conversations unless combined with typing.

Readers may wonder if the hypothesis that stored material is helpful for the rate problem might be falsified. Several aids are currently tested in special schools (A. Waller, F. Booth et al. 1998), but not in order to falsify. To motivate subjects to communicate with computers necessarily introduces observer bias. As convincing tests of the hypothesis necessarily involve people, this can not be helped. If non-disabled users continue to communicate with rate-enhancements even when their cognitive load is increased artificially, we will have failed to falsify.

A hard test for any aid consists in combining rate enhancements with an additional input handicap such as Morse Code (MC). The reason is that for a long learning period MC puts additional cognitive load on learners. If rate enhancements are used nevertheless, cognitive load is acceptable; if communication is satisfactory stored material does the trick; if all functions are used the aid contains no superfluous functions. All these are the case (see below).

## Method

Two non-disabled subjects with interest in communication impairment learned the system "to find out about communicating with two aids". They spent eighteen hours learning and talking with two computers using stored material that was available from earlier experiments. At first, subjects seemed completely absorbed by the machines' suggestions and by the meanings of upper case characters. It took about ten hours before they anticipated on the behaviour of the program. Using the system was fun, subjects easily understood each others' problems to communicate and succeeded having simple conversations. Apart from agreeing on a subject first, the content was not interfered with. See Table 1 for a logfile after eighteen hours of practice.

One subject (EP) took part in a continued study with the author (JV). EP adapted her database to her own taste and used single finger typing. JV had used his prototype for about sixty hours. Table 2 shows he used all mechanisms when communicating with a qwerty keyboard. He learned MC and exercised about twentyfive hours before he could combine it with text prediction. At that time, he keyed MC at a rate of thirty characters per minute. We had two more sessions of free conversations. On several occasions, JV closed his eyes to concentrate on coding MC. This clearly complicated turn taking and we experienced some frustratingly imprecise interactions. Both of us reacted with more stored material, short sequences to ensure we always had a (new) story to tell.

Rate enhancements were used for diverse communicative functions. For instance, the margin menu was used to quote stored material, to quote formulaic utterances, to remind of an earlier topic and for turn-taking. Upper case letter codes were found useful to comment as well as to construct new phrases. As expected, we needed some gestures to clarify our intent. The quality of our interaction improved rapidly and we entered a more formal phase with conversations of 45 minutes speaking by computer only.

## Results

In five conversations that all lasted about 45 minutes we experienced no major communication breakdown and generally felt understood. After each conversation we wrote down our experiences and usually agreed on what had happened. We were amazed by the perceived efficacity of stored texts.

Logfiles were made to document keystrokes per mechanism. See Tables 3 and 4 for logfiles from all five sessions added up, a total of 251 minutes. The margin menu and upper case characters selected much text and all rate enhancements were used. Average (combined) *conversational rate* was (3201+1621)/251 = 19.2 words per minute. Individual rates cannot be computed from this data as we did not measure how much time each subject spent typing. Average *keystroke reduction* was ((18073+8214)-(7179+3529))/(18073+8214) = 59.2%, including arrows and other keys for cursor movement only, otherwise 73.8%. In five conversations with MC, 10713 dits and dashes sufficed to speak 8214 characters: 1.30 dit per character (SD 0.028). Fractions were computed of the length per rate enhancement divided by the length per session (MC only). 24 times out of 25 this figure was above 10% ($p<0.01$, token-test). The sole exception was word prediction that scored only 9.79% in the third session. Even when selections by menu are discarded, to leave out repeating, only (2120+709)/(18073-11182+8214-1890) = 21.4% of the spoken length was typed in. See Tables 3 and 4.

## Conclusion

Non-disabled test subjects documented significant keystroke reductions in realistic communications with a combination of rate enhancements. These results offer clear support to the idea that stored text can be helpful to alleviate problems related to a low rate of communication.

## References

Alm, N., Arnott, J.L. & Newell, A.F. (1992a). Evaluation of a text-based communication system for increasing conversational participation and control. *Proceedings Resna International, 12,* 366.

Alm, N., Arnott, J.L. & Newell, A.F. (1992b). An integrated multi level communication system for physically impaired non-speaking children and adults. *First International Conference of the Saudi Association for Handicapped Children.* Riyadh, Saudi Arabia, 7/10-11-1992.

Alm, N., Arnott, J.L. & Newell, A.F. (1992c). Prediction and conversational momentum in an augmentative communication system. *Communications of the ACM, 35 No 5,* 46-57.

Alm, N., Todman, J., Elder, L. & Newell, A.F. (1993). Computer aided conversations for severely physically impaired non-speaking people. *InterChi 1993,* 236.

Anson, D. (1993). Typing by voice on the Macintosh. *Resna Proceedings,* 13, 1993, 446.

Arnold, A. (1997). Acolug forum, 5 september 1997. [Online] Email:LISTSERVE@VMTEMPLE.EDU. Peer Letter System.

Balandin, S. (1994). Symbol board vocabularies. *Isaac 1994,* 548-550.

Beattie, W., McKinlay, A., Arnott, J. & Gregor, P. (1994). An investigation of the use of recency in word prediction algorithms. *Isaac 1994,* 496-498.

Beukelman, D.R. & Yorkston, K. (1977). A communication system for the severely dysarthric speaker with an intact language system. *Journal of Speech and Hearing Disorders 42*, 265-270.

Beukelman, D.R., Yorkston, K.M., Poblete, M. & Naranjon, C. (1984). Frequency of word occurrence in communication samples produced by adult communication aid users. *Journal of Speech and Hearing Disorders 49*, 360-367.

Beukelman, D.R. & Mirenda, P. (1992). Augmentative and alternative communication. Management of severe communication disorders in children and adults. Paul Brookes Publishing Co. Baltimore, Maryland.

Bühler, C., Heck, H. & Wallbruch R. (1994). Effective communication with Basco: Smart text version in German. *Isaac 1994,* 484-486.

Card, S., Moran, T.P. & Newell, A. (1983). *The psychology of human computer interaction.* Lawrence Erlbaum.

Cypher, A. (Ed.) (1993). *Watch what I do, programming by demonstration.* MIT Press, Cambridge, Massachusetts

Darragh, J.J & Witten, I.H. (1992). *The reactive keyboard.* Cambridge Series on HCI.

Demasco P.W., & McCoy, K.F. (1992). Generating text from compressed input: An intelligent interface for people with severe motor impairments. *Communications of the ACM*, *35,* 68-78.

Dix, A., Finlay, J., Abowd, G. & Beale, R. (1993). *Human-Computer Interaction.* Prentice Hall International (UK).

Ehrenreich, S.L. & Porcu, T. (1982). Abbreviations for automated systems. *Directions in Human Computer Interaction.* Albert Badre, Ben Shneiderman, eds. 111-135.

Foulds, R.A. (1980). Communication rates for nonspeech expression as a function of manual tasks and linguistic constraints. *Resna Proceedings, 83-87.*

Givón, T. (1989). *Mind, code and context. Essays in Pragmatics.* London, Hillsdale, Lawrence Erlbaum Associates.

Goffman, E. (1974). Frame analysis. An essay on the Organisation of Experience. Harvard University Press.

Gombrich, E.H. (1950). Fifteenth edition 1989. *The story of art.* New Jersey, Prentice-Hall Inc.

Guenthner, F., Krüger-Thielmann, K., Pasero, R. & Sabatier, P. (1994). Guided composition of text used in communication aids for handicapped persons. *Isaac 1994,* 474-476.

Hawking, S. (1997). My experience with ALS. Report posted on Broedel's ALS forum.

Hickey, M., Uytenbroek, M., & Alm, N. (1996). Fuzzy information retrieval in an AAC system. *Isaac 1996,* 483-484.

Higginbotham, J.D. (1992). Evaluation of keystroke savings across five assistive communication technologies. *Journal for Augmentative and Alternative Communication, 8,* 258-271.

Higginbotham, D.J. & Wilkins, D.P. (1996). A communication frame based language system for AAC. *Paper, Max Planck Institute,* Nijmegen, Netherlands.

Hoag, L.A., & Bedrosian, J.L. (1992). Effects of Speech output Type, Message Length and

Reauditorisation on perceptions of the Communicative Competence of an Adult AAC User. *Journal of Speech and Hearing Research, 35,* 1363-1366.

Hoag, L.A., Bedrosian, J.L. & Johnson, D.E. & Molineux B., (1994). Variables affecting perceptions of social aspects of the communicative competence of an adult AAC user. *Journal of Augmentative and Alternative Communication, 10,* 129-137.

Horstmann Koester, H.M., & Levine, S.P. (1996). Effect of a word prediction feature on user performance. *Augmentative and Alternative Communication, 12*, 155-168.

Horstmann Koester, H.M., & Levine, S.P. (1997). Keystroke-Level Models for User Performance. *Augmentative and Alternative Communication, 13*, 239-257.

Horstmann Koester, H.M., & Levine, S.P. (1991). The effectiveness of word prediction. *Resna Proceedings, 11,* 100.

Horstmann Koester, H.M., & Levine, S.P. (1994 a). Quantitative indicators of cognitive load during use of a word prediction system. *Resna Proceedings, 14,* 118.

Horstmann Koester, H.M., & Levine, S.P. (1994 b). Quantitative indicators of cognitive load during use of a word prediction system. *Resna Proceedings, 14,* 569.

Joyce M. (1997) Column in Isaac Bulletin 2, 3, 4. *Isaac Bulletin.*

Levelt, W.J.M. (1994). What can a theory of normal speaking contribute to AAC? *Isaac 1994,* 18.

Levelt, W.J.M. (1989, 1993). *Speaking: From intention to articulation.* Cambridge, MA:MIT Press.

Levinson, S.C. (1983). *Pragmatics.* Cambridge, Cambridge University Press.

Light, J. (1997). Communication Is the Essence of Human Life: Reflections on Communicative Competence. Don Johnston-Isaac Distinguished lecture. *Augmentative and Alternative Communication 13,* 61-70.

Light, J., Lindsay P., Siegel L. & Parnas P. (1990).

The effects of message encoding techniques on recall by literate adults using AAC systems. *Journal for Augmentative and Alternative Communication, 6,* 184-201.

Mc Gregor, A. (1996). *Speaking to you.* Video available from Dundee University.

Merchen, M.A. (1997) Message 600. *Acolug forum*. [Online] Email:acolug@temple.edu or <maryann@SOLTEC.COM>. 14 december 1997.

Mirenda, P., & Beukelman, D.R. (1990). A comparison of intelligibility among natural speech and seven speech synthesizers with listeners from three age groups. *Journal for Augmentative and Alternative Communication, 6,* 61-68.

McNaughton, D., Fallon, K., Tod, J., Weiner, F., & Neissworth, J. (1994). Effect of repeated listening experiences on the intelligibility of synthesized speech. *Journal for Augmentative and Alternative Communication, 10,* 161-168.

Newell, A.F., Arnott, J.L., Booth, L., Beattie, W., Brophy, B., & Ricketts, I.W.. (1992). Effect of the "PAL" word prediction System on the Quality and Quantity of Text Generation. *Augmentative and Alternative Communication, 8,* 304-311

Norman, D. (1988). The psychopatholgoy of everyday things. Basic books.

Odell, S.J. (1987). The power of language. A philosophical analysis. In: Leah Kedar ed. *Power through discourse.* 19-39.

Reitman Olson, J. & Olson, G.M. (1990). The growth of cognitive modelling in human computer interaction since GOMS. *Human Computer Interaction 1990,* 5, 221-265. Included in: R.M. Baecker, J. Grudin, W.A.S. Buxton, S. Greenberg. Readings in HCI. Toward the year 2000.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human-Computer Interaction.* Addison Wesley.

Smith-Lewis, M.R., (1994). Discontinuity in the Development of Aided Augmentative and Alternative Communication Systems. *Journal for Augmentative and Alternative Communication, 10,* 14-26.

Stuart, S., Beukelman, D.R., & King, J. (1997). Vocabulary Use during Extended Conversations by Two Cohorts of Older Adults. *Augmentative and Alternative Communication, 13,* 40-47.

Stum, G.M., & Demasco, P.W. (1992). Flexible abbreviation expansion. *Resna 12,* 371-373.

Swiffin, A.L., Arnott, J.L., Pickering, J.A., & Newell, A.F. (1987). Adaptive and predictive techniques in a communication prosthesis. *Journal for Augmentative and Alternative Communication, 3,* 181-191.

Tannen, D. (1986). *That's not what I meant.* Norwood N.J. Ablex.

Tannen, D. Ed. (1993). *Framing in Discourse.* Oxford:Oxford University Press.

Todman, J., Alm, N., Elder, L. (1994). Computer aided conversation: A prototype system for non-speaking people with physical disabilities. *Applied Psycholinguistics,* 15, 45-73

Todman, J., Elder, L., & Alm, N. (1995). Evaluation of the content of computer-aided conversations. *Augmentative and Alternative Communication, 11*, 229-235.

Todman, J. & Alm, N. (1995). Use of a Communicaton Aid (TALK) by a Non-Speaking Person with Cerebral Palsy. *Communication matters, 9*, 3, 18-25

Todman, J., Lewins, E. (1996). Conversational rate of a non-vocal person with motor neurone disease using the 'TALK' system. *International Journal of Rehabilitation Research, 19*, 285-287

Todman, J. (1998). Personal communication.

Trepagnier, C. (1995). Design goals for augmentative communication. *Communication Outlook, 15*, pg 12-21.

Vanderheiden, G., & Kelso, D.P. (1987). Comparative analysis of fixed-vocabulary communication accelaration techniques. *Journal for Augmentative and Alternative Communication. 3,* 196-206.

Vanderheiden, G., & Lloyd, L. (1986). *Communication systems and their components.* In Blackstone S.W, (Ed.) Augmentative Communication: an Introduction, 49-161. American Speech Language Hearing Association.

Venkatagiri, H.S. (1992). The effect of rate and pitch on the intelligibility of synthesized speech. *Journal for Augmentative and Alternative Communication, 7,* 84-89.

Venkatagiri, H.S. (1993). Efficiency of lexical prediction as a communication acceleration technique. *Journal for Augmentative and Alternative  Communication, 9,* 161-167.

Venkatagiri, H.S. (1994). Effect of window size on rate of communication in a lexical prediction AAC-system. *Journal for Augmentative and Alternative Communication, 10,* 105-112.

Verrips, J. (1992). Branching selection of suggestions. *Interacting With Computers, 4-1,* 68-82.

Verrips, J. (1993). Filtered suggestions, demonstration. Amsterdam, *InterChi Conference Proceedings,* 465.

Wood, M., & Lewis, E. (1994). A multi-lingual prediction system. Isaac 1994, 471-473.

Wood, L.A., Rankin, J.L., & beukelman, D.R. (1997). Word Prompt Programs: Current uses and Future Possibilities.

American Journal of Speech Language Pathology. Vol 6, 57-65.

Yorkston, K., Smith, K., & Beukelman, D. (1990). Extended communication samples of augmentative communicators II: Analysis of multiword sequences. *Journal of Speech and Hearing Disorders, 55*, 225-230.

Zipf, G.K. (1949). *Human behavior and the principle of least effort.* Addison Wesley. Hafner, New York, reprinted 1965.

## Appendix 1 Some technical details

Before reading this section readers should consult figures 1 to 3. In the suggestion lists for phrases the first letters of words are in upper case and are given priority during selection. If the phrase "Sorry, I missed my train " is in the edited file, then "Sorry, I Missed My Train " will be in the list of suggested phrases; one may key "simmt" and see little else, often "simt" will work as well. This is slightly unnatural; a phrase is not a name, users may not remember the wording of phrases they want to select and may expect suggested phrases relevant to the current topic. However, first letter coding is a frequently used technique to abbreviate, is easy to learn and to remember, and also allows to select parts of phrases.

Words are ordered on length first, to facilitate visual scanning of the list. Also, the two most recent words in the list are marked by comma and period. This allows to repeat a recent word without looking at the screen, and is a way to bypass scanning. After three keystrokes a second database is consulted to read in less frequent words. For a long word, then, users may look after three keystrokes, then either hit a numeral or finish without looking. For shorter or more frequent words, less keystrokes will do. If the aid loads a file with texts on a certain subject, those words will be considered frequent. Finally, users may skip frequent characters. To speak "liquor ", "lqr," or even "lq," will do.

Access by two-key Morse Code poses several minor problems if combined with text-prediction. First, cursor movement becomes problematic, as the codes for arrow keys are long and one often needs several arrows to move the cursor. To adress this problem the last keystroke gets bound to e(=.) if it moves the cursor. Any other code than . (including nonsense codes) will unbind. Therefor ten times ArrowUp (.-..-), with a total length of 10*5=50 is replaced by once ArrowUp (.-..-) then nine times e (.), a total length of 14. Second, recency accept becomes bothersome as the codes for comma or period are long. Therefor, one can exchange BackSpace -- and comma -..... during text prediction, as was done during the experiments reported in Table 4.

## Some experiences

To get healthy test-subjects to learn to use the aid in simple conversations, two computers were used and stored material that was available from earlier experiments. A student of linguistics and an (american) psychologist learned the system in a few hours: Esther Parigger (EP) and Terry McKee (TK). It took over ten hours before they began to anticipate on the behaviour of the program and used gestures to clarify their intent. Subjects seemed completely absorbed by the machines' suggestions and by the meanings of upper case characters but they reported that using the system was fun. The psychologist (TK) appreciated the word prediction to improve her Dutch. They used eye contact and communicated with facial expressions, as when a word was not understood correctly or was not known, but not during typing. Other noteworthy observations were that phrase prediction and paragraphs were used rarely and mainly for stereotypical utterances like "Shall we stop now ". Training specific activities with specific paragraphs increased frequency of phrase prediction a little. Though high keystroke reduction was documented after only fifteen hours of practice (over 50%), communication remained slow and brittle. Subjects had to work hard and learned to expect much less than usual. A topic had to be decided before each conversation and to explain things was difficult. Sometimes a conversation of twenty minutes seemed almost devoid of content apart from explaining two dutch words. Unlike expected, turn taking and contact were not much of a problem and subjects rarely uttered much emotion when allowed to speak up. Subjects clearly identified with each other; after all, they used the same system. Still, both subjects had part of the screen reserved for phrases that requested to repeat, to expand, or to explain again. Logfiles made by the software as well as direct observation and video showed that the margin menu was used quite frequently. Often a phrase was voiced during typing, repeated one word after another with a function key, then repeated again through the margin menu as if to sum it up and effect turn-taking.

The american subject had significant trouble to understand the synthetic speech and used a rather limited vocabulary. As I wondered how fast the dutch subject would become in a less symmetrical situation the english speaker was then allowed to read her partner's monitor and to speak slowly. This intervention made a big difference as the rate went up impressively. Suddenly, role-plays became possible. Due to reading on the screen, there was less of a need for repeating with the margin menu. See Table 1 for data from a logfile made automatically by the software after about eighteen hours of practice. Phrase prediction was used less then expected and its practical value remained uncertain. It was hard to make subjects change their material: much of it was never used. I conclude from these experiments that motivated subjects combined several rate enhancing mechanisms in simple conversations after a reasonable amount of practice. Trying to communicate with an aid learned us a lot about ourselves and about communication, and is a cheap and valuable experience that we suggest to everybody interested in AAC.

## Appendix 2 More Problems

The literature on Alternative and Augmentative Communication (AAC) contains more problems than the rate problem (Beukelman & Mirenda 1992; Levelt 1994). A limited vocabulary leads to a problem of semantical restriction and of vocabulary selection. Coding is a problem once one has selected some material and even more so once an augmented communicator decides to change it. Learning can be hard and communication poses problems for caregivers and family as well and economical servicing and distribution is a problem for many involved.

Several persons with a speech disability value meaningful work, family life and holiday camps more than new communication aids. One subject refused demonstration of new aids with the words "I do not care if I am slow. If you sit down and take time to listen we will have fun. .. Perhaps I do not want to be with you at all." Another augmented communicator stated "Speed is an actual problem only 10% of the time. The way I look at things is that if people actually want to communicate with me, they don't mind the extra time I need ." (Merchen 1997). These quotes may help to remember that communication is a social process, does not necessarily imply hi-tech and can be slow and convincing at the same time.

## Appendix 3. Stored texts

This appendix gives some background information on the idea to use stored texts to circumvent slow rate. It is not a new idea in AAC and has not worked as well as expected.

"... the problem is that phrases are very context-dependent, and a very large number are needed to allow for many situations. Remembering and accessing a very large number has not proved practicable. Using stored material emphasizes fluency over specificity, and users seem thus far to prefer to struggle on with being specific in their utterances despite the rate problem" (Alm, Arnott & Newell 1992c).

As words are the smallest independent semantical elements, phrase prediction is a somewhat funny proposal. Many phrases we utter are never re-used. Fixed combinations of two words are quite rare, at least in English: over 90% of everyday communication by healthy subjects is reported to be unique (Beukelman, Yorkston, Poblete, & Naranjon, 1984). Samples from augmented communicators show a somewhat different picture (Yorkston, Smith, & Beukelman, 1990). Some words are re-used often: 78% of words spoken by healthy older subjects belongs to the 250 most common words (Stuart, Beukelman & King, 1997).

To organise stored phrases in a natural way is difficult for many reasons. Apart from cultural and individual differences, some have argued that the nature of so-called speech acts is continuous, not discrete (Givón, 1989). If this reasoning is accepted, one wonders if texts, usefull for rate-enhancement, can be stored in a consistent way at all. Some have reported high inter-observer correlations of functional classification of texts (Todman, Alm & Elder 1994) with classifications of "speech acts" and of conversational goals. These data represent classifications of texts, not at all descriptions of actual conversations on a wide range of subjects by a representative sample of augmentative communicators. Likewise, their statement that "given the severe problems that users of current word-based systems face, a compromise between speed and precision may be worth considering" lacks sufficient empirical support. It sounds reasonable indeed, but is it, or is it not worth the effort?

These serious linguistic problems are accompanied by no less important philosophical objections. We speak, it seems, to express and convey meaning. If meaning depends on context as understood and interpreted by human beings, suggestion of meaningful texts is inherently complicated and error prone. The importance of context on meaning making can hardly be overestimated, see for instance Gombrich, (1948), Levinson, (1983), Goffman (1974) or textbooks on psychology. A simple phrase like "Yes dear" can mean an acknowledgment, an insult, or even a compliment and one may even wonder if meaning is preserved by quoting. Through the ages, philosophers have been interested in the ways natural language is used. An example is Odell (1987), who lists such principles as context, emphasis, background, intentionality, creativity, family resemblance, overlapping definitions, continuity, completeness, open texture, and sincerity. Again, the idea that quoted material might do the job is open to criticism.

Three groups of needs have been discerned to support augmentative communicators confronted with the rate problem: to select formulaic speech, to select reusable material and to key in unique expressions (Alm, Arnott & Newell, 1992b). Conversation analysis and speech act theory have been advocated to guide in the construction of text-based communication aids 'to maintain conversational momentum' (Alm, Arnott & Newel 1992c) and to perform a specific tasks like a job interview (Todman, Alm & Elder 1994). Some research even suggests that phrases might be retrieved because their subject is related to the last one according to fuzzy logic (Hickey, Uytenbroeck & Alm, 1996).

Different authors have advocated simple customizable aids that can be used in different ways to train for specific aspects of communicative competence and may be adapted to a user's changing needs. For examples see Light (1997). For support that discontinuity between aids is a problem see Smith-Lewis (1994). From the literature on AAC, it is not clear what phrases should be stored, how often individual phrases would be (re-)used or if people personalise stored texts. Many say that more studies are needed and that caregivers have to consult with patients and family.

Many people with acquired severe motor problems such as caused by Amyotrophic Lateral Sclerosis (ALS) use text-based communication aids. The case for stored material for this patient group is unclear. Some studies show communicative effectivity of stored material (as Todman, Elder & Alm, 1995), and a rate of 42 words per minute was reported after little training by an adult non speaking patient (Todman & Lewins, 1996). Some augmentative communicators report effective use of such systems in specific circumstances (McGregor, 1996, Hawking, 1996). Perhaps in a few hundred hours of practice one might - adapt material in unforseeable ways. Quite possible one can also learn to use word prediction more effectively. As common words are often short (Zipf, 1949) their selection seems to require a system that allows for automaticity. Word prediction as currently implemented is weak on automaticity compared to Morse Code, wordboards or Minspeak$^{TM}$, but practice might change that.

Different caregivers indicated the need to store specific messages for, say, occupational therapy. These messages may be selected rapidly for frequent functions as to introduce oneself, to say goodbye, to confirm, to express emotion or to request attention. Possibly augmentative communicators might get more attention for slowly typed in messages as well. AAC-users are rated higher in communicative competence when long phrases are uttered as compared to single word messages (Hoag & Bedrosian 1992, 1994). To understand synthetic speech requires attention and is probably easier with small phrases as compared to single words (McNaughton, Fallon, Tod, et al 1994; Mirenda & Beukelman 1990). Taken together, these are sufficient reasons to enable selection of stored phrases in a research context and to try out with selected patients.

A tempting way to attack slow rate is to let a computer program predict the next word. Such predictions have been proposed for letters, phonemes, words or even for short phrases and can be based on the last few keystrokes that a user has entered. Overviews of the literature can be found in Darragh and Witten (1992) or Horstmann Koester and Levine (1996). Different word-based programs are commercially available and display a list of suggestions from which users can select with numerals. The word "different " might be selected through the keystrokes "dif7" from a list that might also contain "dif-fident ", "difficult ", "differential ", "diffamation ", "diffraction " or "diffuse ". Typing "dif7" requires four different keystrokes to select "different ", ten characters. In this example word prediction allows a keystroke reduction of (10-4)/10 = 60%.

To operate predictive systems is difficult and it is hard to design systems that combine a high reduction of keystrokes with simple and efficient user's process. Just to read a suggested word on screen or to scan a list of words can easily outweigh time gained by a reduction of keystrokes (Horstmann & Levine 1991, 1994a). Predictions should come close to what users need, in order not to distract. To specify and implement what people really need is rarely easy and can be very difficult in AAC.

Many different predictive systems have been constructed to type in computer programs, generally speaking short texts with a restricted vocabulary. They are typically combined with an editor and a formatter and propose only syntactically valid suggestions. Language sensitive editors are not used often, presumably because text prediction on phoneme-level and on word-level almost inevitably slow users down. In a prize-winning study, the text-generation rate of a group of disabled users was studied. Word prediction led to a significant rate decrease (Horstmann Koester & Levine 1994 b, 1996).

Word prediction might be useful for subjects with severe motor problems and excellent linguistic and cognitive abilities. The cost per keystroke would be high and the cost to operate a complicated interface might be worth it. However, published evaluations with healthy test subjects do not show convincing time savings due to word prediction (Swiffin, Arnott, Pickering & Newell 1987; Venkatagiri 1992, 1994). Prescribers might arguably not propose systems to speech-disabled people if they were not shown to offer concrete and measurable advantages to healthy test subjects.

Some research on word prediction for computer programmers focussed on time and attention requiring user's process (Verrips 1992, 1 subject; Verrips 1993, 1 subject). In a copying task, predictive mechanisms did speed up selection by about 15% as compared to ordinary typing and reduced the number of keystrokes by 50% or more. This editor knew about the syntax of the programming language, unique suggestions were accepted automatically and auditory feedback minimised the need for verification on screen. Upper case letters were added to the suggestions and allowed a rapid decrease of list length. Users were slowed down, it seemed, by thinking, scanning of the list on screen, amazement due to list content, choosing an item, reorientation of attention to the keyboard, verification of screen-image after selection and reading of paper for a next line. In a conversaton, one would expect more of such activities, not less.

More research suggests that many different time and attention requiring processes intervene during word prediction. Presumably normal subjects with an artificial handicap took 46 centiseconds more per keystroke during a 15-word window condition task (2.44 seconds) compared to a 5-word window condition (1.98 seconds), (Venkatagiri, 1994). No comparison was made with a condition without prediction. Horstmann Koester & Levine (1996) compared typing with word prediction and typing without word prediction, in both conditions with an artificial handicap. A subtle increase in time per keystroke could be documented in the first group and seems to reflect a more complicated mental model. Combining a Keystroke-Level Model and analytical methods to study user performance showed that "list search during use of word-prediction is a complex process influenced by the particular conditions of the search. Both serial search and anticipation of the list contents significantly affected subjects' list search times" (Koester & Levine 1997).

From a linguistic point of view, the case for word prediction does not look better. It is hard to imagine simple methods that greatly reduce the large number of possible words or that would do better than guessing by a patient's family member. Complicated methods will almost certainly lead to much amazement. Long words are less frequent then short words, have more specific meaning and are available in impressive quantities as any dictionary testifies. Frequency of use depends on the context in an actual conversation but such contexts cannot be easily modelled, due to the importance of expectation and individual perceptions. See Deborah Tannen (1986, 1993) for indirect support for this opinion.

Nevertheless, word prediction remains a fascinating technique to explore and to apply and is tried by many developers. It can be used in different contexts and many ideas to make it work in AAC have not been sufficiently researched. Some ideas available in the literature are improving predict-ahead of the next word once a word has been finished, using syntax, associative semantics, frequent letter combinations, banded recency, other combinations of recency and frequency and doing more vocabulary studies. See for instance Wood & Lewis, (1994), Guenthner, Krüger-Thielmann, Pasero, & Sabatier, (1994), Beattie, McKinlay, Arnott & Gregor, (1994), Bühler, Heck & Wallbruch, (1994), Balandin (1994). Some authors consider word prediction useful for subjects with language and learning disabilities especially spelling problems (Wood, Ranking & Beukelman, 1997; Newell, Arnott, Booth, Beattie, Brophy, & Ricketts, 1992). Word prediction (or word prompting) may also be useful for learning foreign languages. Some languages would require very subtle algorithms to make word prediction work at all. Like possible advances in natural language programming, this lies outside the scope of this text.

Some voice output communication aids (VOCAs) allow macros, special key combinations that automatically expand into some meaningful message (abbreviation expansion); +y might mean "Yes I think so". Macros closely resemble text prediction and are useful during ordinary text editing. Over a thousand macros may be defined and small lights may suggest the next key as in Minspeak$^{TM}$.

## Appendix 5. The Predictive Mechanism

This appendix describes the rather unusual algorithms used to implement a forgiving form of word prediction and phrase prediction. It can be skipped by non-programmers. Mathematical background can be found in logic and in recursion theory, the mu-operator selects the smallest element of a set that has a certain property. Other analogies are so-called boolean short cut evaluation (in programming) and analysing interpretation of ironic utterances (see Levinson, 1983) and programming by demonstration (Cypher Ed. 1993). Whether there is a need for such mechanisms in other contexts is an open question.

Much literature exists on abbreviations and predictive systems for speech disabled subjects (VanderHeiden & Kelso, 1987; Darragh & Witten, 1992; Vanderheiden Lloyd, 1986; Stum, 1991; Demasco & McCoy, 1992). Abbreviations are used in other contexts and guidelines suggest a simple rule that is communicated to the operators and fixed length abbreviations (Ehrenreich & Porcu, 1982). Examples of rules are truncation and truncation plus selective vowel deletion (trn pls slc vwl dlt). Such rules propose an abbreviation like "trn" when a text is given such as "truncation".

The predictive mechanism of this system computes a set of texts when an abbreviation is proposed, almost the opposite mechanism. To do so, several sets of texts are computed using predicates, functions that accept some items in a set and reject all others. A predicate defines a subset of all items with a certain property. If {"Hello", "How are you?", "Nice to see you"} is a set of texts, then the predicate 'starts with "H"' defines a subset of the first two items. 'Contains an "a"' now defines the subset with only the second item: {"How are you?"}. A subset is empty if the predicate defines a property that no item satisfies; sets or subsets of texts are displayed on screen as numbered lists.

We shall call a suggested text-string a_string and what the user typed in shall be called some_abbreviation. When a first letter of a word or a phrase is typed in the software initialises a list of suggestions from a database internal to the program. In our terminology it is a list of a_strings. When a second or a third letter is typed in, six different elementary predicates compute which suggestions comply and store results in six different lists. The first nonempty list from those six lists is displayed on screen.

Predicates $p_1$ to $p_6$ are used:

$p_1$. A_string contains all upper case letters from some_abbreviation and ordered from left to right and consecutively.

$p_2$. A_string contains all upper case letters from some_abbreviation ordered from left to right.

$p_3$. A_string contains all upper case letters from some_abbreviation in any ordering.

$p_4$. A_string contains some_abbreviation as a left substring.

$p_5$. A_string contains all letters, either upper case or lower case, from some_abbreviation and ordered from left to right.

$p_6$. A_string contains all letters, either upper case or lower case, from some_abbreviation in any ordering.

Predicates can be thought of as advisers. If the earlier one has nothing to say, the next one may speak: advisers are ordered hierarchically. If $P_e$ defines a predicate that accepts no items and $P_x$, $P_y$, $P_z$ are predicates the following rules apply.

. $P_x\,P_y$ means:

if the result of $P_x$ is not empty

then $P_x\,P_y = P_x$

else $P_x\,P_y = P_y$

. $P_x\,P_x = P_x$

. $P_x = P_x\,P_e$ (if no item is accepted, the empty set is proposed)

. $(P_x\,P_y)\,P_z = P_x\,(P_y\,P_z)$

The ordering $p_4\,p_5\,p_6$ (1a) is used for words but will not do to select phrases using FLUC; that would require something like $p_1\,p_2\,p_3\,p_4\,p_5\,p_6$ (1b). Some_abbreviations that consist of first letters of words rarely are left sub-strings of words as is required by $p_4$ and therefore 1b will generally result in short lists of suggestions on the screen. However, to type part of the first word of a phrase can be an effective way to filter a list of phrases using $p_4$ and need not exclude use of $p_1$ for other phrases. For words 1a is good enough and for phrases the implemented ordering is ($p_4$ or $p_1$) $p_2\,p_3\,p_5\,p_6$ (1c). This will work for a small database. If a user types on presumably the word is not found and a larger database is searched, using $p_5$ after three keystrokes followed by $p_4\,p_5$. Perhaps in the future this could be a two step process, searching a small database of recent and frequent words after two and a larger database after three or four keystrokes.

Users need only rarely mentally represent the interaction of individual predicates. In rare cases however one may be disappointed. If one types "exm" to select "example" the database might contain "exmarried". In this case "example" will NOT be suggested! This will usually be corrected if one types on, "exmp" will do.

See Table 9 for all the some_abbreviations that one might use to suggest the a_string, "Dag Goeden Avond" (Hello good evening). All other predicates are ignored as well as all other texts. Predicates $p_5$ and $p_6$ have very long lists of some_abbreviations but will only be used infrequently as mostly $p_1$ and $p_4$ will do. No matter how selective the predicates are, users will need a numbered list on the screen to pick one from pairs such as "I Will Not Go There " and "I Will Never Go There ". One may select the last one after "iwnv" using $p_1$ and then $p_5$.

To get an impression of the effectivity of the FLUC- algorithm, an English article was processed. Variable length abbreviations were determined of every line that consist of first letters of every word. These abbreviations were sorted and counted. 120 of 541 lines had unique two letter abbreviations, 389 of 494 lines had unique three letter abbreviations (541 - 494 = 47 lines did consist of two words only and were ignored), 433 of 443 sentences had unique four letter abbreviations, ignoring 98 (541 - 443 = 98) lines with less than four words. A unique three letter abbreviation means that to type in three first letters will create a list of a single suggestion. I conclude that to select indicidual lines from this short article the first predicate would be remarkably effective. Selection from long lists of known phrases using FLUC will require few keystrokes.

To compare word prediction with and without recency enhancement another study was done. A simple text was typed in either with word prediction only or word prediction plus recency marking and only short words after the first keystroke, a property that is rarely used. In both cases, no other rate enhancements were used and listlength was eight. The text was taken from (Higginbotham 1992), fourth grade example with average word length at 5.4, simple words and rich in punctuation. With word prediction-only 406 keystrokes selected a total length of 625 (116 words, 35 % saving). Word prediction-plus-recency achieved 39% keystroke reduction, 30 out of 81 word predictions ended through recency selections. Results would have been slightly better if use of other rate enhancing techniques would have been allowed as in (Higginbotham, 1992), but not much.

The forgiving implementation of word prediction might allow

to rapidly decrease listlength. To test this hypothesis

the second chance algorithm was set off and a file was used

with about 630 English words. A set of 20 words was prepared with a vowel as a second letter, picking the longest such word for each letter in the alphabet, if present. Listlength was then compared after either the first two characters or first character, next consonant. Thus "de" and "dt" would be used to find "determination". In the second condition listlength was equal three times, longer two times and shorter fifteen times. This is a significant difference (token test; alfa = 0.05; Wilcoxon rank test alfa = 0.005). Total listlength after twenty times two keystrokes was 87 in the first and 40 in the second condition. However, in three out of twenty cases the second condition required another keystroke to display the word one needed. In those cases other words interfered with selection. For example, "argument ", that starts with "ar" might interact with selection of "aeroplanes " in the second condition. Note that actual users may use the system in very different ways.

## Appendix 6. Task analysis to optimize text prediction

A problem with the design of all interactive software is how to model what users will go through. This can be done on very different levels of abstraction and should also be done with potential users. This appendix describes some such models as are used in Human-Computer Interaction, recent textbooks are Dix et al, 1993, or Preece et al, 1994. Apart from help in design, such models can be used to minimise the time and attention users need to accomplish their tasks. These models not easy to validate and typically ignore that the perceptive, cognitive and motor components may show some overlap, can be trained and are influenced by the complexity of the task at large.

Tables 5 and 6 assume that users decide first what kind of utterance they require. Users have to choose between slowly type in a specific answer, select a phrase or part of a phrase that comes close, first select a comment bound to an upper case letter, and so forth. To realize such a plan, users may need scanning, reading, deciding, recall etcetera, see Table 8. When one reflects long on Table 6 it strikes that users have to decide if a word is bound to uppercase before they hit its first character, which may interfere with automaticity. Therefor, a function was implemented that interpretes the last keystrokes as if they had been uppercase, even if text prediction has been started up, see Table 7. Other function keys are bound to phrases, stop text-prediction (F7), expand the selecting string to a recent word (F6), selects part of a recent phrase (F10) or reinitialise to find phrases with a certain substring (F9).

Presentation and ordering of suggested texts is important as scanning a list of texts and can take much time and attention. Lists therefor have to be both informative and predictable (Horstmann & Levine 1991; Koester & Levine 1997; Anson 1993; Venkatagiri 1993; Venkatagiri 1994; Card, Moran & Newell 1983). See figures 1 to 3 for an impression of the system used. Words are ordered on length, longer words first, and comma and period mark (and select) the two most recently used words.

One way to analyse word prediction is to abstract from individual differences, from training, from the word a user has to type in and from the context of use. A user can then be described as a stream of choices, reading, keying and so forth. Two ways to describe this stream are a grammar called a Goal Oriented Method Selection (GOMS) analysis or a Keystroke Model (Card, Moran & Newell, 1983; Olson & Olson 1990). These techniques offer reasonable first order estimations because elementary processes take time that can be estimated quite reliably, though not necessarily reliably enough for applications for the disabled. See figures 5, 6 for imprecise models of what a specific user goes through selecting a specific word with a specific database.

The forgiving implementation of appendix 5 that suggests the word "forgiving" when one types "fgv" will not necessarily be helpful for all people as is clear from figures 5 and 6. List length will be shorter if the right letters are chosen but only at the cost of a decision: *Decide_Which_Letter. Key_That_Letter* instead of *Key_Next_Letter* and this decision may well take a second according to what psychologists call Fitt's Law.

Users may know if the word that they want is in their database and if it has been selected recently before they type the first letter. If certain that it is not the user process can be simplier, see figure 7. A user who is certain that it was selected recently may use *Key_First_Letter.Key_Komma*. This resembles the scheme of the Liberator, where the "core" vocabulary is selected by Minspeak$^{TM}$, and the "fringe" can be accessed by word prediction. Different mechanisms are also present in EZ-Keys and in other vocas. For long words, people will often type three keystrokes before they look at the screen, because after three keystrokes the background database is consulted. See figure 8 for another imprecise grammar of user processes.

Scanning makes process *Key_Next_Letter* quite complicated, something like *Look_At_Screen.Find_Letter.Decide_Code. Select_Code.Verify-Result*. Still, when five keys are available quadrant selection can be fast and requires little instruction. Users will decide the code first then select without further ado. Frequent codes will be learned easily if the ordering of characters is consistent and simple. Scanning with a single switch however is quite a different matter and is error prone. Errors must be corrected easily (hitting 5) and users need acoustical metacommunication to hear which quadrant is suggested and how small it is. Users wait for the machine and therefor cannot easily automate their actions. Access is faster with two keys using ,,=1 ,.=2 .,=3 and ..=4 (with space=5) but Morse Code seems still faster for those who learn it and can be supported directly from the keyboard.

One might search for improvements to word prediction in semantics. Disabled subjects may experience difficulty to indicate persons or objects in a new communicative context labeled frame by (Higginbotham & Wilkins 1996). Traditionally frame means 'perceived context' as in Tannen (1993), Goffman (1974) or Givón (1989). Communicative context might influence word prediction through adaptation of recency information. As an example, consider the phrase "John went into the restaurant. He ordered a hamburger and a coke. He asked the waitress for the check and left." This phrase has often been quoted to illustrate the concept of a frame: the term restaurant makes one infer there must be a specific (the) waitress. If copied twice, to see the effect of quasi optimal adjusting of recency marking, 18 out of 21 words can be selected the second time with just the first character then either '.' or ','. To exploit this effect, recency is updated when one reads in a new paragraph.

## Appendix 7. Pros and Cons

Software engineering confronts one with an immense amount of details and futilities. To concentrate on the process of design can help to focus on the users and can be valuable after the fact as shows from the curious title "A Rational Design Process, how and why to fake it" (Parnas and Clement, 1986). Writing lists of pros and cons was helpful to keep the system small. They are not exhaustive.

### General characteristics.

Cheap enhanced editor

Pro: Can be shared with researchers to studythe value of stored material.

Con: No environment control, no Minspeak, no access to other software.

Edit functions

Pro: Allows users to maintain their own texts and facilitates learning.

Con: Confusing, there is more than an editor involved.

Text file as format

Pro: Easy integration of data from other systems and conceptually simple.

Easy maintenance of user's own vocabulary.

Con: No icons. Does not look very attractive or intuitive.

Stored toggles

Pro: Allows to reduce the inter-face to what a special user needs.

Con: Enlarges programming effort, cognitive load.

## Selection mechanisms

Cursor mediated

Pro: Direct, presumably fast and maintains the data

Allows slow selection of fragments, sometimes necessary for comprehension

(see Higginbotham and Baird, 1995).

Con: Increases memory load. The cursor will not always be near the intended text.

Margin menu

Pro: Supports story telling and paragraph selection.

Con: Distracts some, binds keys.

Upper case letters as macro keys

Pro: Fast and flexible. Allows automaticity.

Con: Increases memory load.

26 comments is not much compared to thousands of codes in Minspeak.

Phrase prediction

Pro: Allows to quote stored messages without referral to paragraphnames.

Allows to quote a few words from a phrase.

Con: It is not clear how far communication with stored phrases can get you.

Subjects will not normally remember phrases precisely enough.

Prediction uses FLUC, First letters upper case

Pro: High keystrokes reduction and possibility to select parts of phrases.

Con: FLUC may lead to much reading on screen, offers little semantical support and requires attention shifting.

Word prediction

Pro: Keystroke reduction.

Supports many different abbreviation schemes without memory load

Con: Disturbs some users and not physiological. No significant rate enhancement reported in the literature.

Second chance algorithm

Pro: Supports large databases word prediction while list length can be relatively short.

May facilitate reformulating using own words.

Con: Not always intuitively clear, as during the first few keystrokes word-prediction

is based on the file used, that may be changed often.

Order first with recency then with frequency is a defendable alternative.

Lists ordered inversely on length

Pro: Word prediction will mainly be used for long words because many frequent short words are faster typed in anyhow. Frequency based ordering cannot respond to changing context and recency based ordering leads to unstable screen-images.

Con: It can be annoying that frequent short words are not suggested and can not all be bound to uppercase letters.

Recency marking in prediction lists

Pro: Due to quality of synthetic speech and other factors, repetition is a much practised technique in AAC. In roleplays users seem to remember if they have recently selected a certain long word or phrase and use that knowledge. Lesser need to search on screen during selection.

Con: Increases cognitive load.

Text prediction with computed abbreviations

Pro: Users need not memorise codes and can select long words efficiently. allows the machine to react forgivingly, prf may suggest both "profit " and "performance ". Supports a broad class of languages and allows to select parts of phrases.

Con: Hardly any user testing has been done and it takes time to really understand it.

Presentation and organisation

Marking of selected text

Pro: Helps recall what has just been said. Visual cueing can help interpretation of synthetic speech (Fucci et al 1995). Useful for those that cannot afford synthetic speech or do not want to be overheard.

Con: Another unusual property.

Paragraphs

Pro: Flexible semantical selection support and maximal freedom to organise textual material.

Allows some abstraction and allows to start with little material.

Con: Increases memory load. The organisation of material in paragraphs will not necessarily support users during selection.

Paragraphs can be stored on disk.

Pro: Allows to grow selection lists gradually as more texts are integrated in the file.

Con: A flexible database is not necessarily easier to learn.

Groupwise scanning integrated

Pro: Necessary for some users. Combination with a separate system might interfere with presentation on screen. Allows to present the meaning of redefined keys.

Con: Insufficient research base. Some would advocate eye-tracking or infrared pointing as better techniques.

Cursor navigation with F6 <Search backwards>

Pro: Allows users to rapidly retrieve text based on a single word.

Con: Another property to remember and understand.

The system is almost modeless

Pro: Most functions are few keystrokes away.

Con: If they contain many functions, modeless systems can be hard to learn.

Toggle for speech

Pro: Allows quiet composition of messages that can be voiced when necessary.

Con: Another toggle to learn and understand.

## Implementation

Program in Turbo Pascal and Delphi

Pro: Fast system that supports a profiler. Code for text-prediction and for the editor was available from earlier projects and prototypes seemed unavoidable.

Con: Nevertheless, programming took much time.

Originally, no mouse functions

Pro: Those who need a single switch shall not appreciate the effort to emulate mouse-movement with scanning.

Con: Problematic learning and difficult integration with other software.

## Disadvantages

Recency information is stored separately from the text and is not updated after deletions. Therefore, recency based suggestions will not always be correct.

Combining different mechanisms may confuse users, especially those with poor reading skills.

To construct individualized databases presents problems.

User's memory load.

| Function | Knowledge |
|---|---|
| Conversation | History and content of the actual conversation. |
| Fragment storage | Editor, texts, upper case letters, some paragraph names, organisation of material. |

| | |
|---|---|
| Menu | Numeral keys, hyper-text. |
| Types then speak | Mechanism of speech. |
| Text prediction | Word-prediction, FLUC, recency binding. |
| Cursor mediated selection | End key and F1-F4. |
| Cursor navigation | F5-F10, PgUp, PgDn, Arrows. |

## Appendix 8. Several small studies

To evaluate a voca with speech disabled subjects poses many ethical and practical problems that I tried to circumvent using playful conversations with friends and family members. I wanted to find out if I would be able to work with it myself, which functions would be useful, if more bugs would appear and how we would experience the communication.

However, role plays require a database of stored material. To my amazement, researchers and distributors were not able to share databases of stored material as they had none. A notable exception was Susan Balandin who made available interesting data from vocabulary studies and organised by topic. I therefore inquired into the vocabulary of Bliss, stored material of aids using digitized speech, teaching material for afasics, frequency counts in written Dutch text and basic vocabulary for Minspeak. A file was compiled over a period of about a year with textual material from many different sources and was tried out informally. Many small changes resulted from those sessions. Keystroke reductions varied between 40% and 80%.

Interruptions and confirmations were especially effective. My children were annoyed by the standard character of most responses and by the low quality of the cheap speech card used. They said I became too talkative again. To get more experience with the system, some role plays were done with friends.

In the first study subjects were not paid, and the author played a speech disabled subject to make sure that the system was mastered adequately. Five situations were created to test the system's efficacy. Three subjects were asked to read their role aloud and then improvise and to react naturally. Documentation of fifteen sessions (3*5=15) consisted in audiotapes and logfiles to analyse keystroke reduction. Before the first session, each subject was allowed some time to familiarize with the sound of the synthetic speech and with the elapse of time between keystrokes and sound. In the role plays, goals were not totally compatible with one another, as in real life. After each session both subjects wrote down their impressions followed by some discussion to record comments when required.

The fifteen sessions took a total of four hours and led to some suggestions for technical improvements. Several bugs were found and repaired. With the exception of the margin menu, all rate enhancements were used. Test subjects reported they generally achieved the goals they set out to. Keystroke reduction varied during discussions. A total of 13920 characters were sent to the speech card with 6217 keystrokes. The average reduction is (13920 - 6217)/13920 = 55% and takes no account of the loss of specificity compared to ordinary typing. For this subject, word prediction only rarely leads to a time gain (Verrips 1992). It is not clear compared to what measurement such a time gain would have to be computed. The CPI or composite performance index was proposed to compare different aids (Venkatagiri 1993) and can be estimated at 1.29, assuming PI, CI, and DI are all 1 and that MAI = 0.55. Clearly, to really compare aids would require an elaborate research effort including third party evaluations of communicative effectivity.

This first study documented appreciable keystroke reductions and test subjects reacted encouraging. Results should be used with extreme caution due to the small number of subjects, the probability of observer bias, the very imperfect database used, the large variation among real voca-users and the simplicity of the role plays themselves.

To document which parts of the functionality are used a version was made that logs number of words, total word-length, number of keystrokes and number of selections per selection method. A print-out was made of 5 logfiles from the first pilot study and the same texts were again selected. See Table 10 for results. Interestingly, more keystrokes were spent on typing words than on all other mechanisms combined.

Different paid students were asked to learn the system and to criticise. The problems they experienced and their many suggestions for improvements helped to enhance ease of learning and functionality. The manual was rewritten several times and numerous changes were made to the program and to the database. These changes include to reinitialise the lists of suggestions during text prediction, select paragraphs with PgUp and PgDn and creating paragraphs of jokes and of insults.

The system was often demonstrated, both to friends and to health care workers. Describing scenes from comic books was found useful to change the database and to enlarge the range of things one might express easily. Table 11 documents a session that lasted 20 minutes. Keystroke reduction was well over 50 % and most mechanisms were used. During this particular conversation 98 utterances were documented, 41 of which were selected but a single time, and 19 others that were selected between 2 and 9 times each. The quality of such conversations was not impressive. People often misunderstood my intentions and said that they felt pity for me speaking without prosody. Some would try for themselves and came up with valuable suggestions, for instance the function to cut a few words from a phrase.

Later Allaert Troost agreed to learn the system. Allaert is a musician, a composer and a teacher. He plays the piano and the guitar and was paid. As his mother suffers from lack of speech due to laryngectomya he was highly motivated. Allaert learned the system superficially in about six hours and we then practised for another six hours. To imitate a high cost per keystroke he typed with a single finger. According to him, driving a car was easier to learn. We experienced many minor problems to establish satisfactory communication.

Again, numerous small changes were made to the program, to the manual and to the database. Upper case letters were bound to comments, reasoning it is more important to show one is present than to rapidly select frequently occurring words. Many phrases were split up to facilitate re-use. Most phrases that started with the meaning of a redefined upper case letter were revised, to facilitate the process of reformulating and coding. If the database contains many phrases that one might easily construct otherwise, users will be puzzled.

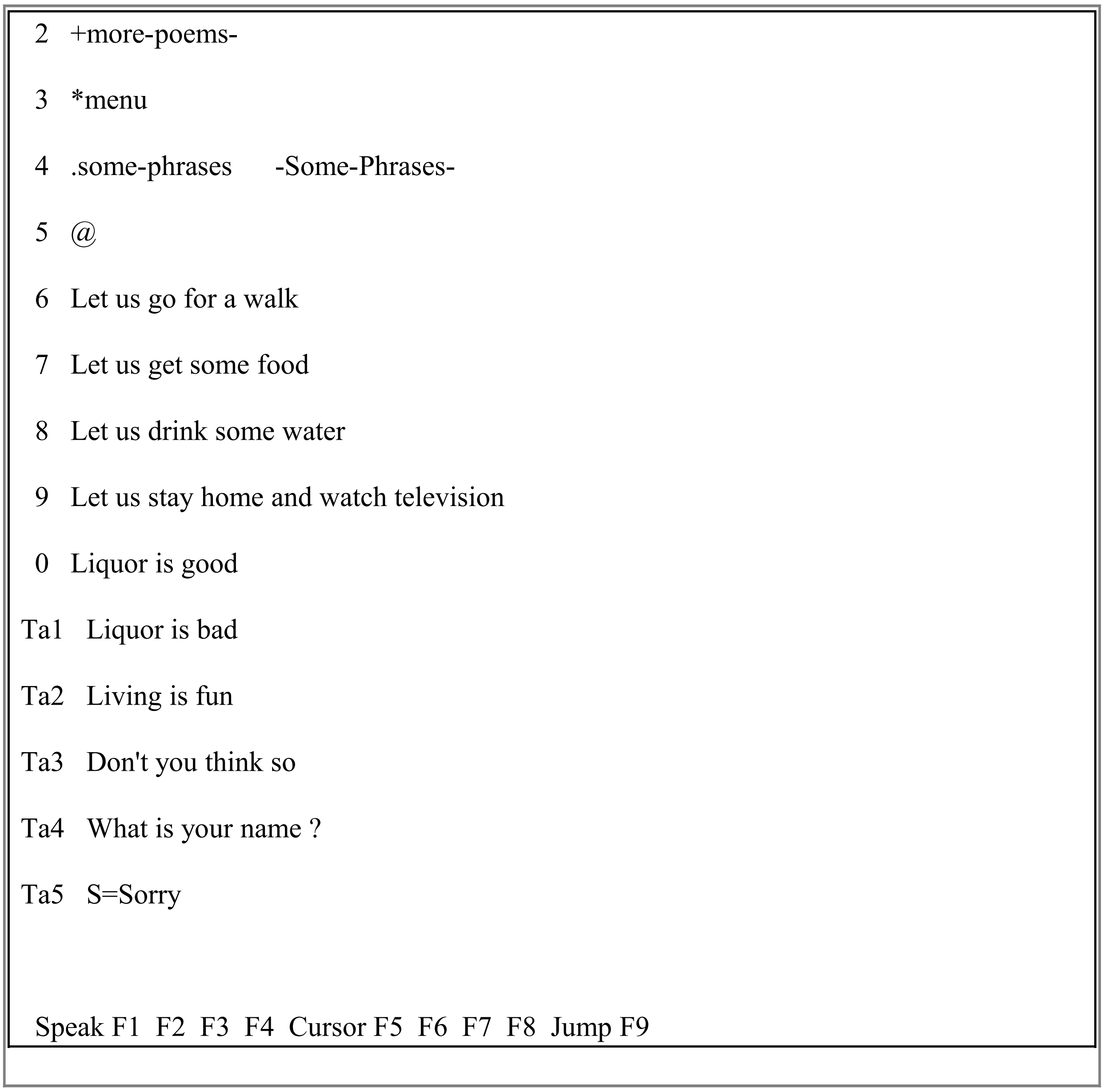
Learning to combine different selection methods and remembering the material were serious problems for this particular user. Video images were made and show that Allaert spent much time thinking and looked less at the screen as he learned the system better. Role plays were done after twelve hours of learning. See Table 12 for results from three consecutive role plays. Data should be interpreted with utmost caution and no conclusions about any actual patient group are justified.

Allaert expressed confidence that he could achieve communicative goals and used many ironic comments. He felt that more practice would help him to improve his speed and to fine-tune the database. After practicing nine more hours at home to study the new manual he again proposed some small improvements and found two more bugs. His logfiles showed that he achieved almost 60% keystroke reduction, 4239 letters selected by 1960 keystrokes, arrows and cursor movement not included. Allaert felt he needed more practice to personalise his database and to learn to communicate with group-wise quadrant scanning. After he spent some time thinking he proposed to better isolate paragraphs from each other and to facilitate repetition of recently used phrases using an output buffer. His proposals were implemented and some seem useful. A musical and motivated test subject learned to use the system in about 20 hours and came up with interesting improvements in the months thereafter. Such results would not have been possible with dependent patients and helped the author to better understand user problems. Also, these studies helped to create a database that worked for the author of the system and that later was used in other experiments with other test-subjects.

```
                      Partner

    <    <    <  speaks messages <   <   <   <



                    Monitor with     Partner may

     User sees         suggestions   >  verify the   >

<    suggestions <    & messages      messages that

     & messages                      were spoken


                                         P

U        An          Pc with the     Speech card    a

s    (alternative)       software       makes the     r

e  >    keyboard     >   to select    >  messages    >t

r     for input         messages        audible      n

                                         e

                                         r
```

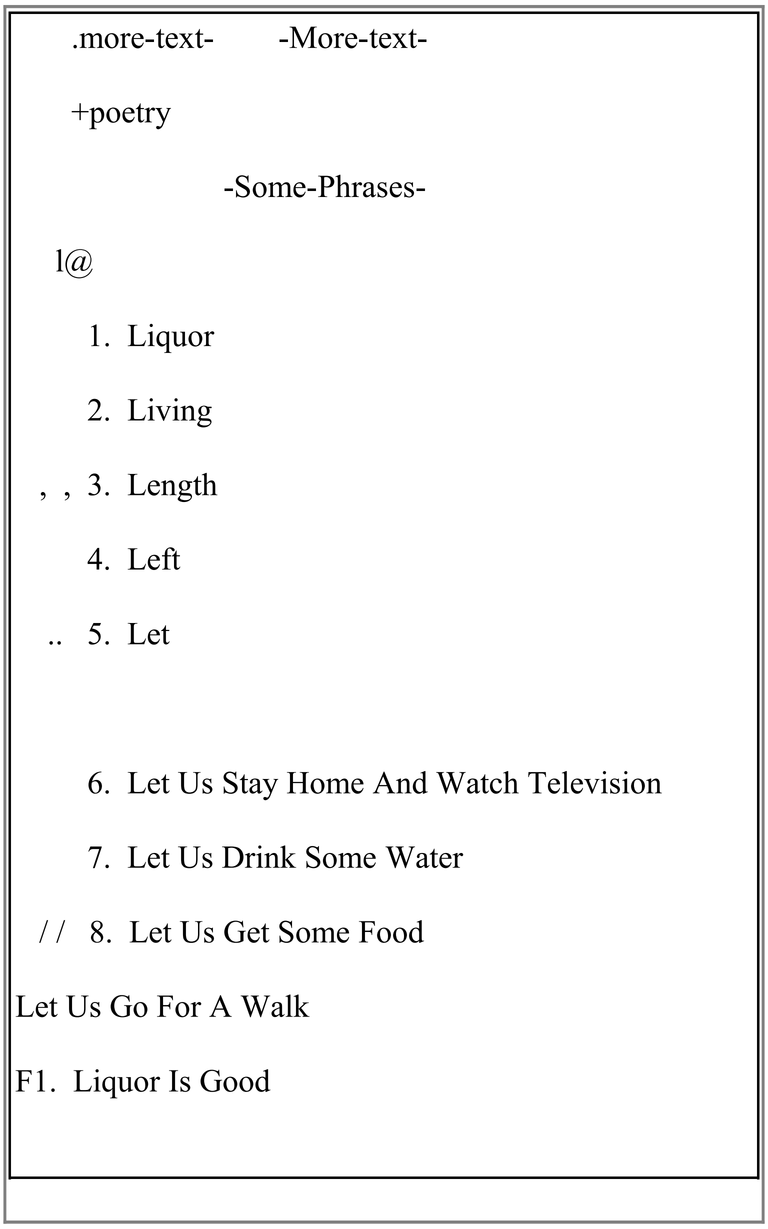
Fig 0. Communication system with two people, schematically indicated as User and Partner.

1 .more-text-



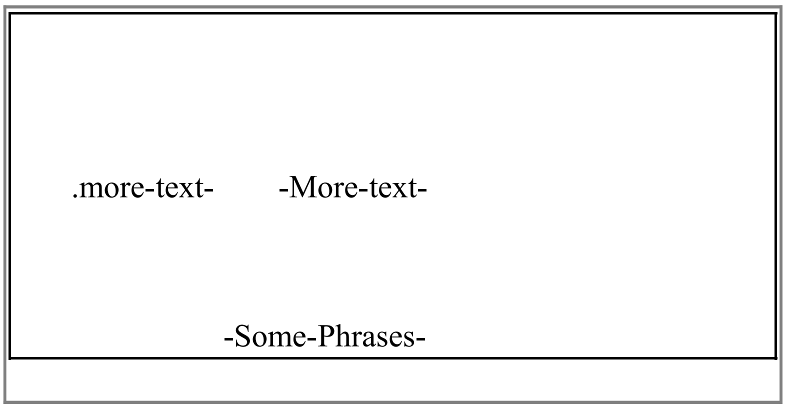
```
  2   +more-poems-

  3   *menu

  4   .some-phrases      -Some-Phrases-

  5   @

  6   Let us go for a walk

  7   Let us get some food

  8   Let us drink some water

  9   Let us stay home and watch television

  0   Liquor is good

Ta1   Liquor is bad

Ta2   Living is fun

Ta3   Don't you think so

Ta4   What is your name ?

Ta5   S=Sorry



  Speak F1  F2  F3  F4  Cursor F5  F6  F7  F8  Jump F9
```

Figure 1.   Screen image with margin menu. Text may be entered at the cursor @, located in paragraph -Some-Phrases-. Phrases are voiced through the menu in the left margin, if a user hits *6* to *Tab 4*. *1* reads in paragraph -More-Text-, *2* reads in file more-poems-.spr and discards the current text. The line *Menu is accessed by *3* and represents a hypertext link to a paragraph -Menu-. Paragraphs are also selected by wordprediction. Underneath one sees uppercase *S* defined as "Sorry "; if a user enters S the machine will speak Sorry. Meaning of function keys is displayed at the bottom. *F1* speaks the word to the left; *F3* speaks the phrase to the left; *F5* moves the cursor one word; *F9* moves to -Notebook-.



```
    .more-text-      -More-text-

    +poetry

                 -Some-Phrases-

   l@

     1.  Liquor

     2.  Living

 , ,  3.  Length

     4.  Left

  ..  5.  Let


     6.  Let Us Stay Home And Watch Television

     7.  Let Us Drink Some Water

 / /  8.  Let Us Get Some Food

Let Us Go For A Walk

F1.  Liquor Is Good
```

Figure 2. Same screen as figure 1, a user entered *l* and text prediction started up automatically. Two lists of suggestions are shown, the words above the phrases, long words first. *1* to *5* now select a word and period or comma select a recently used word. */* selects the same phrase as *8*, the one that was most recently used or was most recently read in.



```
    .more-text-      -More-text-


                 -Some-Phrases-
```

```
  lu@

,..,  1.  Liquor



      2.  Let Us Stay Home And Watch Television

      3.  Let Us Drink Some Water

/ /   4.  Let Us Get Some Food

      5.  Let Us Go For A Walk
```

Figure 3.   After *lu* only one word is shown and all phrases that start with L and contain a U. In this screen  F10 selects "Let Us ", two words from the phrase bound to /. "lugF10" will select "Let us get ". Space will select "lu ".

```
   Hello



   h@

 , , 1. Hello

  .. 2. How        **1**                  **2**

                SPARET a  b   m  n  q  r

 Ret  3. How Do You Do   ,  .  c  d   o  p  s  t

     4. How Are You     e  f  i  j   u  v  y  z

     5. How Is It       g  h  k  l   w  x  BS DEL

     6. How Much                   5

                 1  2  5  6  F1 F2  F5 F6

                 3  4  7  8  F3 F4  F7 F8

                ESCTAB 9  0 PGU  ↑  F9 F10

                SHIEND \  / PGD  ↓  ←  →
```

**3** **4**

Figure 4. Screen image with text prediction and a dynamic screen. In the centre of the square, characters "1" to "5" are suggested in turn to scan with a single switch. Each selection activates a quadrant, and again, until a single character is communicated to the rest of the software shown on the left. 'a' is selected by "121". "5" shows another square with upper case letters and other characters. At the top of the screen and to the left, the word "Hello " is seen and was presumably selected with a redefined 'H' by "5134". 'h' was then selected by "134" and text prediction was started up automatically. 'Ret' marks the most recently used phrase with 'h', "How do you do". This phrase is selected with 'RET', "112" or with '3', "313". ', ,' marks the most recent word, "Hello ", selected by comma, "113". "How " was used before "Hello " and is bound to period, "114".

```
Key_First_Letter  .
(Key_Next_Letter       +
 Look_At_Screen .
 Search_List    .
 (Amazement + Nothing + Think) .
 (Choose_Word         .
  Select_With_Numeral .
  Forget_Screen_Image     +
  Forget_Screen_Image .
  Continue    )       )*          .
(Verify_Result + Nothing)
```

Figure 5.   User process during classical word prediction. + means OR, . means NEXT and * means REPEAT. The choice  of different user processes depends on word length, number of characters typed in and individual differences.

```
Key_First_Letter  .
(Key_Next_Letter            +
 Select_Most_Recent        +
 Look_At_Screen .
 Search_List    .
 (Amazement + Nothing + Think)   .
 (Choose_Word           .
  (Select_With_Numeral  +
   Select_Most_Recent)  .
  Forget_Screen_Image   +
  Forget_Screen_Image   .
  Continue    )       ) *   .
(Verify_Result + Nothing) .
```

```
Prepare_for_Next_Word
```

Figure 6. User process during word prediction with recency binding.

```
Key_First_Letter    .
Key_Next_Letter *   .
Key_Space                  .
(Verify_Result + Nothing)
```

Figure 7. User process typing a new word.

```
Key_First
Key_Second
Key_Third   (*      machine now also presents words
                            from background database *)
Look_At_Screen
(Key_Rest_Of_Word       +
 Key_Some_Letters . Accept  +
 Key F6 . Key_Rest          +
 Improvise)
Verify_Result
```

Figure 8. User process to select a long word that is probably in the background database.

Table 1.

| Different | Frequency of | Number of | Total | Number of |
|---|---|---|---|---|

| Mechanisms | Mechanism | Words | Length | Keystrokes |
|---|---|---|---|---|
| Words typed in | 66 | 66 | 402 | 402 |
| Arrows etcetera | 174 | 0 | 0 | 174 |
| F1..F4, End, space | 10 | 10 | 40 | 10 |
| Redef & Upper case | 16 | 16 | 45 | 16 |
| Word prediction | 46 | 46 | 284 | 151 |
| Phrase prediction | 1 | 4 | 31 | 5 |
| Part of Phrase | 1 | 3 | 13 | 4 |
| Menu selections | 14 | 34 | 226 | 14 |
|  | ------ + | ------ + | ------ + | ------ + |
| Totals | 328 | 179 | 1041 | 776 |
| No arrows etcetera | 154 | 179 | 1041 | 602 |

Table 1. Data from logfile after 18 hours of practice, single finger typing, qwerty keyboard (EP). Note that menu selections (14) are more frequent than phrase predictions (1).

Table 2.

|  |  | Number of | Total | Number of |
|---|---|---|---|---|
|  | Frequency | Words | Length | Keystrokes |
| Words typed in | 135 | 135 | 630 | 630 |
| Arrows etcetera | 415 | 0 | 0 | 415 |
| Selection cursor | 7 | 31 | 160 | 7 |
| Redef & Upper case | 90 | 98 | 470 | 90 |
| Word prediction | 49 | 49 | 357 | 137 |
| Phrase prediction | 29 | 106 | 508 | 91 |
| Space repeated word, menu selections | 87 | 71 | 359 | 87 |

| | | | | |
|---|---|---|---|---|
| | ---- + | ---- + | ---- + | ----- + |
| Totals | 812 | 490 | 2484 | 1457 |
| No arrows etcetera | 397 | 490 | 2484 | 1042 |

Table 2. Data from a conversation between the author and his son, then aged 7 years. Qwerty keyboard, two- finger typing, after about 40 hours of practice. The author answered questions on the subject of how to program a computer. This particular discussion lasted twenty two minutes, 490/22=22 words per minute. Since, both logfiles and software have changed.

Table 3.

| Different Mechanisms | Frequency of Mechanism | Number of Words | Total Length | Number of Keystrokes |
|---|---|---|---|---|
| Words typed in | 381 | 381 | 2120 | 2120 |
| Arrows etcetera | 2773 | 0 | 0 | 2773 |
| F1..F4, End, space | 33 | 33 | 98 | 33 |
| Redef & Upper case | 287 | 301 | 965 | 287 |
| Word prediction | 392 | 392 | 2369 | 1317 |
| Phrase prediction | 57 | 169 | 1014 | 198 |
| Part of Phrase | 28 | 71 | 325 | 99 |
| Menu selections | 352 | 1854 | 11182 | 352 |
| | ------ + | ------ + | ------ + | ------ + |
| Totals | 4303 | 3201 | 18073 | 7179 |
| No arrows etc | 1630 | 3201 | 18073 | 4406 |

Recent words accepted: 223 times out of 392.

Table 3. Data from logfiles with single finger typing, qwerty key-board. Five conversations with a total time of 251 minutes starting after about 25 hours of practice (EP). Note frequent use of menu selections. With part of phrase is meant selection by F10 in figure 3.

Table 4.

| Different Mechanisms | | Frequency of Mechanism | Number of Words | Total Length | Number of Keystrokes | |
|---|---|---|---|---|---|---|
| Words typed in | | 130 | 130 | 709 | 709 | |
| Arrows etcetera | | 1041 | 0 | 0 | 1041 | |
| F1..F4, End, space | * | 238 | 313 | 1461 | 238 | * |
| Redef & Upper case | * | 297 | 307 | 1130 | 297 | * |
| Word prediction | * | 200 | 200 | 1293 | 795 | * |
| Phrase prediction | * | 99 | 321 | 1559 | 306 | * |
| Part of Phrase | | 14 | 37 | 172 | 51 | |
| Menu selections | * | 112 | 313 | 1890 | 92 | * |
| | | ------ + | ------ + | ------ + | ----- + | |
| Totals | | 2131 | 1621 | 8214 | 3529 | |
| No arrows etc | | 431 | 1621 | 8214 | 2488 | |

Total morse length: 10713 dits & dashes.

Recent words accepted: 185 times out of 200.

Table 4. Totals from five sessions with two key Morse Code from a qwerty keyboard after over 60 hours of practice (JV). Menu selections contribute 1890/8214 = 23.0% of the length. In five conversations, *-marked functions contributed over 10% of total length in 24 out of 5*5=25 instances. During text-prediction, a short-code (--) was bound to comma. Therefor, to accept a recent word was often less effort then to select by a numeral, with a code length of 5.

Table 5. Task decomposition.

| | | |
|---|---|---|
| 1. Short remark | . | Redefined upper case letter |
| OR | . | Short phrase (use FLUC) |

OR . Type words (with prediction or not)

2. Make statement

. Perhaps ask for patience through stored utterance

. Perhaps reformulate in terms of stored material

. Type in statement(s), using rate enhancements

. Repeat with function keys or margin menu

3. Quote story . Move to paragraph using either word prediction, cursor movement or hyper-text Select by the margin menu

OR . Use last-word based cursor navigation then select

OR . Phrase prediction for first line then cursor navigation then select

4. Repeat oneself . Margin menu or phrase prediction

5. Maintenance . Edit functions, block functions, file functions, typing and prediction

Table 5. A tentative global task decomposition.

Table 6.

Known comment ? YES: Select upper case letter (TAB then letter)

ELSE

Recently used word ? YES: Type first letter, then select using recency binding.

ELSE

Short word ? YES: Type in, ignoring text prediction.

ELSE

Known word ? YES: Type two letters then choose from list or improvise.

ELSE

Long word seems new YES: Type three letters and either choose from list, type on or improvise.

Table 6. Intermediate task decomposition. It is assumed that a user knows what word she wants and can not be distracted. The words in her file are the "known" words. After three keystrokes the list of suggested words is enlarged with words from the external database. Tab acts as a "sticky shift".

Table 7.

```
Type First Letter then

Evaluate What Word.


Evaluate What Word:

    Comment word?  YES Try next word

        ELSE

    Recent Word?   YES Accept by '.' or by ','

            ELSE

    Short Word? YES Type Rest

            ELSE

    Frequent Word? YES  Type Next Letter

                Select From List

            ELSE

    Not Frequent Word   Type Second Letter

            Type Third Letter

            Select From List  OR  Type Rest
```

Table 7. Intermediate task decomposition. It is assumed that the users also know that F8 interpretes the last few characters as if they had been uppercase.

Try Next Word represents a user that decides if the next word is a comment word too (in this case select it and repeat) or is not a comment word, in this case hit F8.

The advantage of Table 7 above 6 is that Type First Letter might partially overlap with the next stage: Evaluate What Word, and might share some resources like user's "short time memory".

With Try next word is meant that if the next word is comment as well, users may as well key in its first letter immediately.

Table 8.

1 'Yhv<look at screen><select "have" with numeral> '

2 'yh<look at screen> <Choose from:<select "you have " with numeral> or <do backspace and select you> >'

3 'Yhs<look at screen> <search for "have said ">'

4 'Yhave said Tbf<look at screen><select "before" with numeral>'

Table 8. Some possible selection sequences if one wants to select "You have said that before". "You " and "that " are bound to 'Y' and 'T', the user knows that all words are known to the system, and does not know if either "You have " is in it or "have said " is in it.

Table 9.

| Single predicate | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ |
|---|---|---|---|---|---|
| some_abbreviations | d | d | d | d | d |
| | dg | da | da | da | da |
| | dga | dg | dag | dg | dg |
| | | dga | | dv | dov |
| | | dag | | dov | dvo |
| | | | | .. | ... |

Table 9. Texts one could type using a single predicate to suggest "Dag goeden avond".

Table 10.

| | Number of | Total | Number of |
|---|---|---|---|
| Times | Words | Length | Keystrokes |

| | | | | |
|---|---|---|---|---|
| Words typed in | 273 | 263 | 1216 | 1216 |
| Cursor selections | 2 | 6 | 19 | 2 |
| Redefined Upper Case | 146 | 149 | 870 | 146 |
| Word prediction | 62 | 62 | 461 | 205 |
| Phrase prediction | 142 | 467 | 2307 | 488 |
| Menu selections | 0 | 0 | 0 | 0 |
| | ------+ | -----+ | -----+ | ------+ |
| | 623 | 947 | 4873 | 2057 |

Table 10. Data from logfiles after five small role plays done by the author early in development. Keystrokes for cursor movement and deletions were 110, not shown in this table and rather low. Ten times typing a word ended by deletion of the first few characters (273 - 263 = 10). Average overall word length = 5.14 (4873 / 947), word length of words selected with word prediction = 7.435 (461/62). The database for word prediction was quite immature, which only partially explains that many more words were typed in (263) than accepted from word prediction (62). In this version, the menu only started from the cursor downwards. Phrase prediction was therefor used to repeat phrases.

Table 11. Table from logfile.

| | | Number of | Total | Number of |
|---|---|---|---|---|
| | Times | Words | Length | Keystrokes |
| Words typed in | 65 | 64 | 301 | 301 |
| Cursor | 8 | 69 | 301 | 8 |
| Upper case | 67 | 67 | 350 | 67 |
| Word prediction | 27 | 27 | 182 | 79 |
| Phrase prediction | 34 | 107 | 525 | 115 |
| Menu selections | 4 | 12 | 78 | 4 |
| | ----- + | ----- + | ----- + | ----- + |

| | | | | |
|---|---|---|---|---|
| | 205 | 346 | 1737 | 574 |

Table 11. Logfile from an informal demonstration.

Table 12. Table from logfile.

| | Fre-<br>quency | Number of<br>words | Total<br>length | Number of<br>keystrokes |
|---|---|---|---|---|
| Words typed in | 174 | 184 | 823 | 823 |
| Arrows etcetera | 114 | 0 | 0 | 114 |
| Selections by<br>cursor | 0 | 0 | 0 | 0 |
| Upper case letters | 31 | 38 | 204 | 31 |
| Word prediction | 38 | 38 | 276 | 112 |
| Phrase prediction | 22 | 71 | 389 | 67 |
| Menu selections | 1 | 0 | 1 | 1 |
| | ------ | ----- | ----- | ----- + |
| Totals | 380 | 331 | 1693 | 1148 |
| Totals without<br>arrows etcetera | 266 | 331 | 1693 | 1034 |

Table 12. Data from three simple role plays after twelve hours of practice by a paid, healthy test subject (AT). Three tables were added up.