An Interface for Reducing Errors in Intravenous Drug Administration

Frode Eika Sandnes

Institute of Information Technology, Faculty of Technology, Art and Design, Oslo and Akershus University College of Applied Sciences, N-0130 Oslo, Norway
Frode-Eika.Sandnes@HiOA.no

Yo-Ping Huang

Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan 10608, E-mail: yphuang@ntut.edu.tw

Abstract. Input errors occur with drug infusion pumps when nurses or technicians incorrectly input the prescribed therapy through the control panel. Such number copying tasks are cognitively and visually demanding, errors are easily introduced and the process is often perceived as laborious. Stressful working conditions and poorly designed control panels will further add to the chance of error. An alternative scheme is proposed herein, termed intravenous prescription phrase, based on dictionary coding. Instead of copying number sequences the user copies sequences of familiar words such that the cognitive load on the user is reduced by a factor of five. The strategy is capable of detecting errors and is easy to implement.

1. Introduction

The operation of medical devices, in particular the input of numeric values has received much attention [1]. Intravenous drug administration involves the input of rate, dose, time and volume of drugs prescribed by doctors and input errors can be lethal. Studies have shown that error rates in intravenous drug administration can be as high as 50-80% [2, 3]. The technical complexity of menu structures constitute one problem with such medical devices is [4]. Another key issue is the copying of digits such as when placing phone calls. Often this involves a number copying task where the number is read from some source such as a phone directory and input using the numeric keypad [5]. Airlines often use booking references which users input on self-service check-in terminals in the airport to obtain their boarding cards. These higher base numbers, such as base 36, often comprise combinations of letters and digits.

Shannon [6] identified that a decimal digit is approximately equal to 3 1/3 bits. Similarly, base 36 numbers are equal to 5.2 bits. One may argue that the increased information capacity gained through large base-digits does not justify the increased complexity of the copying tasks. One reason is that certain letters and digits look similar such as 0 - O, 1 - l, 2 - Z, 5 - S, 6 - G and 8 - B. Differentiating such symbols is even more difficult when the user has reduced visual acuity and given certain fonts. High base numbers appear as random strings. The lack of internal structure means that users are unable to resolve ambiguous looking characters. This problem is well known in the optical character recognition literature [7].

It has been found that number input errors are caused by either motor slips where the fingers do not perform as expected, recall slips where numbers are remembered incorrectly, or perception slips where the source number is read incorrectly [9]. These errors can result in digit substitutions, insertions or omissions. Most digit and text input studies operates with typical error rates of 5-10%, which means that up one in every 10 digits are likely to be incorrect. A study of the frequency of digits in drug infusion pumps showed that 0 is the most frequent digit, followed by 1 and 5, while 3, 4, 6, 7 and 8 are comparatively rare [10]. Numbers ranged in 1 to 5 digits in length with 3 digits being the most common. One class of errors termed “out by 10 errors” is caused by incorrectly entering a decimal point or a zero [11, 12]. Out of 10 errors greatly affect the magnitude of numbers and will have serious consequences.

Several studies have investigated various number input interfaces including touch based [13] numeric keypads, displays with individual incremental up-down buttons, incremental left-right/up-down buttons or 5-button interfaces [14] that are often attached to mobile equipment in ambulances, hospital wards, etc., and results show that the slower incremental up-down interfaces leads to fewer errors than numeric keypads [15]. Another approach is to avoid the input of digits altogether where the prescription is printed on bar-codes and input using bar-code readers [16]. However, bar codes require printing and they are less flexible, as the information cannot be communicated orally. This study focuses on the manual input of prescriptions.

2. Number copying challenges

Number copying tasks are problematic for several reasons. Miller’s limit on humans’ short term memory of 7+/-2 pieces of information [17] is often cited and subsequent studies have narrowed this limit down to a maximum of 5-7 pieces [18]. Studies involving memorizing phone numbers have shown that recall performance rapidly degrades beyond 6 digits [5]. A digit copying task therefore has to be split up into several read-input cycles, each cycle requiring alternating the visual attention between the source and the target. This is difficult if the source digits are not grouped such as 16 digit credit cards numbers presented groups of four. Without grouping additional effort is required to locate where to resume when shifting attention between the source and the target. Another issue is a lack of standard interface layout. If there is a mismatch in both information sequence and information location on the printed prescription and the target it will harder for users to pair source and target.

Second, digit copying tasks are error-prone and parity checks are unable to capture all errors.

Third, digit copying is perceived as time-consuming and laborious. Many portable laptop computers do not have numeric keypads and it has been demonstrated that the input of digits on QWERTY keyboards without numeric keypads are significantly slower than using full keyboards with numeric keypads [8]. It is occasionally necessary and practical to communicate number information from one individual to another orally via phone, such as if receiving a prescription from a medical doctor via phone. Each digit will have to be read out one by one and often repeated several times to confirm the correctness of the data. This further reduces efficiency and increases the chances of error and frustration.

3. Memory aids

One popular memory aid is to remember number sequences associated with word input on mobile keypad, for example 2326 m as ADAM. Instead of recalling the individual digits where each digit counts as one piece of information, the word is remembered counting as one piece of information. Unfortunately, not all number sequences have corresponding words as 0 and 1 are not assigned letters.

This paper proposes to use a fixed dictionary for encoding and decoding. Digit sequences are split into groups, where each group is converted to a linguistic word. The user is presented with a word sequence instead of a number sequence and will thus be able to copy more information per copy-input cycle. Group sizes of 5 digits are chosen as this relies on wordlists with 100,000 entries.

For example, 01234 56789 is first split into two groups of 5 digits, namely 01234 and 56789. Each number is then looked up in the wordlist (see Fig. 1). The digits 01234 could correspond to the English word “stir” and 56789 correspond to “galumphs” and the word sequence “stir galumphs” is presented to the user, who has the memory capacity to simultaneously hold both words in working memory while copying the words to the target. Next, the words are identified in the wordlist on the receiving end and the indices of the two words identified.

The wordlist can be organized such that the magnitude of the number correlates with the word length such that smaller numbers with many prefix zeros are assigned shorter words and larger number are assigned longer words. Another strategy would be to assign the most frequently used numbers shorter words.

A key advantage of linguistic words is that readers read words as whole units, while digit sequences are read individually. The reader recognizes the height signature of lower case letters. This greatly adds to the reading speed. Moreover, users are able to spot simple errors due to the internal word structure. Visual perception errors are reduced since words are read as one unit rather than individual unrelated units. Recall errors are also reduced for the same reason.

4. Error detection

The literature on spell checking classifies input errors as deletions, insertions or replacements [19]. This strategy proposes two levels of error detection. The first level of error detection catches misspelled words not found in the wordlist.

Next, the wordlist is organized such that similar words are associated with dissimilar digit sequences. The Levenshtein distance is often used to measure the distance between words [20], but the Hamming distance can be used if the words have equal lengths. The following scheme is employed. The list of 100,000 words is first sorted into groups of equal length and words in each group shuffled into random order. Finally, the groups are recombined into one list organized according to increasing word lengths while exhibiting a random internal structure.

For example, the word “buys” can easily be mistyped as “buts” by substituting the character y with t. The Hamming distance between these words are 1. The corresponding numbers for the two words are 1225 and 3022. These numbers have a Hamming distance of 3 and a numeric distance of 1797. With running numbers the two numbers would be 1352 and 1354, respectively, yielding a Hamming distance of 1 and a numeric distance of 2. Fig. 1 illustrates the wordlist construction process.

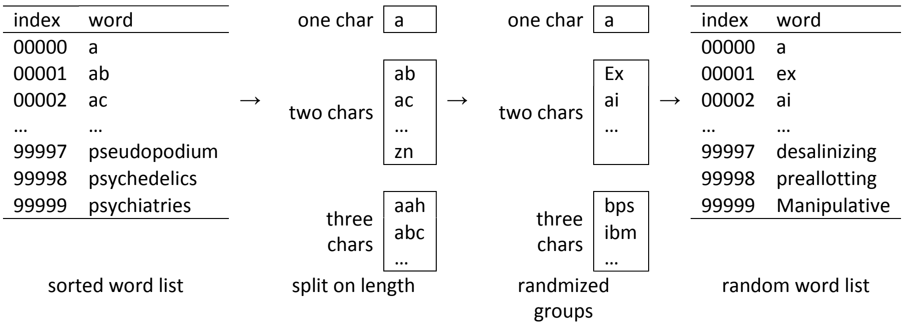

Fig. 1. Composing the error tolerant wordlist.

A second level of error detection is achieved by introducing a parity check [21] by summing each 5-digit chunk multiplied by unique factors and computing the desired modulus 100,000 of the total. By multiplying each number with a factor the parity check will detect if elements are swapped. A modulus 10 scheme misses 10% of the errors, while modulus 100,000 only misses 0.001% of the errors.

The linguistic digit representation can also assist detecting errors when comparing numbers. Imagine a customer comparing the dates 23-06-2012 and 23-08-2012, coded as 23062 and 23082, respectively. Each date comprises three information parts, day, month and year. If the user focuses on the day he or she may overlook the difference in month, that is, June versus August since the shape of 6 is similar to 8. This will not happen when comparing their linguistic representations “thrives” and “marxist”.

5. Information coding schemes

This section illustrates how to code values with the proposed scheme. Each unit should be fitted to a chunk to avoid overlap across chunks.

Large numbers with more than 5 digits are split into several chunks. Some values with more than 5-digits can be reduced to 5-digit chunks without information loss. Dates are often described with eight digits, namely day, month and year. One simplification is to use a modulo-10 single digit for the year. This will work as most applications operate within a limited time scope. This date format allows dates to be specified using a single linguistic word which is useful for making comparisons. Comparisons across years are simplified further if the year is dropped altogether.

Numbers comprising 5 digits or less can simply be represented using heading zeros such as small quantities, drug doses, etc. Analysis of commercially available equipment showed that numbers frequently represent volumes in ml or rates in ml/hr with rarely more than three digits and one digit after the decimal point. Such numbers can therefore be represented using the least significant digit of the 5-digit sequence to represent the digit after the decimal point and the four most significant digits to represent the digits on the left of the decimal point. For example 0.1 would be 00001, 10 or 10.0 would be 00010, 450 would be 04500, etc.

However, most drug infusion pumps do not actually have the precision indicated by the operating panels. Some equipment specifies an error in dosage of about 12% from the set value. It is thus pointless to distinguish between 450.1 ml and 450.0 ml.

Pairs of related parameters with similar magnitudes can be combined into 5 digit numbers as a type of double floating point numbers. A parameter is represented using its two most significant digits, that is A and B, for example AB0, AB or A.B, and 3 bits are used to indicate the position of the decimal point from 10-1 to 104 represented by the values from 0 to 6, respectively. For example to code the parameter pair 2.0 and 1.5 using this scheme one would get 02015. The first digit 0 indicates the decimal point in the most leftmost position. To code the number pair 45 and 35 the decimal is moved one position to the right and the first digit is therefore 1 yielding the value 14535. Similarly, the pair 450 and 350 can be coded as 24535.

6. Materials and Methods

An English wordlist published by the SIL International Linguistics Department containing 109,582 entries was used. The entries were sorted according to increasing word length and the 100,000 shortest words were kept. The average word length is 8 and the maximum words length 12 characters. The size of the final wordlist was less than 1 Mb making it applicable to devices with limited memory. A proof of concept coder and decoder were implemented in Microsoft Excel. Excel was chosen in order to demonstrate that it is simple to implement.Several drug infusion pump user manuals were studied to acquire information about common intravenous drug administration practices and prescription parameters, namely Curlin Medical's 4000 Plus and 4000 CMS Ambulatory Infusion Systems, Abbott Laboratories' PLUM A+, Eureka's IP and LF infusion pumps and BodyGuard's 323 Multi-Therapy ambulatory infusion pump. Common parameters are listed in Table 1. Of the less obvious parameters titration limit specifies the maximum infusion rate, keep vein open rate specifies the rate of fluid transmitted when the devices is in an open state. This measure is specified as the overall rate is achieved by injecting drugs in smaller doses, and this specifies the rate for each dose.

The following parameters apply to patient controlled analgesia therapies: Loading dose indicates an initial dose of drugs infused at the beginning of a therapy. Bolus dose indicates the amount of drugs in lm. Delta rate and delta time specifies the gradual increase of a dose per time unit. Maximum bolus dose indicates the maximum dose that can be demanded by the patient, minimum bolus interval specifies the minimum time interval between consecutive bolus doses and number of boluses allowed per hour specifies an upper limit to how many bolus doses the patient are allowed per hour. Up-ramp and down-ramp parameters can be used to gradually increase and decrease doses.

parameter	resolution	max	unit	type	Flag no.
Bag volume	1	9999	ml	N/A	
Concentration	0.1	999	mg/ml	required	
Start time	00:01	23:59	hours:minutes	optional	1
Titration limit	0.1	400	ml/hour	optional	2
Amount to infuse	0.1	9999	ml	required	
Rate of injection	0.1	400	ml/hr	required	
Vein open rate	0.1	10	ml/hour	optional	2
Loading dose	0	50	ml	PCA	3
Bolus dose	0	50	ml	PCA	4
Delta rate	0	50	ml	PCA	5
Delta time	0	60	minutes	PCA	5
Max bolus dose	0	100	ml	PCA	4
No. bolus dose/hr	0	15	doses	PCA	4
Min bolus interval	0	60	minutes	PCA	4
Up-ramp	00:01	99:59	hours:minutes	optional	6
Down-ramp	00:01	99:59	hours:minutes	Optional	6

Table 1. Common drug infusion pump parameters.

This parameter list comprise six optional parameters, therefore a check word can be introduced to indicate both which optional parameters that are included in the message and a parity symbol for error detection purposes. The two most significant digits are used to indicate the parameters included and the three least significant digits constitute a parity symbol.

7. Results and discussion

The following example illustrates how the proposed strategy is applied to the following simple fictitious prescription:

Concentration: 1.5 mg/ml
Amount to infuse: 75.0 ml
Rate of injection: 15.0 ml/hr

The corresponding number sequence for this prescription is:

00695 00015 00750 00150

Here 00695 is the control value where the first two heading zeros indicate that no optional parameters are present. The parity value 695 is computed as the modulus 1000 of the sum of the three parameters computed with weights 3, 2 and 1. The corresponding word prescription phrase sequence is

but md lip pub

Setting three digits may not seem hard, but it would probably require three read-input cycles. However, under stress one may misread one of the digits or swap the numbers that are relatively similar in structure and magnitude.

The corresponding four word phrase “but md lip pub” is easier to remember, and can be input in one read-input cycle. Moreover, any mistakes would be detected, for instance if the word “but” is incorrectly input as “byt” since y is next to u on the QWERTY-keyboard. The letter sequence “byt” would be detected as an invalid word.

Next, imagine the word “bit” input instead of “but” as the character i is next to u on the keyboard. The word bit gives parity 456, which indicates that it is a mistake somewhere since it does not match the parity value 695.

If two of the words are accidentally swapped, for instance md and lip, giving the incorrect phrase “but lip md pub”, the parity becomes 430 which does not match 695.

If we include an extra word accidentally, for example md twice, giving “but md md lip pub” the error is detected as there are no flags indicating optional parameters. Also, the parity values do not match. Next, consider the following prescription:

Concentration:	1.5 mg/ml
Titration limit:	30.0 ml/hr
Amount to infuse:	75.0 ml
Rate of injection:	20.0 ml/hr
Vein open rate:	10 ml/hr
Loading dose:	5.0 ml
Bolus dose:	1.0 ml
Delta rate:	1.0 ml
Delta time:	5 minutes
Max bolus dose:	10 ml
No. bolus dose/hr:	2
Min bolus interval:	30 minutes
Up-ramp:	03:00
Down-ramp:	06:00

This prescription involves 14 data items with 27 digits and 5 separators that possibly would have to be input through an intricate procedure involving different menu screen in 14 read-input cycles. This prescription is represented using the following string:

37983 00015 00300 00750 00200 0010 00050 00010 00010 00005 00010 00002 00030 00300 00600

All but the first parameters are included meaning that all bit flags apart from the first bit are set, giving 37. The parity value of the parameters is 983 using weights from 13 to 1 from left to right. The corresponding drug prescription phrase is thus

flipper md fen lip lap us od us et us ai on fen doz

The resulting 15 word phrase can be copied in three read-input cycles, as “flipper md fen lip lap”, “us od us et us” and “ai on fen doz”.

The two examples illustrate a very general and flexible approach. However, some values occur repeatedly such as 10 resulting in monotonous phrases with several similar (“us”). Since most drug infusion pumps do not actually have the precision indicated by the operating panels the final example illustrates the use combined parameter pairs for obtaining shorter intravenous prescription phrases. The following related parameters from the previous example can be paired, titration limit (30 ml/hr) and keep vein open rate (10 ml/hr) coded as 13010, loading dose (5 ml) and bolus dose (1 ml) coded as 05010, delta dose (1 ml/hr) and delta time (5 mins) coded as 10105, max bolus dose (10 ml), bolus doses per hour (2) and minimum bolus interval (30 mins) coded as 10230, and finally up-ramp (03:00) and down ramp (6:60) coded as 11836. The resulting string is thus

37271 00015 00750 00200 13010 05010 10105 10230 11836

which gives the following intravenous prescription phrase

cappers md lip lap vegans divvy nitty babes milton

This phrase is easily remembered in two steps, namely “cappers md lip lap” and “vegans divvy nitty babes milton”. This optimized phrase does not compromise accuracy.

8. Conclusions

A strategy for reducing the errors in intravenous drug administration is proposed. It comprises a memory aid that simplifies manual number copying tasks. The strategy converts the prescription data comprising digits sequences to sequence of linguistic words. Ordinary drug prescriptions can therefore be memorized for short durations. In addition the strategy allows users to easily verify the correctness of information as it is easier to compare linguistic words than digit sequences.

References

[1] Oladimeji, P.: Towards safer number entry in interactive medical systems. In Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems (EICS '12), pp. 329-332. ACM, New York (2012)

[2] Barber, N., Taxis, K.: Incidence and severity of intravenous drug errors in a erman hospital. European Journal of Clinical Pharmacology 59, 815–817 (2004)
[3] Taxis, K., Barber, N.: Ethnographic study of incidence and severity of intravenous drug errors. BMJ 326, 684 (2003).
[4] Nunnally, M., Nemeth, C. P., Brunetti, V., Cook, R. I.: Lost in Menuspace: User Interactions With Complex Medical Devices. IEEE Transactions on System, Man and Cybernetics Part A 34, 736-742 (2004)
[5] Raanaas, R. K., Nordby, K., Magnussen, S.: The expanding telephone number Part 1: Keying briefly presented multiple-digit numbers, Behaviour & Information Technology 21, 27-38 (2002)
[6] Shannon, C. E.: A mathematical theory of communication. SIGMOBILE Mobile Computing and Communication Review 5, 3-55 (2001)
[7] Kahan, S., Pavlidis, T., Baird, H. S.: On the Recognition of Printed Characters of Any Font and Size. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9, 274 – 288 (1987)
[8] Sandnes, F. E.: Effects of common keyboard layouts on physical effort: Implications for kiosks and Internet banking. In Proceedings of Unitech 2010, pp. 91–100. Tapir Academic Publishers, Trondheim (2010)
[9] Wiseman, S., Cairns, P., Cox, A.: A taxonomy of number entry error. In Proceedings of the 25th BCS Conference on Human-Computer Interaction (BCS-HCI '11), pp. 187-196. British Computer Society, UK (2011)
[10] Wiseman, S.: Digit distributions, What digits are really being used in hospitals? Proceedings of the Fourth York Doctoral Symposium on Computer Science, pp 61-68, The University of York (2011)
[11] Thimbleby, H., Cairns, P.: Reducing number entry errors: solving a widespread, serious problem, Journal of the Royal Society Interface 7, 1429-1439 (2010)
[12] Thyen, A. B., McAllister, R. K., Councilman, L. M.: Epidural Pump Programming Error Leading to Inadvertent 10-Fold Dosing Error During Epidural Labor Analgesia With Ropivacaine. Journal of Patient Safety 6, 244-246 (2010)
[13] Isokoski, P., Koki, M.: Comparison of two touchpad-based methods for numeric entry. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '02), pp. 25-32. ACM, New York (2002)
[14] Cauchi, A.: Differential Formal Analysis: Evaluating safer 5-key number entry user interface designs, EICS'12 Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems, pp. 317-320, ACM, New York (2012)
[15] Oladimeji, P., Thimbleby. H., Cox, A.: Number Entry Interfaces and Their Effects on Error Detection, LNCS, vol. 6949, pp 178-185. Springer, Heidelberg (2011)
[16] Meyer, G. E., Brandell, R., Smith, J. E., Milewski, F. J., Brucker, P., Coniglio, M.: Use of bar codes in inpatient drug distribution. American Journal of Health-System Pharmacy 48, 953-966 (1991)
[17] Miller, G. A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review 63, 81–97 (1956)
[18] Simon, H. A.: How Big Is a Chunk? Science 183, 482-488 (1974)
[19] Kukich, K.: Techniques for automatically correcting words in text. ACM Computing Surveys 24, 377–437 (1992)
[20] Navarro, G.: A guided tour to approximate string matching. ACM Computing Surveys 33, 31–88 (2001)
[21] Wagner, N. R., Putter, P. S.: Error Detection Decimal Digits, Communications of the ACM 32, 106-110 (1989)