Published paper

Judge, S., Friday, M. (2011) *Ambiguous keyboards for AAC*, Journal of Assistive Technologies, 5 (4), pp. 249-256
http://dx.doi.org/10.1108/17549451111190650

Manuscript reference: JAT-12-11-09

Ambiguous Keyboards for AAC

Abstract

Purpose

'Ambiguous keyboards' and 'disambiguation processes' are becoming universally recognised through the popularisation of 'predictive text messaging' on mobile phones. As this paper shows, although originating in the AT and AAC fields, these terms and techniques no longer appear to be widely understood or adopted by practitioners or users. The purpose of this paper is to introduce these techniques, discussing the research and theory around them, and to suggest them as AT and AAC strategies to be considered by practitioners and users.

Approach

This is a conceptual paper that describes the use of ambiguous keyboards and disambiguation. The hypothesis of the paper is that ambiguous keyboards and disambiguation processes offer potential to increase the efficiency and effectiveness of AAC and should thus be considered further in research and practice.

Findings

The two broad methods for removing the ambiguity from the output of an ambiguous keyboard are presented. A summary of the literature around the use of disambiguation processes provided and the use of disambiguation processes for AAC discussed.

Key Words

Disambiguation; ambiguous keyboards; augmentative and alternative communication; communication aids; language entropy.

Background

Ambiguous Keyboards

An unambiguous keyboard is one where each key equates to (the same) single symbol output, an ambiguous keyboard is where each key equates to a number of possible symbol outputs. Ambiguous keyboards are now relatively common place thanks to the rapid growth of mobile phone text messaging using the numeric keypad[1].

Insert figure 1 here

Within the Augmentative and Alternative Communication (AAC) field ambiguous keyboards are commonly used as an input method – many AAC systems allow a user to say more things than there are keys. However, this is not a term that is commonly referred to in AAC practice or literature. In general, some confusion can exist in describing keyboards used as an input to AAC and it can be argued that input methods, including ambiguous keyboards, are poorly described and characterised in AAC literature and practice.

The output of keyboards within AAC can be graphemes (i.e. associated with a written language) or sounds (i.e. associated with spoken language). In this paper the output of a keyboard will be defined as a symbol to encompass both these possibilities. Keyboards are commonly considered to be an array of electro-mechanical keys, however keyboards are not restricted by their physical composition – within Assistive Technology and AAC it is common to use 'virtual' keyboards.

Keyboards, when used with communication aids and any system in general, can be characterised by a number of factors such those described in Higginbotham et al. (2007) such as form factor, feedback, representation method, learning requirement, cognitive load etc. Ambiguity is proposed as one of these characteristics.

[1] In 2005/6 79% of households in Great Britain had a mobile phone and 66% of adults in Great Britain had sent a text message (Office for National Statistics, 2007)

Unambiguous keyboards are exemplified by symbolic keyboards such as those used on AAC devices commonly described as ‘low tech’ or ‘medium tech’ communication aids, where each key is identified by a symbol and represents a single word or phrase. AAC keyboards commonly described as ‘dynamic screen’, ‘encoded’ and ‘multi-meaning’ can be described as ambiguous.

Disambiguation

The process of determining the un-ambiguous desired output from the user input to an ambiguous keyboard is called disambiguation. Disambiguation involves the user confirming the intended output and may involve a computer process that attempts to predict the intended output. Broadly speaking disambiguation could be described as either using coding (the user clarifies the output through further key presses) or involving a disambiguation process (the computer suggests the possible output for the user to confirm).

Coding

Coded entry requires a combination of key presses to define the output. This combination may be synchronous (i.e. pressed together) or asynchronous (i.e. pressed in order). When asynchronous, the (time) order of pressing may, or may not, be critical.

Coded keyboards include devices commonly referred to as ‘dynamic screen communication aids’, where the coding is often determined by the symbolic on-screen display and codified by semantic categories (e.g. food). Other communication aids also rely on coded input but codify the input differently, for example the Minspeak family of devices codify the words output using the semantic ‘double entendre’ of icons. Direct coding using switches (which can still be considered as a keyboard), including Morse Code, is also a recognised technique within AAC although possibly not widely used (Judge & Colven 2006).

Other coded keyboards exist and are used in different domains: ‘chorded keyboards’ are often cited and possibly little used; Stenotype keyboards are generally cited as being used in court recording or closed captioning and are an example of a synchronous coded keyboard that is codified by phonemes rather than letters; Morse code is an example of a single-key (switch) coded input method where the codification is achieved through timevariation (rather than combinations of keys); Multi-tap, used on mobile phones, where the key is pressed repeatedly until the desired letter is displayed, is another example of a coded input (involving time-based codification). These methods are well documented in the Human Computer Interaction (HCI) field for example in I. Scott MacKenzie & Kumiko Tanaka-Ishii (2007). Coding has been examined from the research perspective in the HCI field for example Perlin (1998) proposes a type of encoding for stylus entry.

Disambiguation Processes

A disambiguation process looks at the users input on an ambiguous keyboard and makes suggestions as to the desired output based on some pre-determined knowledge. A disambiguation process looks at the key presses you have made (for example, using the keyboard in Figure 1: ‘2abc’, ‘3def’, ‘3def’, ‘7pqrs’) and makes suggestions about what you wanted to say (‘beer’). The ambiguity is completely removed when you look at the word and confirm it is correct (or select an alternative word). The method described is the disambiguation process commonly used on mobile phones, the most common method of which is called T9 and patented by Tegic in 1995 (M. T. King et al. 1995). There are other ways of disambiguating and it is also possible to have any number of keys down to three, rather than the twelve keys commonly used on a numeric keypad.

Disambiguation processes rely on the fact that language has a certain amount of redundancy – about 50% in English. This is because the way we use letters is not random, it is partially defined by the structure of the language, for example in English, no words contain the string ‘pq’, not many contain ‘dr’ and lots contain ‘er’. These features of language were first identified by Shannon & Weaver (1963) who described the ‘Information Entropy’ of language.

Review of the Literature

A brief overview of the literature around the use of disambiguation processes is presented below with the aim of demonstrating the potential for the further integration of the technique into AAC systems.

Initial work on ambiguous keyboards was carried out in the early 80s and driven by the desire to allow deaf people to communicate over the telephone network. The telephone keyboard (or dial) presented a challenge to researchers wanting to use the system to transmit text. Glaser (1981) looked at encoding methods for the telephone keypad for deaf people – encoding required the user to dial two numbers for each letter – the user having looked up on a chart which numbers to dial for a certain letter. Johnson & Hagstad (1981) also looked at similar encoding methods, and are the first to consider the application for people with speech impairments.

Some authors at this time discuss language based disambiguation systems, however the methods they propose for removing ambiguity in the keypad use coding, not a language based disambiguation process. This work, however, formed the basis for the next developments in ambiguous keyboards – the use of a language models in a disambiguation process.

In his book Witten (1983) discusses the possibility of a word-level disambiguation system, although the system is not detailed. The book also introduces the concept of ambiguous keyboards combined with speech synthesis as an output (over the telephone network); and also discusses the 'disambiguation accuracy' for keystrokes on a telephone keypad. Disambiguation accuracy is defined as a measure of how many times the disambiguation system will correctly predict the desired word. Witten reported that only 8% of words in a 24,500 word dictionary would be ambiguous if using 9 keys for entry. Witten also proposed a feedback mechanism to allow the user to discriminate between possible duplicate words as well as describing other entry methods for an ambiguous telephone keypad – including the method known today as 'Multitap'. It can thus be seen that this book described many of the techniques now used by people using mobile phones to send text messages.

Although Witten describes a language based disambiguation method, he does not propose a particular system to achieve this. Minneman (1985) is the first to document a system for disambiguation of an ambiguous keyboard, describing a system developed called the 'T decoder' and reporting a disambiguation accuracy of 95%. The described system uses both word and character level disambiguation and also allows for the adding of novel words into the dictionary. Minneman does not present a method for distinguishing between ambiguous words but does refer to AAC applications, suggesting the use of the DECTALK speech synthesiser in the context of speech problems associated with hearing impairment. Minneman also describes a small study (n=12) comparing the disambiguation system to coded entry and 'Multitap'. The results of the study report a preference for the disambiguation entry method and a novice typing rate of 11 words per minute.

Minneman’s work was quickly built upon, with a number of authors investigating the topic (Kondraske & Shennib 1986; Sh Levine et al. 1987; Foulds 1987; Kreifeldt et al. 1989; Sh Levine & Goodenough-Trepagnier 1990; Arnott 1992). Areas investigated by these authors include using syllable level disambiguation (i.e. looking up pairs of syllables instead of waiting until the end of the word); the use of a ‘Retry’ button; optimising the layout of the letters on the keys; modelling input rates for different methods and varying the number of keys. Throughout this period there is a recognition in the literature that this work is orientated towards text-input or communication for people with disabilities.

The first patent related to disambiguation was registered by Tegic in 1995, the company that license the T9 disambiguation technique (M. T. King et al. 1995; J. King et al. 1995). C. Kushler (1998) and James & Long (2000) describe the T9 system, explicitly mentioning AAC as an original motivation for its development. More recently the rising importance of mobile communications especially for short text messages has made such systems familiar to many more people – reviews of mobile text input methods are available (Starner 2004; Johansen & J. P. Hansen 2006; I. S MacKenzie & Soukoreff 2002). Research into the area has mostly moved into the field of Human Computer Interaction and the interest in language based disambiguation for AAC/AT has reduced. Recent research into disambiguation processes can be grouped into a number of topic areas:

Word and character level disambiguation: Although the very first systems (Minneman, 1985) proposed word-level systems early research tended to be on character level systems since these used less memory. The most prominent current mobile phone disambiguation techniques are based on word level disambiguation where the probability of words, not characters, are stored. Recent work has tended to examine word-level systems looking at improving disambiguation accuracy (e.g. Gong & Tarasewich, 2005) although there is some investigation into character or prefix based disambiguation (I. Scott MacKenzie et al. 2001; Zielinski 2006).

Letter distribution*:* Conventionally, letters are arranged alphabetically on a telephone keypad, this arrangement is purely conventional and is not optimal for using a disambiguation process. Foulds (1987) demonstrated some small changes in typing rate using alternative layouts on a standard phone keypad. Significantly larger changes between layouts are noted however as the number of keys is reduced (Venkatagiri, 1999).

In addition, optimising layouts on reduced key sets can allow ambiguous text entry to approach 1 keystroke per letter (Arnott, 1992).

Corpus**:** The generation of word frequency tables – on which disambiguation relies - requires a corpus (examples of text) to be examined, and the choice of corpus can greatly alter input efficiency (K. Tanaka-Ishii et al. 2002). Ideally a large representative volume of texts should be considered, but in some cases it is better to examine fewer documents more directly related to the domain in which the user will be writing – be it conversation or a scientific article.

Man-Machine Interface: Finally overall good design must not be forgotten: as Johansen & J. P. Hansen (2006) point out although theoretically efficient key sets and selection techniques can be used often more humble factors of interface design can be more important in determining input efficiency.

The use of disambiguation processes for AAC

Although the ability to use disambiguation processes does exist in some AAC software, there is little reference to its use for AAC in practitioner literature, AAC research literature or from anecdotal evidence.

The main research looking at the use of disambiguation within the field of AAC has been by Harbusch & Kuhn (2003) who developed a disambiguation system that can be used with switches and scanning and claimed to out perform equivalent non-ambiguous keyboard layouts. Other work by the authors using this system investigated the use of different layouts and disambiguation methods (Hasan & Karin Harbusch 2003; Karin Harbusch et al. n.d.). Their system, UK0-II is available as open source software. Other authors have also, more recently built on this work, for example Mackenzie & Felzer (2010) have recently published the results of a user study of the use of an ambiguous keyboard with switch scanning.

Other recent research has been focused around the potential synergy between ambiguous keyboards and eye gaze as an input method (J. P. Hansen et al. 2001; J. P. Hansen et al. 2006). Use of a disambiguation process has also been proposed in a novel way within the PhonicStick device – a joystick that outputs spoken phonemes (Black et al. 2008) - in order to aid spoken language production using this device Trinh et al. (2010) propose the use of phonic disambiguation and/or word disambiguation with phonemic input.

It can be said that some of the ‘principles’ of disambiguation, or at least of the theory of entropy of language on which it is founded, can be found in some modern AAC techniques. However disambiguation does not appear to be accepted as a popular and widespread technique in its own right. ‘Traditional’ word prediction (Koester & S. P. Levine, 1994) and more recent methods of prediction such as Dasher (Ward et al. 2000) also rely on the entropy of language in their method of operation but do not use ambiguous keyboards.

The hypothesis of this paper is that disambiguation offers an appropriate method for consideration as an AAC entry method. A number of characteristics of ambiguous keyboards used with a disambiguation process are proposed as being relevant for AAC users:

- Reduced keyboard size – disambiguation has been popularised through use on mobile phones which have 12 keys. Reducing the physical size of the keyboard and the number of keys required to be accessed may be advantageous for many AAC users with motor difficulties.
- Minimising keystrokes – disambiguation processes strive to minimise the number of key presses per symbol output – this should minimise the effort required by a user.
- Familiarity of ‘texting’ – mainstream adoption of texting means that many people are familiar with the concept of disambiguation (if not the word itself). Increasingly, AAC users and their carers will have been exposed to and be familiar with the method.

Summary

Ambiguous keyboards and disambiguation processes have been researched and used in practice since the early 1980s where they were initially developed in the Assistive Technology and AAC fields. Having achieved ‘upward’ technology transfer into mainstream use, paradoxically disambiguation processes do not now appear to be a widely used as a technique within AAC.

Ambiguous keyboards are used within many AAC devices commonly known as ‘high tech’ however they are rarely referred to as such. This paper suggests that ambiguity should be adopted as a characteristic of an AAC keyboard as should the method of removing ambiguity – namely either coding or a disambiguation process. It is suggested that acknowledging these characteristics will help researchers and practitioners compare and contrast the benefits of various communication software or techniques. Finally it is also suggested that the use of disambiguation processes are reinvestigated in research and practice as a potential method for increasing efficiency and communicative effectiveness.

References

Arnott, J., 1992. Probabilistic character disambiguation for reduced keyboards using small text samples. Augmentative & Alternative Communication, pp.238-253.
Black, R. et al., 2008. Introducing the PhonicStick: Preliminary evaluation with seven children. In Montreal, Canada: ISAAC.
Foulds, R., 1987. Statistical disambiguation of multi-character keys applied to reduce motor requirements for augmentative and alternative communication. Augmentative & Alternative Communication, pp.192-195.
Glaser, R., 1981. A telephone communication aid for the deaf,“. Proceedings of the Johns Hopkins First National Search for Applications of Personal Computers to Aid the Handicapped.
Gong, J. & Tarasewich, P., 2005. Alphabetically constrained keypad designs for text entry on mobile devices. In Portland, Oregon, USA: ACM, pp. 211-220.
Hansen, J.P., Hansen, D.W. & Johansen, A.S., 2001. Bringing Gaze-based Interaction Back to Basics. Available at: http://www.it-c.dk/research/EyeGazeInteraction/Papers/Paulin_Hansen_Johansen_2001.pdf .
Hansen, J.P., Lund, H. & Itoh, H.A.& K., 2006. Gaze communication systems for people with ALS. In pp. 35-38. Available at: http://www.cogain.org/w/images/a/a2/ALS_Workshop_Yokohama2006.pdf .
Harbusch, K. & Kuhn, M., 2003. An Evaluation Study of Two–Button Scanning with Ambiguous Keyboards. In AATE 2003. AAATE. Available at: http://www.uni-koblenz.de/~harbusch/harbusch-kuehn-aaate-2003.pdf .
Harbusch, Karin et al., Topic- and Author-Specific Suggestion Lists for Typing with Ambiguous Keyboards. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.1545 .
Hasan, S. & Harbusch, Karin, 2003. N-Best Hidden Markov Model Supertagging to Improve Typing on an Ambiguous Keyboard. Available at: http://www.cs.rutgers.edu/TAG+7/papers/hasan.pdf .
Higginbotham, D.J. et al., 2007. Access to AAC: Present, past, and future. Augmentative and Alternative Communication, 23(3), pp.243-257.
James, C. & Long, M., 2000. Bringing text input beyond the desktop. In ACM Press, pp. 49-50.
Johansen, A.S. & Hansen, J.P., 2006. Augmentative and alternative communication: the future of text on the move. Universal Access in the Information Society, V5(2), pp.125-149.
Johnson, A. & Hagstad, R., 1981. DTMF telecommunications for the deaf and speech impaired,“. Proceedings of the Johns Hopkins First National Search for Applications of Personal Computers to Aid the Handicapped.
Judge, S. & Colven, D., 2006. Switch access to technology - A comprehensive Guide. , pp.1-56. Available at: http://ace-centre.hostinguk.com/assets/Product Downloads/SwitchScanningMaster_8_472.pdf .
King, J. et al., 1995. Small talk in adult conversations: Implications for AAC vocabulary selection. Augmentative and Alternative Communication, 11(4), pp.260-264.
King, M.T. et al., 1995. Reduced keyboard disambiguating system patent. Available at: http://www.google.com/patents?vid=USPAT6011554 .
Koester, H.H. & Levine, S.P., 1994. Modeling the speed of text entry with a word prediction interface. Rehabilitation Engineering, IEEE Transactions on [see also IEEE Trans. on Neural Systems and Rehabilitation], 2(3), pp.177-187.
Kondraske, G. & Shennib, A., 1986. Character pattern recognition for a telecommunication aid for the deaf. IEEE Trans Biomed Eng, 33(3), pp.366-370.
Kreifeldt, J., Levine, Sl & Iyengar, C., 1989. Reduced keyboard designs using disambiguation. Proceedings of the Human Factors Society 33rd Annual Meeting, 1, pp.441-444.
Kushler, C., 1998. AAC USING A REDUCED KEYBOARD. Available at: http://www.dinf.ne.jp/doc/english/Us_Eu/conf/csun_98/csun98_140.htm .
Levine, Sh & Goodenough-Trepagnier, C., 1990. Customised text entry devices for motor-impaired users. Applied ergonomics, 21(1), pp.55-62.
Levine, Sh et al., 1987. Multi-character key text entry using computer disambiguation. RESNA 10th Annual Conference, San Jose, California, pp.177-178.
MacKenzie, I. S & Soukoreff, R.W., 2002. Text entry for mobile computing: Models and methods, theory and practice. Human-Computer Interaction, 17, pp.147-198.
Mackenzie, I.S. & Felzer, T., 2010. SAK: Scanning ambiguous keyboard for efficient one-key text entry. ACM Trans. Comput.-Hum. Interact., 17.
MacKenzie, I. Scott et al., 2001. LetterWise: prefix-based disambiguation for mobile text input. In pp. 111-120. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.7.2834 .
MacKenzie, I. Scott & Tanaka-Ishii, Kumiko, 2007. Text Entry Systems: Mobility, Accessibility, Universality (Morgan Kaufmann Series in Interactive Technologies), Morgan Kaufmann.
Minneman, S., 1985. A simplified Touch-Tone telecommunication aid for deaf and hearing impaired individuals. Proceedings of the 8th Annual Conference on Rehabilitation Technology.
Office for National Statistics, 2007. Use of ICT at Home. Available at: http://www.statistics.gov.uk/cci/nugget.asp?id=1710 .

Perlin, K., 1998. Quikwriting: continuous stylus-based text entry. In ACM Press, pp. 215-216.
Shannon, C.E. & Weaver, W., 1963. A Mathematical Theory of Communication. Available at: http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html .
Starner, T., 2004. Keyboards redux: fast mobile text entry. Pervasive Computing, IEEE, 3(3), pp.97-101.
Tanaka-Ishii, K., Inutsuka, Y. & Takeichi., M., 2002. Entering Text with a Four Button Device. In Procs. of the 19th International Conf. on Computational Linguistics,. pp. 988-994.
Trinh, H. et al., 2010. Further Development of the PhonicStick: The application of phonic-based acceleration methods to the speaking joystick. In Barcelona: ISAAC. Available at: http://www.computing.dundee.ac.uk/projects/phonicstick/download/ISAAC2010_PS_Ha_et_al.pdf .
Venkatagiri, H.S., 1999. Efficient keyboard layouts for sequential access in augmentative and alternative communication. Augmentative and Alternative Communication, 15(3), pp.174-183.
Ward, D.J., Blackwell, A.F. & MacKay, D.J.C., 2000. Dasher - a data entry interface using continuous gestures and language models. In San Diego, California, United States: ACM, pp. 129-137.
Witten, I.H., 1983. Principles of Computer Speech, Academic Press, Inc.
Zielinski, P., 2006. Optimal codes for human beings. Available at: http://www.cl.cam.ac.uk/~pz215/files/t9.pdf .

Figure 1: A standard mobile phone keypad

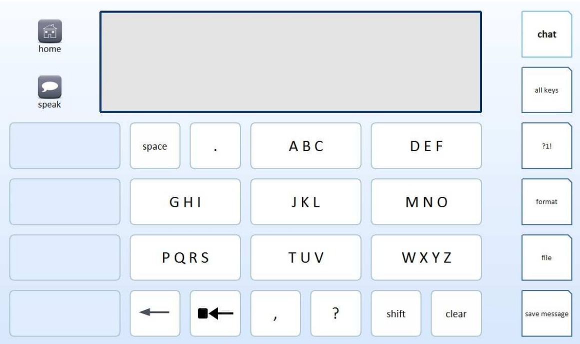
(source: http://en.wikipedia.org/wiki/File:Telephone-keypad2.svg – no copyright)

Currently Available Software Disambiguation Systems

It may be a surprise to some that a few AAC systems exist that can be used with a disambiguation process. Software AAC solutions available in the UK which allow disambiguation and that are known of by the authors are summarised below, no recommendation or preference is implied.

The Grid 2

http://www.sensorysoftware.com/thegrid2.html
http://grids.sensorysoftware.com/Members/Sensory/fast-talker



A disambiguation process has been added as an option to the commercially available 'The Grid 2' software. This was recently added (since version 2.7) as an option and has been implemented in one of the available grid sets called 'Fast Talker'. The disambiguation process can also be combined with word completion, phrase banks and other communicative features. The disambiguation keyboard can be customised (i.e. the letters on the keys can be rearranged).

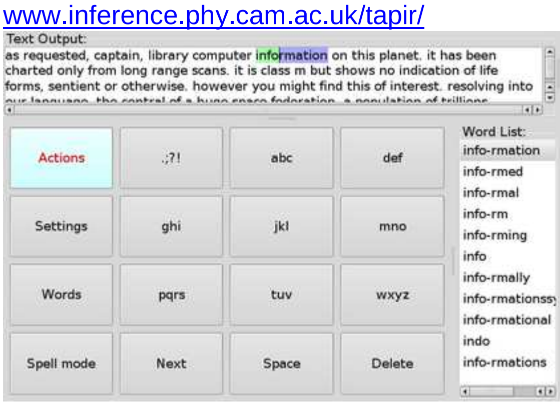
JayBee

http://www.jaybee.org.uk/Overview.html

A 'Predictive Keyboard' is available as one of the options within the relatively new commercially available 'JayBee' software. This uses a disambiguation process that is combined with the inbuilt word and phrase bank prediction.

Tapir

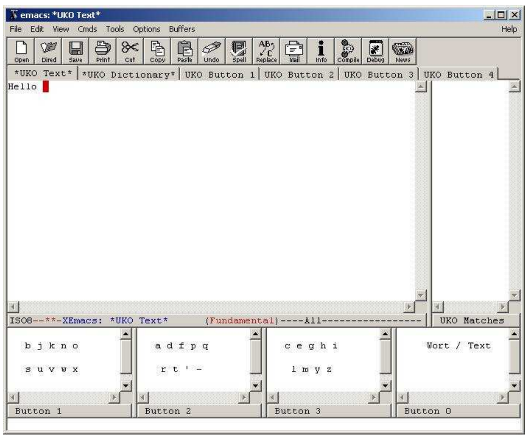
www.inference.phy.cam.ac.uk/tapir/



Developed by Piotir Zielinski of the Cambridge Inference Group, this system was optimised for use with eye-gaze, however it will work with any mouse input. It uses a novel form of disambiguation documented in Zielinski (2006), however it behaves as you would expect it to if you are used to the T9 method on a mobile phone. The main noticeable difference is the word list - making it seem more like 'standard' word prediction. This software is free and open source and will run on any operating system, however the down side of this is that it does not currently send text to other applications (you can, however, copy and paste the text). The window size can be dragged to any size/shape and font altered, however the keyboard layout and dictionary are not configurable. It is possible to switch to 'spell mode' for Multitap entry.

UKO-II

http://www.cogain.org/wiki/UKO-II



This software was written by Harbush and Kuhn for their research (Harbusch & Kuhn 2003). The software is predominately designed with switch input and a small number of keys (e.g. 4) in mind, however it can be used with mouse input too.

The free and open source software runs through xEmacs – an open source text editor, normally used in the Linux operating system. This makes it quite difficult to configure without technical support. Once setup, the character and key layout can be configured as can the input method and the dictionary.

Like Tapir, UKO also features a word list on the right hand side, text is sent to the text area of the screen and can be copied and pasted into other programs.

DKey

http://www.oatsoft.org/Software/dkey

This software allows someone to use the number pad keyboard as an ambiguous keyboard. This could be useful for people who are able to use a keyboard (maybe with a keyguard) but would rather not move across the full keyboard – e.g. people with tremor, ataxia, mouth stick users or people with weak movements such as spinal muscular atrophy. The keyboard layout and dictionary are configurable as are the keys used to access it. The window size, font size and colour can all be adjusted. The software is only accessed using a keyboard, and not using a mouse or scanning.