Augmentative and Alternative Communication: The Future of Text on the Move

Anders Sewerin Johansen & John Paulin Hansen

The IT University of Copenhagen,
Glentevej 67
DK-2400 Copenhagen NV
Telephone: +45 38 16 88 88
Fascimile: +45 38 16 88 99
{dduck, paulin}@it-c.dk

Abstract. The current methods available for text entry on small mobile devices suffer from poor performance which presents a potential barrier to acceptance and growth. Our analysis of mobile text entry indicates that the likely solution lies in aggressive use of language technology which is beyond the capabilities of current mobile devices. We argue that research in augmentative and alternative communication is highly relevant to the mobile text entry problem and offers the opportunity to research solutions that will be possible on the future generations of mobile devices.

1 Introduction - Mobile Texting

Mobile texting is a term for short text messages which are sent and received in a mobile setting, i.e. in situations where the user is not at the primary work setting (home or office) and may be engaging in another activity.

The most common example of mobile texting ("texting") is Short Message System (SMS). SMS is a part of the GSM standard for mobile phones and has gained popularity and users worldwide, especially in Scandinavia [1] where it is common to use SMS while on the bus, waiting in line, using ski lifts and even while strolling or riding a bicycle. The use of SMS is strongly correlated with age (primary user group is the 15-24 age group), use of the Internet and an active lifestyle [1].

Mobile text messages are primarily of a social nature and are commonly used as an alternative to phone conversations rather than data entry or creation of actual documents. Topics are thus typically of a personal character, and the content is in general actual conversation or coordination of social activities.

In contrast to actual documents, presentation (paragraphs, text justification and the use of different fonts) is relatively unimportant. Correct spelling, punctuation and the use of capital letters *is* a priority, but the main focus is swift and precise communication.

Slang and abbreviations is very common among Scandinavian SMS users. A Danish study conducted in December 2001 recorded the 600 most commonly used slang words and abbreviations used in a group of 700 SMS users [29]. The most common reason quoted for use of these shorthand notations is the low maximum length of SMS messages (<160 characters) and the performance characteristics of the input methods. Thus, the evolution of a domain specific language which parallels the evolution of codes and abbreviations used in Morse and among radio amateurs [13] in order to increase communication speed and lower costs.

1.1 On the Relevance of Mobile Texting

The low cost of an SMS message is often quoted as a major factor in the success of the system [2]. This is, in fact, not likely to be a contributing factor as the price of an actual SMS conversation, which usually spans several SMS exchanges, may exceed the price of a short voice call, and indeed in most countries it does.

The reason is more likely that text as a media complements speech in several areas. The most obvious of these are presented in Table 1.

Table 1. Qualitative and social differences between texting and voice conversation.

	Mobile texting	Voice call
Qualitative	Asynchronous/not interactive	Synchronous/interactive
	Unambiguous	Ambiguous
	Easy/cheap to archive	Hard/expensive to archive
	Easy to search/skim	Hard to search/skim
Social	Not impeded by noise	Demands quiet environment
	Discreet	May be overheard

It bears mentioning that many of these arguments are known from the use of email – the Internet's first "killer application". On this basis, we believe that mobile texting is a distinct media for communication just like email and regular letters rather than a substitute for voice calls.

In many situations, just one of the listed attributes may determine that texting is preferable to voice calls. A few examples from daily life is the asynchronous exchange of random thoughts ("Remember to pick up the kids after work"), unambiguous exchange of data (phone numbers, account numbers, addresses...) and the discreet exchange of personal messages while in a public setting ("I love you!").

1.2 The Mobile Input Problem

The three primary commercially successful text input methods are the standard SMS Multi Tap system, the pen based Graffiti used in the Palm Pilot PDA and the scaled-down QWERTY soft keyboard.

Multi Tap is the standard input system in the GSM mobile phone standard. It is described and evaluated in [3] and [4]. In brief, a Multi Tap user selects the desired letter from the International Telecommunication Union (ITU) standard phone keypad [14] by pressing the key marked with the letter one or more times. Selecting two letters residing on the same key (e.g. 'a' followed by 'b') requires the user to wait for a time-out (1.5 seconds on Nokia phones) or to use a "time-out kill" key.

Graffiti is the primary entry method for the Palm Pilot PDA. It is a simplified version of the roman capital letters, which reduces all English letters, signs and numbers to a single stroke. Graffiti and its immediate usability are examined in [5].

The QWERTY soft keyboard is available as an alternative input method in the Palm Pilot PDA. The features and performance of soft QWERTY is discussed in [6].

Unfortunately, all these input methods achieve rather low input performance, in the range of 8-30 words per minute (WPM), which is far below the rates commonly seen on the standard full-size QWERTY keyboard as shown in Table 2. Please note that expert QWERTY users are known to exceed the quoted speed by a comfortable margin, and that speech in ordinary conversation routinely reaches WPM rates in excess of 150 WPM. In other words, the numbers quoted here are indicative of the performance achievable on the above-mentioned input methods but not conclusive.

Table 2. Predicted and measured input speeds of common mobile and desktop text input methods

Input method	WPM predicted	WPM measured
Multi Tap	27.2 [4]	7.93 [3]
Graffiti	-	20 (estimate) [7]
Soft QWERTY w. stylus	8.9-30.1 [6]	-
QWERTY	-	64.9 [8]
Speech recognition	-	39 [9]

All these mobile text input methods are fairly easy to learn, Graffiti being probably the hardest with an estimated learning period of 5 minutes needed to achieve 97% accuracy [5].

All of them do, however, handle non-English localization in an unsatisfactory manner. As an example, QWERTY relegates the Danish letters 'æ', 'ø' and 'å' to the undesirable far right position (right hand little finger), and Multi Tap requires 8, 7 and 6 key presses respectively to enter these characters. As for Graffiti, entering several western European special characters and accents, including 'ø' and 'å', requires two strokes, whereas entering any English letter only requires one stroke.

In the case of QWERTY and Multi Tap, these issues are not possible to resolve gracefully as it is impossible to place these letters in a manner that corresponds to their likelihood without changing the position of several other letters. In the specific case of Multi Tap, this would require diverging from the ITU standard telephone keypad. The localization issues in Graffiti could certainly be resolved and are likely due to a less than careful implementation.

1.3 Full-Size Keyboards and QWERTY

The full-size QWERTY keyboard is the most common text input device. The primary characteristics of the QWERTY keyboard are:

- Direct selection of all characters.
- Can be operated with one or two hands and without visual attention (touch-typing).
- Performance exceeds 100 WPM with training.
- High performance (touch-typing) requires the full use of all 10 fingers.

- Shallow learning curve.
- Acceptable performance when used sub-optimally (hunt-and-peck estimated at ~23 WPM [27]).
- Well-known paradigm for text entry (one character at a time).
- Only two modes (shift) when entering ordinary text.
- Suitable for text entry, navigation and editing.
- Can be used by left-handed users.
- Just as easy (or hard) as handwriting for dyslexic users.
- Static layout.

When one considers the features listed, it is not surprising that the QWERTY has remained the standard text entry device despite competition from several alternative keyboard layouts – e.g. Dvorak [26] – which arguably are superior with regards to performance, learning curve and ergonomics. The QWERTY keyboard is flexible, efficient, easy to learn and forgiving for the novice user. If used with modern word processor software and spell-checkers, most errors are found and corrected at the earliest possible moment.

The primary limitations of the QWERTY keyboard are the limited mobile usability, the relatively high demands on motorical performance for expert users and an upper limit on performance. However, in the most common scenario, namely text composition, an entry speed of 40-50 WPM is usually adequate, and this level of performance is in the reach of most users with a minimum of training.

When studied from an information theory-based point of view, as seen in [22], the QWERTY keyboard has far too many keys for the task at hand. The information content (entropy) of English text is approximately 1 bit per character. One selection on a QWERTY keyboard signals approximately 6.3 bits of information indicating that the QWERTY keyboard is inefficient by a factor of 6!

1.4 The Mobile Challenge

We, along with other researchers [2], believe that the performance limitations on the primary mobile text input methods are a barrier to the continuing use and proliferation of mobile texting beyond the largely young user group that is currently the primary SMS users. In the field of web-design, it is commonly held that difficult or slow operation limits the users inclination to activity [30]. If mobile texting can be made more efficient and/or easier to use, it is likely to increase the usage. The goal is to approach the performance (i.e. ~40 WPM for casual users) and relatively low cognitive load of the common QWERTY keyboard but in a mobile setting while still making mobile texting accessible to new and casual users.

2 Mobile Text Input: Theory, Technology and Design Principles

Entering text messages can be viewed as a system consisting of a usage scenario, a user, a user interface for entering the text, a device hosting the interface and finally the actual text. In the following sections, we study these elements in order to identify the factors which are likely to be decisive in developing the replacement for current inadequate mobile text entry methods.

2.1 The Mobile Scenarios

As mentioned in the introduction, there is no one, single mobile usage scenario. The mobile text user desires the ability to communicate anywhere and anytime in the fashion of voice calls or ordinary conversation. Thus, the mobile scenarios span a wide range of physical limitations on motorical performance and the degree of attention available for the task.

We believe that these four scenarios are a fair representation of most mobile texting situations.

Table 3. Mobile texting scenarios.

Scenario	Posture	Physical freedom	Degree of attention	External interruptions
”Business class”. Near-office conditions. Long journey by plane or train	Seated, table space available	Full mobility for both hands and arms. Device can be placed on table	Full	None or few
”Coach”. Limited office conditions. Long journey by bus	Seated, no table available.	Limited mobility for both hands and arms with device in lap *or* full mobility for one hand and arm, other hand holding the device	Full	None or few
”Metro”. Out of office conditions. Short journey by bus or subway or standing in line	Standing.	Full mobility for one hand and limited mobility for the arm, with the other arm and hand holding the device	Limited	Often. Noisy environment.
”Strolling”. Full mobility in active environment. Walking down a busy street	Standing/ walking	Limited mobility for one hand and arm. Same hand holds and operates the device.	Very limited	Very frequent. Noisy environment

The mobile text user usually has the option to “upgrade” at will (or need) to a scenario that allows for a higher degree of motorical freedom and available attention, e.g. by sitting down on a public bench.

In order to support the most extreme scenario – “strolling” –, a user interface must perforce be very robust in the face of constant interruption and should preferably be usable in “eyes-free” mode, i.e. without demanding constant or frequent visual attention.

2.2 Factors in User Performance

When attempting to optimize or model user performance on some task, such as data entry, one common approach is to construct a model of the task and estimate user performance – either on the basis of a modeling framework such as GOMS, or by using rules for estimating motorical and cognitive components of the task. For the latter purposes, the most commonly used rules are Fitts' law [16] which attempts to predict expert users' motor performance on target selection, and Hick-Hyman [17] [18], which attempts to predict the time used on visual scan in target identification.

The predictions obtained from Fitts' law are dependent on the relative size and relative placement of the targets and the likelihood of transitions between the targets.

Predictions using Hick-Hyman are dependent on log2 of the number of available alternatives. This seems rather optimistic as a log2 dependency on number of items indicates a very efficient search algorithm, i.e. binary search, which requires a sorted list of items to search in. This is supported by user comments recorded in post-interviews in the FOCL study [11]. These indicate that the actual strategy for visual search was a combination of gestalting the area surrounding the current center of interest – a 3x3 matrix of letters – and using a linear search (which is dependent on the number of items) as a backup strategy.

Be that as it may, none of these criticisms dispute that the visual search component is in some way dependent on the number of items. This is therefore likely to be an important factor in dynamic or semi-dynamic input systems.

All in all, the general conclusion is certainly that the optimal system presents as few targets as possible with a function as close to user expectations as possible, ideally placed as close as possible to the current area of interest and possibly sized according to likelihood.

2.3 Language, Text and Language Technology

Text is highly redundant as can be seen from the fact that one usually achieves compression rates of 80-90% when compressing text with programs such as bzip and WinZip. Although we usually store text using 8 bits per character, experiments by Shannon [19] indicate that the entropy of English text is between 0.6 and 1.2 bits per character.

As mentioned in Section 1.4, this means that on average a very low input rate is required in order to compose text. This, however, is only possible if the device used for text entry is able to interpret this information correctly in effect inferring the other 7 bits from general knowledge of language and context using a *language model*.

Language models can be based on many different Natural Language Processing (NLP) algorithms with varying results in terms of performance, capabilities and resource consumption. A simple language model (frequency count) achieves an entropy of 4.03 bits per character [19]), and an advanced state-of-the-art model (maximum entropy) achieves 1.2 bits per character [20]. The traditional n-gram approach is relatively high performing at 1.5 bits per character [21]. David J. Ward tabulates the performance of most current approaches and discusses the construction of language models extensively in [22].

In general, language models come in three qualitatively different orders which are presented in Table 4.

Table 4. Different types of language models and their resource consumption.

	Technology	Typical usage	Resource consumption
Static analysis	Tables of letter and n-gram frequencies	Design of static and semi-static keyboard layouts [11]	None or very little
Simple interactive NLP	n-grams	Word prediction in text with no or few errors, dictionary-based input systems	Megabytes of RAM for storing dictionaries and language models
Advanced interactive NLP	Hidden Markov models, probabilistic parsing	Speech input, cursive handwriting input	Tens of MB of RAM for storing dictionaries and language models, heavy CPU usage

Most language models need to be primed in order to perform optimally. In the case of a 3-gram model, which predicts a word on basis of the two preceding words, it needs to be primed with two words in order to supply any predictions at all. In other words, most (possibly all) language models will supply unreliable predictions for the first few words or characters of input. As a consequence, when designing text input systems based on language models, one must allow for the possibility that the predictions are wrong!

One of the most common problems when using word-level language models is the dictionary problem: What happens when the user wants to enter a word which is not in the dictionary? It is possible to estimate the probability of an unknown word, but it is obviously impossible to predict the actual (unknown) word. It is not just impractical to add all known words to the dictionary (Websters Ninth New Collegiate Dictionary boasts almost 160.000 entries!) – it is virtually impossible as new words are constantly added to the vocabulary of all living languages.

As mentioned in the introduction, the Scandinavian SMS users do in fact use a very large number of slang words and abbreviations, many of which are local to the SMS domain. It is therefore to be expected that mobile texting is significantly more prone to the dictionary problem than the average newspaper article or novel.

2.4 The Mobile Text User Interface

Mobile devices are subject to several unique UI-related factors which make designing mobile solutions even more of a challenge than the more common desktop solutions. In [23] it is pointed out that mobile units are subject to extreme internationalization issues; a large proportion of new, novice or casual users; a lack of established metaphors and limited physical real estate for input, output and labeling of input devices. In Section 2.1 we discussed the usage scenarios which are rather more extreme than the usual office scenario.

The user population for mobile texting is to a large degree composed of new users, novices and casual users as indicated by the demographic breakdown of SMS usage patterns data in [10] and the simple fact that SMS is a very young media.

Research in UI design has led to the formulation of a series of general guidelines [25] [31]. The primary guidelines that apply to the mobile texting task are:

- Avoid a steep learning curve.
- Design the system in a fashion that enables the user to easily construct a mental model, preferably one that has similarities with comparable and well-known tasks in other domains.
- Give early, precise and clear feedback on operator errors.
- The system should be efficient for the task giving the user the satisfaction of a job well done with a suitable tool.
- Enable frequent users to use shortcuts.
- Allow for human diversity such as color blindness, left-handedness and mild dyslexia/illiteracy.
- Avoid the use of modes.

In the cases where technical or design limitations mandate that one or several guidelines are ignored, one should do so with the guidelines in mind and soften the blow (if possible). This can be done e.g. by documentation, on-line help and guides ("Wizards"). Not doing so presumes that the need of the users to use your device is so acute that they are willing to put up with bad design and a long and trying learning period. This is usually not the case when a simple and well-known and/or significantly easier alternative is available even if this alternative delivers lower performance.

That this is specifically not the case with regards to mobile texting is indicated by the results from a survey of SMS users, who have tried and later discarded the T9 text input system in favor of the less efficient but well-known Multi Tap system [24]. The mean number of tries before discarding the T9 system was only 3.6! T9 uses the ITU keypad as an ambiguous keyboard, thus only requiring the user to press a key once for each letter in the desired word. The ambiguous input is then compared to a built-in dictionary, and the user is required to acknowledge the systems' interpretation of the ambiguous input, and in some cases select from several alternative interpretations. An interactive demonstration of T9 is available at http://www.tegic.com.

The reported reasons for discarding the T9 system indicate that one or several of the guidelines were violated as they show a lack of understanding of the text entry paradigm used in T9 and/or general operator difficulties. The top 5 reasons cited are: 45% of the 102 respondents deem the system "difficult/complicated"; 36% complain that the system does not supply the desired words; 29% say they don't know how to use it; 18% say it is "too slow" and 13% say they are comfortable and/or familiar with the old system. Both model predictions [4] and empirical measurements [3] indicate that T9 is substantially more efficient than Multi Tap, and the process of adding new words to the dictionary is relatively simple. One must therefore assume that the disgruntled users were primarily influenced by design-related factors, or that the designers failed to anticipate the potential problems and supply sufficient documentation and/or design elements which compensated for the potential problems.

2.5 Current and Future Capabilities of Mobile Devices

The currently available mobile devices are clustered around three basic form factors, which offer differing levels of performance and richness of features as shown in Table 5.

Table 5. Current classes of mobile devices and their available resources.

	Mobile phone	PDA	Laptop
Price	\$10-\$100	Hundreds of \$	Thousands of \$
Storage	<10 MB	Hundreds of MB	Tens of GB
CPU	<10 MHz	50-200MHz	~1GHz
Size	~3x10 cm	~6x10cm	~20x30cm
Input	Keys	Keys, touchscreen and pen	QWERTY + pointing device
Screen area	~10 cm2	Tens of cm2	Hundreds of cm2
Purpose	Communication	Organizer/data collection	Office tasks

In terms of performance and price, it is remarkable that each successively larger form factor offers an increase in available resources and price of approximately an order of magnitude maintaining a roughly equivalent price/performance ratio.

We can estimate the available resources in a given class of devices at a given point in the future with an acceptable margin of error by using Moore's "law". This rule-of-thumb says that the price/performance ratio for semiconductor-based devices doubles with a frequency of 18 months leading to an increase of two orders of magnitude over a period of approximately 10 years. In other words, we can expect the hand-held form factor to have the level of performance of a current (2002) notebook in approximately 10 years time (2012). The high-end notebook in 1992 did indeed sport a CPU speed of 16-32 MHz and 1-4 MB of RAM with the option of a few dozens of MB static storage, which is in the same order of magnitude as the current crop of mobile phones.

As new capabilities are added (eg. games, music playback), the input and output capabilities of the devices increase. A common feature in new models of both mobile phones and PDAs is selection devices such as scroll- and jogwheels suitable for e.g. choosing the next song to play from a selection of hundreds.

2.6 Theory, Technology and Design Principles – Discussion

It is indeed theoretically and technologically possible to devise an efficient input method for mobile texting as the required input bandwidth is low (~1 bit per character on average). This requires that the necessary resources for hosting an efficient language model are available. The resources needed for well-performing language models are already available on the smallest mobile devices (the phone form-factor) or will be soon.

The limiting factor is likely the quality of the design of the UI as results from investigations of current solutions indicate. This may in turn be dependant on factors that are related to the usage scenario and/or physical limitations of the device due to the form factor (which imposes limitations on the available input and output devices) rather than the available computing resources.

One obvious way of investigating the future UI for mobile text entry is experiments in the desktop segment as we can predict with reasonable confidence that a solution that demands the computational resources of a desktop is viable on even the smallest mobile device in only 10 years time.

3 Adaptive and Augmentative Communication – a Testing Ground for Future Mobile Text Input Solutions

In our research group, we mainly concentrate on developing new approaches to communications aids for severely physically disabled persons, i.e. augmentative and alternative communication (AAC). Our solutions are designed for use by people who suffer from amyotrophic lateral sclerosis (ALS), also known as Lou Gehrig's Disease. Our systems are built from off-the-shelf components, and one of our major sub-goals is to develop robust eye-tracking based AAC solutions. Our design motivation is detailed in [28].

This class of AAC solutions has several obvious parallels to mobile text input solutions, primarily:

- The need to do without QWERTY as an input device for text.
- Focus on producing conversation-like text.
- Focus on speed and accuracy rather than presentation.
- Focus on multimodal interaction in one unified interface.

The major difference is that we are not bound by the computational and (in some cases) physical limitations of current mobile devices (GSM mobile phones and first generation PDAs). This allows us to design solutions that use computational resources and storage 1-2 orders of magnitude beyond those available in the mobile setting.

Our initial goal was to re-think the usage- and user scenario (as reported in [28]) in the hope that this would lead to revolutionary rather than evolutionary advances. Our quantitative goal was to improve input speed by a factor of two from the 6-7 WPM found in contemporary AAC systems [15] to 10-15 WPM. At this point in time, we have developed a range of systems which incorporate language technology as a central design feature rather than as an enhancement of traditional operation. Our systems all share these features:

- Significantly fewer buttons/targets than QWERTY (4-10 buttons).
- Aggressive use of language technology.
- Multimodal input (mouse, keyboard, head/eye tracking).

3.1 Users and Usage Scenarios

Although there are many disabilities that lead to a need for an AAC solution, e.g. muscular dystrophy, strokes or spinal cord injuries, we focus our development on solutions suitable for ALS patients. ALS is a progressive neurodegenerative disease that attacks nerve cells in the brain and the spinal cord leading to progressive loss of motorical functions. The average life expectancy of an ALS patient from the date of diagnosis is approximately 2 years.

ALS is a suitable starting point for designing AAC systems as ALS patients progress through a series of stages (detailed in Table 6) which represent most other scenarios for AAC usage. The effect of designing for ALS patients is that the system is most likely usable by a wide range of other potential AAC users.

It is worth noticing that the physical limitations in the various stages of ALS are very similar to those imposed by the mobile scenarios shown in Table 3.

Table 6. Typical progression of ALS

	Symptoms	Input devices
1. stage	Fatigue is noticeable. Reduced mobility and strength in arms and hands. Often slurred speech	Keyboard with hand/arm rests and modified operation (sticky shift, no repeat)
2. stage	Fatigue is a factor. Unable to move arms due to lack of strength, but mobility is usually retained in one or both hands. Severely slurred speech, largely unintelligible to outsiders	Mouse, joystick, reduced keyboard (5-10 keys)
3. stage	Almost full lock-in. No speech function. Severely reduced mobility of all extremities	One or two switches, eye tracking
4. stage	Full lock-in	Eye tracking

For several reasons, AAC systems for ALS users must be designed with multimodal input and walk-up-and-use in mind:

- The user should be able to use the same system through all stages of the disease.
- Many ALS patients have little or no previous IT experience and are quite busy adapting to the severity of their situation.
- Several assistants who need to be able to help the user complete letters, edit text and use other functions in the program without having to learn an unfamiliar interaction method.
- Limited time and other resources among the specialist responsible for selecting, installing and configuring the system means that the duration and quality of user training is severely limited in many cases.
- The progression through the stages of ALS is gradual, and the fatigue factor often makes it necessary for the user to switch to a less efficient input method during the day.

As a final design constraint, it is vital to allow for the use of large fonts. Many common complications associated with ALS lead to reduced sight and altered color perception, and in many cases it is necessary to place the screen at a relatively large distance from the user in order to ensure accessibility for e.g. wheelchairs and assistants.

3.2 Traditional AAC Solutions for ALS Patients

The generic AAC solution for ALS patients is an on-screen keyboard which can be operated through the use of a pointing device (mouse, joystick, eye tracking...) or with the use of a single switch in "scanning" mode. When using scanning, a cursor moves automatically at a preset rate over the available targets, and a target is selected by the use of a switch which is wired to some motor function that the user is able to control reliably. Most on-screen keyboards integrate word completion or word prediction. It is usually possible to use the keyboard for text entry and navigation even when the program is configured for some other input modality.

Obviously scanning is slow which is the motivation to developing new and faster text entry modalities which are available to users who suffer from severe limitation on physical mobility as well as the implementation of a wide range of laborsaving functions in AAC systems

One alternative which, in theory, is promising is Morse code. Morse code is designed for efficient communication in a low bandwidth situation and assigns the shortest possible codes to the most frequent letters.

Unfortunately, Morse is a hard skill to learn, it is difficult to design software that gives meaningful visual feedback and it is no longer a common skill. Additionally, it is rather difficult for a computer to interpret single-switch Morse input reliably. One problem is determining the length of the dot ("dit") and dash ("dah") signals, and there is also a segmentation problem, i.e. determining where one letter ends, and the next letter starts. On the basis of the above-mentioned problems as well as measurements of input rates from actual users, a study that compared Morse input to mouthstick [12] eventually concluded that Morse is an unsuitable solution to the input problem, despite the potential for high performance.

3.3 Four AAC Solutions

As mentioned in Section 3, our quantitative goal was to achieve typing speeds in the 10-15 WPM range. At the earliest stage, we decided to base our system on the following design elements:

- The use of eye tracking as the primary input method. Eye tracking confirms to Fitts' law according to [33], and is the highest bandwidth input method available to all ALS patients.
- In order to use eye tracking based on cheap off-the-shelf components and to increase robustness to environmental factors, such as ambient light, and increase

the possibility of use in a mobile setting, we had a severe upper limit on the number of buttons/targets.
- The design should encourage the user to use the word prediction and completion function as this was the most likely way to increase typing speed.

A group of design students without initial knowledge of the AAC field were then given the task of suggesting suitable UI designs based on the available elements and limitations. Based on their suggestions, we decided to base our solution on a 4 by 3 grid of buttons (Fig. 1), two of which are used as a text editor window.

(A1)	(B1)	C1	D1
A2	B2	C2	D2
A3	B3	C3	D3

Fig. 1. Our basic layout. An editor window occupies the positions A1 and B1

Our three experimental solutions are all based on this layout, and we intend to implement a fourth solution based on our current research prototype which we describe in the following sections.

The programs can be operated by mouse or eye tracker and uses a built-in dwell time activation function to select targets when in eye-tracker mode. As opposed to most commercial eye tracking solutions, there is no repeat on activation. If the user desires to activate the same button twice, she has to move the cursor away from the button to re-activate dwell activation. This feature helps to avoid stressing the user by eliminating the so-called “Midas Touch” problem: Everything you look at gets activated.

In addition to the point-based interface, the QWERTY keyboard is also available to assistants as a possible input device for text entry and editing.

Our First Solution: Dynamic Soft-Keyboard with Letter Placement According to Likelihood. Our first solution consists of three modes:

1. The main letter entry mode, which features a dynamic 3 by 2 keyboard. It presents the currently most likely letters; buttons for backspace and space; and access to word prediction/completion mode and alphabetical letter entry mode.
2. Word prediction/completion mode which presents the current 8 most likely words in a 4 by 2 matrix and features access to alphabetical letter entry mode and the main letter entry mode.
3. Alphabetical letter entry mode which enables the user to select the desired letter in a two-stage process.

In the main letter entry mode, the most likely letters were placed according to the workings of the parafoveal vision, i.e. with the most probable suggestion in the center position (C2), and the suggestions placed according to probability in a clock-wise fashion around the center position (D2, D3, C3, B3 and B2). This was done on the assumption that users would quickly learn to anticipate the placement of the desired letter and then – in case the letter prediction did not supply this as the primary candidate – quickly be able to evaluate the other candidates with a minimum of eye movement. The A2 button displays up to 8 word completions/suggestions and selection switches to word prediction/completion mode. A space button was placed at position A3 and a backspace button at D1.

In the word prediction/completion mode, we arranged the word suggestions in keeping with the Western European tradition for visual search, i.e. with the suggestions occupying the positions A2..D2, A3..D3 with the most probable word at position A2, the second-most probable word at position B2, etc. After selection of a word, the program remains in word selection mode in order to encourage the user to use the suggestions for increased typing speed.

The alphabetical letter entry mode consisted of two sub-modes and required the user to first select a group of letters (e.g. “ABCDEFGH”, “IJKLMNOP”...) and then the letter. We deliberately kept this entry mode as simple as possible as this was the backup strategy for new and/or confused users.

The system is conceptually similar to the proposed FOCL deriviate shown in fig. 10 in [11], in the sense that it supports two distinct strategies for letter entry: A primary strategy based on probability and a secondary based on the well-known paradigm of alphabetical search for the desired letter.

Our Second Solution – Incremental Refinement of the Original Solution. Based on user comments and observations from a usability test with 25 subjects of the first solution, we decided to implement a second iteration which features a minor and a major improvement.

The minor improvement was to change the layout of the word predictions in the word selection mode. Many users remarked that they felt the most intuitive search strategy was A2, A3, B2, B3, etc., i.e. placing the predictions in a left-to-right order based on their likelihood.

The major improvement was a re-working of the dynamic letter 3 by 2 keyboard. Many users remarked on the obvious problem that selecting the same button twice in a row was very time-consuming which was an obvious and frequent problem as the most probable letter always occupied the same position (D2). Furthermore several users felt that the placement of letters was arbitrary, i.e. the strategy of placing letters in accordance with their current likelihood was not in keeping with the users’ expectations and led to time-consuming visual searches.

We therefore implemented a placement algorithm which assigns home positions to all letters as well as secondary and tertiary positions and so forth. The algorithm then attempts to place as many letters as possible in the most desirable positions while ensuring that the currently most likely letter is not placed on the position that was selected previously thus keeping the need to re-activate dwell time activation by leaving the button to a minimum, and at the same time attempting to keep the visual search component of operation to a minimum.

This solution is currently in external testing among ALS patients, and the initial response to the improvements has been favorable.

Our Third Solution – T9-like Ambiguous Keyboard. We also investigated another solution to the problem of unintuitive placement of letters – an ambiguous keyboard. This was partly motivated by the desire to offer the users a static placement of letters, which should eliminate the visual search component of operation and partly motivated by the reports of positive experiences with a highly ambiguous keyboard solution which featured a 6-button design, assigning only four buttons to letters [32].

The result was a major re-working of the main letter entry mode which was necessary in order to accommodate the dictionary-based entry method that is required by this type of ambiguous keyboard entry. We decided to split the letters in 6 groups by alphabetical order, i.e. "ABCDE", "FGHIJ", etc. We retained the placement and function of the word completion mode change button. The space button was replaced by a "end word + space" button, and the position previously used for access to alphabetical letter selection was used for a "end word + punctuation" button.

We added two modes: A mode for selecting the current word candidates and a mode for entering a word letter-by-letter in case the word the user was trying to enter was unknown to the system.

Initial internal testing in the research group gave positive feedback. The elimination of the visual search component of the letter entry task was felt to be a big improvement over the previous two systems, and the dictionary-based input paradigm was not seen as a barrier to productivity or initial acceptance.

A usability test with a group of 25 novice subjects was far less encouraging. Most subjects reported confusion with regards to the paradigm which was increased by the three different ways of entering a word (word completion and *two* ways to end a word) as well as the lack of an obvious way to check whether the intended button was indeed selected. This led to poor orientation in the current entry task (a word) which in turn made many subjects express concern about their ability to correct mistakes without deleting a partially written word and starting over.

Our tentative conclusion from this usability test is that most (but not all) of these problems can be handled by a re-design and minimal changes in functionality. Our main design error was probably that we had not devised a way of gently introducing the users to the unfamiliar text entry paradigm, which was compounded by the absence of a well-known backup strategy, i.e. alphabetical letter entry.

Although we did not get any comments on the size of the dictionary most of the other criticisms leveled at the T9 system in [24] were echoed by our subjects.

Our Fourth Solution – Maximal Ambiguousity Keyboard. Inspired by Shannon [19], the report on highly ambiguous keyboards in [32] and the positive internal feedback on our first ambiguous keyboard solution, we decided to try to take the ambiguous keyboard entry method to the logical conclusion: A two-key keyboard.

This solution has many potential advantages, both from a theoretical and a practical point of view. Theoretically this would be the ultimate in key-based text input, as it would virtually eliminate all motor and visual search components of the text entry process when operated with two fingers which would then only have to leave the home keys in order to select alternative interpretations of the ambiguous input at the end of word entry. From our primarily practical point of view it would allow us to free sufficient buttons to integrate the word selection mode in the main

letter entry mode, by making 5 buttons available for word suggestions, instead of the single button used to switch to word completion mode in our current systems.
We have implemented a mouse-operated prototype for initial internal testing, which uses left and right mouse button to signal letters 'a' to 'm' and 'n to 'z' respectively, and requires a point and click to select the intended word at word completion. Initial internal feedback is extremely encouraging, and as a consequence we intend to skip the planned second iteration of the original 6-key ambiguous keyboard solution, and instead design a 2-key ambiguous keyboard as out next experimental system.

3.4 Discussion of the Solutions

Iterative usability test using novice subjects has shown to be of great importance to our design process, as we, the members of the design team, are unable to estimate whether the systems satisfy the demands to walk-up-and-use usability. The comments from these novice subjects have turned out to be very useful as well as inspiring.

During the usability tests we have logged user performance. Individual performance has been in the range of 2.5 WPM and 6.8 WPM for the first 160 words. Our experience so far is that these performance figures from novice users are far from conclusive, as even simple UI design errors decrease user performance markedly. Thus, despite the fact that the 6-key ambiguous keyboard has been the worst performing solution so far, we have been able to gather sufficient relevant feedback from user comments that we feel confident as to which features should or should not be retained for the 2-key ambiguous keyboard.

3.5 Relevance to Mobile Text Entry

While we obviously feel that AAC is interesting in and of itself, we also feel that the parallels to mobile text input methods are clear. We are trying to solve the same problem, i.e. how to input text in a low-input-bandwidth situation in an efficient and user-friendly manner. Therefore it is unsurprising that we have to deal with the same problems as the designers of mobile text input methods, and it is also indicative that there may be a large potential for knowledge transfer between these to research areas.

As a supporting argument, we submit that all of our AAC solutions described in previous sections are potential mobile text entry methods, since they – given the ability to host a sufficiently powerful language model – could easily be implemented under the physical limitations imposed by even the mobile phone class of mobile devices. As mentioned in Section 2.5, the required computational resources for even advanced language models are likely available in mobile phones within 5-10 years.

On this basis we believe that designing and evaluating AAC solutions for physically disabled people is a valid and useful method to gain insights in, and relevant feedback on the possible mobile text entry methods that can be hosted on future mobile devices with increased computational resources.

4 Conclusion

Linking the fields of AAC and mobile text input has the potential to benefit both research areas. Researchers in the mobile text input field gain awareness of a potential testing ground for the interfaces that are beyond the capabilities of the current crop of mobile devices. The AAC research area would obviously benefit from increased attention and awareness.

The potential gain for future users of mobile text input is also significant as a thorough evaluation of potential input methods in an AAC setting would likely identify and – if possible – help avoid potential design pitfalls when implementing new input solutions *before* they are implemented in mass-market products.

1 Smoreda, Z., Thomas, F. ”Use of SMS in Europe”, analysis of data from EURESCOM P903 study "Cross-cultural attitudes to ICT in everyday life" performed October to December 2000, http://www.eurescom.de/~ftproot/web-deliverables/public/P900-series/P903/sms_use/w1-sms.html, verified 17. April 2002

2 MacKenzie, S., Zhang, Shawn X., ”An Empirical Investigation of the Novice Experience With Soft Keyboards” *Behaviour & Information Technology*, *20*, 411-418

3 James, Christina L. ; Reischel, Kelly M. ”Text Input for Mobile Devices: Comparing Model Prediction to Actual Performance”, *SIGCHI’01 March 31-April 4, 2001*, pp. 365-371, Seattle, WA, USA

4 Silfverberg, Miika ; MacKenzie, I. Scott ; Korhonen, Panu ”Predicting Text Entry Speed on Mobile Phones”, *Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2000*, pp. 9-16, New York: ACM

5 MacKenzie, I. Scott ; Zhang, S. ”The Immediate Usability of Graffiti” *Proceedings of Graphics Interface ’97*, pp. 129-137, Toronto: Canadian Information Processing Society

6 Soukoreff, R. William ; MacKenzie, Scott I. ”Theoretical Upper and Lower Bounds on Typing Speed Using a Stylus and Soft Keyboard”, *Behavior & Information Technology 14*, 1995, pp. 370-379

7 Zhai, Shumin ; Hunter, Michael ; Smith, Barton A. ”The Metropolis Keyboard – An Exploration of Quantitative Techniques for Virtual Keyboard Design” *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2000),* November 5-8, 2000, pp. 119-128, San Diego, California

8 Matias, E.; MacKenzie, I. S., & Buxton, W. “Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY” *Proceedings of the ACM Conference on Human Factors in Computing Systems – INTERCHI ‘93*, pp. 88-94. New York: ACM

9 Jecker, Diane ”Benchmark tests: Speech Recognition” *PC Magazine*, november 1999, http://www.zdnet.com/products/stories/reviews/0,4161,2385302,00.html verified 17. April 2002

10 A. T. Kearney/Judge Institute of Management ”Mobinet Index # 4”, February 2002, http://www.atkearney.com/pdf/eng/Mobinet_4_S.pdf verified 17. April 2002

11 Bellman, T. ; MacKenzie, I. S. “A probabilistic character layout strategy for mobile text entry” *Proceedings of Graphics Interface '98*, pp. 168-176. Toronto: Canadian Information Processing Society

12 Levine, Simon P ; Gauger, John R. D ; Bowers, Lisa D. ; Kahn, Karen J. : ”A comparison of Mouthstick and Morse Code Text Inputs”, *Augmentative and Alternative Communication, 1(2)*, 1986, pp. 51-55, Decker Periodicals Inc., Hamilton, Ontario, Canada

13 Wikipedia in http://www.wikipedia.com/wiki/Morse+code, verified 17. April 2002

14 International Telecommunication Union recommendation E.161, available from http://www.itu.int

15 Lesher, G.W. ; Moulton, B.J., ; Higginbotham, D.J.. “Techniques for augmenting scanning communication” *Augmentative and Alternative Communication*, 14, 1998, pp. 81-101, Decker Periodicals Inc., Hamilton, Ontario, Canada
16 Fitts, P. M. ”The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement” *Journal of Experimental Psychology 47*, 1954, pp. 381-391
17 Hick, W. E. ”On the Rate of Gain of Information” *Quarterly Journal of Experimental Psychology*, 4, 1952, pp. 11-36
18 Hyman, R ”Stimulus Information as a Determinant of Reaction Time” *Journal of Experimental Psychology*, 45, 1953, pp. 188-196
19 Shannon, C. E : ”Prediction and Entropy of Printed English” *Bell Systems Technical Journal* 30, January 1951, pp. 50-64
20 Rosenfeld, R ”A Maximum Entropy Approach to Adaptive Statistical Language Modelling” *Computer, Speech and Language*, 10, 1996, pp. 187-288
21 Tilbourg, H ”An Introduction to Cryptology”, 1988, Kluwer Academic Publishers
22 Ward, David J. ”Adaptive Computer Interfaces”, Ph.D.-thesis, Inference Group, Cavendish Laboratory, University of Cambridge, November 2001, http://www.inference.phy.cam.ac.uk/djw30/ verified 17. April 2002
23 Marcus, Aaron “Babyface Design for Mobile Devices and the Web” *Proceedings , Vol. 2, Human-Computer Interface Internat. (HCII) Conf.,* 5-10 Aug., 2001, pp. 514-518, New Orleans, LA, USA, Lawrence Erlbaum Associates, Mahwah, NJ USA
24 Gutowitz, Howard ”Barriers to Adoption of Dictionary-Based Text-Entry Methods: A Field Study”, submitted to CHI2002, September 2001 http://www.eatoni.com/research/field-study.pdf verified 17. April 2002
25 Schneiderman, Ben ”Designing the User Interface: Strategies for Effective Human-Computer Interaction”, 2. edition, 1992, Addison-Wesley Publishing Company
26 Dvorak, A. ; Merrick, W. L. ; Ford, G. C. ”Typewriting Behaviour”, 1936, New York: American Book Company
27 Wiklund, M. E ; Dumas, J. S. ; Hoffman, L. R. ”Optimizing a Portable Terminal Keyboard for Combined One-Handed and Two-Handed Use” *Proceedings of the Human Factors Society – 31st annual Meeting* 1987, pp. 585-589, Santa Monica, CA: Human Factors Society
28 Hansen, Paulin J. ; Witzner, Dan ; Johansen, Anders S. ”Bringing Gaze-based Interaction Back to Basics” *Proceedings of Universal Access in Human-Computer Interaction (UAHCI 2001)*, New Orleans, Louisiana
29 User Evaluation dot Com, http://www.userevaluation.com/ verified 17. April 2002
30 Krug, Steve ”Don’t Make me Think”, 2000, Que
31 Jordan, Patrick W. ”An Introduction to Usability”, 1999, Taylor & Francis, London
32 Kühn, Michael ; Garbe, Jörn “Predictive and Highly Ambiguous Typing for a Severely Speech and Motion Impaired User” *Universal Access in Human-Computer Interaction. Proc. of UAHCI 2001*, New Orleans, August, 5-10. Mahwah (NJ): Lawrence Erlbaum Associates
33 Miniotas, Darius “Application of fitts' law to eye gaze interaction” *CHI 2000 Extended Abstracts*, pp. 339-340, 2000, New York, ACM