Maximizing the Guessability of Symbolic Input

Jacob O. Wobbrock, Htet Htet Aung, Brandon Rothrock and Brad A. Myers
Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{ jrock, hha, rothrock, bam }@cs.cmu.edu

ABSTRACT

Guessability is essential for symbolic input, in which users enter gestures or keywords to indicate characters or commands, or rely on labels or icons to access features. We present a unified approach to both maximizing and evaluating the guessability of symbolic input. This approach can be used by anyone wishing to design a symbol set with high guessability, or to evaluate the guessability of an existing symbol set. We also present formulae for quantifying guessability and agreement among guesses. An example is offered in which the guessability of the EdgeWrite unistroke alphabet was improved by users from 51.0% to 80.1% without designer intervention. The original and improved alphabets were then tested for their immediate usability with the procedure used by MacKenzie and Zhang (1997). Users entered the original alphabet with 78.8% and 90.2% accuracy after 1 and 5 minutes of learning, respectively. The improved alphabet bettered this to 81.6% and 94.2%. These improved results were competitive with prior results for Graffiti, which were 81.8% and 95.8% for the same measures.

Categories & Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User interfaces—*evaluation/methodology, input devices and strategies, user-centered design*.

General Terms: Design, Experimentation, Measurement.

Keywords: Guessability, immediate usability, symbols, referents, proposals, gestures, commands, command-line, keywords, labels, icons, text entry, unistrokes, EdgeWrite.

INTRODUCTION

The guessability of a system determines a great deal about its initial user experience. It is unrealistic to expect that users will have the time or desire to undergo extensive training with systems, whether by tutorial, on-line help, printed manual, or human instruction. Thus, a user's initial attempts at performing gestures, typing commands, or using buttons or menu items must be met with success despite the user's lack of knowledge of the relevant symbols. This requires high guessability.

Guessability is particularly important in *symbolic input,* where users enter or access symbols to indicate associated *referents*. Examples of symbols and referents are stylus strokes that enter ASCII characters, command-line names that execute programs, and graphical buttons that access features. In these cases, users often know what referent they desire (e.g. the letter, program, or feature they want), but they do not know what symbol to use (e.g. the corresponding stroke, command name, or graphical button).

High guessability is even more important when using small devices for off-desktop computing. Small devices mean contrived input schemes, limited screen real estate for help screens, and "on the go" mobile use without access to unwieldy manuals. Also, the typical intermittent use of handheld devices means that users have less time for in-use learning. Modern users expect success right from the start.

Experts, not just novices, also need systems with high guessability. When an expert must perform an uncommon action, like entering an obscure character in a unistroke alphabet, his otherwise high performance may be significantly impeded unless the symbol is guessable.

This paper offers an approach to maximizing and evaluating the guessability of symbolic input. It defines guessability and offers a formal measure. This approach can be used in the design of a new symbol set, or in the evaluation or redesign of an existing symbol set. It is particularly relevant to designers of symbol sets that map to large numbers of referents—the more referents there are, the more important high guessability becomes.

We define *guessability* in symbolic input as:

> *That quality of symbols which allows a user to access intended referents via those symbols despite a lack of knowledge of those symbols.*

Guessability is contrasted to immediate usability [7] in that the latter involves the holistic evaluation of the initial user experience after a brief learning period; guessability evaluates only the input symbols without prior learning.

RELATED WORK

Guessability is crucial in command-line interfaces. Prior studies [4,5] show that designers often supply only one command-line term per referent. But one term, no matter how "natural," results in guessability failures of 80-90% [3]. A proposed solution is "unlimited aliasing" [4], where the system makes the best guess at the intended referent in the event of an unrecognized symbol. Having multiple

CHI 2005, April 2–7, 2004, Portland, Oregon, USA.
ACM 1-59593-002-7/05/0004.

synonyms has also been recognized as a key to achieving high guessability in command-line interfaces [4,5].

The guessability of text labels and graphical icons has also been studied [8]. Guessable labels and icons are important for the usability of buttons, toolbars, and menus. This paper's method for maximizing guessability can be applied to studies where participants devise text labels or sketch graphical icons for described features. Procedures for such studies have been delineated elsewhere (e.g. [1], p. 316).

The immediate usability of handheld text entry methods, most notably Graffiti [7], Graffiti 2 [6], and the Palm OS virtual keyboard [2,6], has also been studied. Immediate usability has been defined as initial usage *after minimal training*. An example is a user acquiring a new Palm PDA, studying the Graffiti character chart for a minute, and then trying to write. Results for both Graffiti and Graffiti 2 accuracy show respectably high immediate usability after minimal practice. Not surprisingly, virtual keyboards also have high immediate usability, since the symbol-to-referent mappings are obvious (i.e. labeled buttons enter corresponding ASCII characters).

Although immediate usability is important, it is a separate metric from guessability. Guessability is focused only on the quality of the input symbols without prior learning. Immediate usability assumes prior learning and evaluates the system as a whole, not just its input symbols.

MAXIMIZING GUESSABILITY

It is possible to design a highly guessable symbol set by acquiring guesses from participants. With the same participant data, we can also evaluate the guessability of an existing symbol set. The following sections describe our procedure. Then a concrete example is given for a real unistroke symbol set.

Achieving High Guessability with Participants

Participants are first recruited to propose symbols for specified referents within a given domain. The more participants, the more likely the resulting symbol set will be guessable to external users. The goal is to obtain a rich set of symbols from which to create the resultant symbol set.

Participants should be informed only of the details *essential* to proposing intelligent symbols. For example, if unistroke symbols are required, participants must be told what unistrokes are so that they refrain from making multi-stroke symbols. Participants should not be shown any example symbols or symbols from preexisting symbol sets. Of course, they must know the referents to which their symbols refer. Example referents are the ASCII letters to which unistrokes refer, the functions to which commands refer, or the features to which icons or text labels refer.

Capturing Symbols

Participants propose a symbol for each referent in turn. Symbols are captured and coupled with their intended referents. It is important not to bias the forms of the symbols by displaying the referents. For example, if participants are proposing unistroke gestures for ASCII letters, they should not see typeset letters as prompts. Similarly, if command names are being proposed, prompts containing ideal keywords should be avoided.

It is essential for conflict resolution (below) that captured symbols be testable for equality. Testing equality may be trivial, as in the case of keyword symbols, or more complex, as in the case of (x, y) point traces for unistrokes. For more complex symbols, designers may already have software to interpret them. Human judgment can also determine equality among, for example, sketches of icons.

Resolving Conflicts

One might imagine that we could simply lump together all participants' proposed symbols as our resultant symbol set and trivially achieve 100% guessability for the participants used. In practice, however, this is not usually possible due to conflicts—i.e. the same symbol will have been used to indicate different referents. An example from the literature [5] is the email command "To Dennis" being proposed to mean "send a message to Dennis" and also "list messages sent to Dennis." Similarly, the same unistroke gesture may be proposed for "h" and "n" [7]. But only one referent can be indicated by a given symbol. How do we decide which referent gets the symbol?

Symbols are tested for equality and grouped so that identical symbols form a "conflict group." After grouping, the different referents within each group are identified and the number of referring symbols counted. Then a scoring function determines which referent within each group is assigned that group's symbol. To maximize guessability, the referent that "wins" the symbol is the one with the most proposed symbols. Equation 1 expresses this as a function.

$$score = \left|symbols\right| \qquad (1)$$

For example, in 20 participants, if the same unistroke were proposed for "n", "h", and "a" with counts of 14, 5, and 1, respectively, the gesture would be assigned to referent "n".

In general, the more conflicted the set of proposed symbols, the lower the maximized guessability of the resultant symbol set. Intuitively, high conflict means participants are using identical symbols for different referents. Designers may improve this situation by making referents more distinct, by relaxing constraints on symbolic forms, or by asking participants to resolve all conflicts within their own sets of proposed symbols before they are finished.

Domain-specific considerations may be accommodated by using alternate scoring functions, although guessability may not be maximized. For example, in alphabetic entry we may wish to favor common letters over uncommon ones. Equation 2 is an example of an alternate scoring function that balances both letter frequency (0..1) and the number of proposed symbols.

$$score = frequency^{\frac{1}{\left|symbols\right|}} \qquad (2)$$

Calculating Guessability

Guessability has not been formalized in the literature. We therefore introduce a measure of guessability for symbolic input. The guessability G of the resultant symbol set S for the captured set of proposed symbols P is:

$$G = \frac{\sum_{s \in S} |P_s|}{|P|} \cdot 100\% \qquad (3)$$

In equation 3, P is the set of proposed symbols for all referents, and P_s is the set of proposed symbols using symbol s, which is a member of the resultant symbol set S. For our example of “n”, “h”, and “a” above, S = {“n”} and G = 14/20 · 100% = 70%. This means our resultant symbol set S was able to accommodate 70% of the symbols proposed by the participants.

Agreement

We may wish to know the agreement among symbols proposed by the participants. We therefore introduce a formalization of agreement A among symbols from our captured set P. Intuitively, agreement should be 100% when proposed symbols are identical, and ≈0% when they are unique. For example, in 20 proposals for referent r, if 15/20 are of one form and 5/20 are of another, there should be higher agreement than if 15/20 are of one form, 3/20 are of another, and 2/20 are of a third. Equation 4 captures this:

$$A = \frac{\sum_{r \in R} \sum_{P_i \subseteq P_r} \left(\frac{|P_i|}{|P_r|} \right)^2}{|R|} \cdot 100\% \qquad (4)$$

In equation 4, r is a referent in the set of all referents R, P_r is the set of proposals for referent r, and P_i is a subset of identical symbols from P_r. The range of equation 4 is $1/|P_r| \cdot 100\% \leq A \leq 100\%$. The lower bound is non-zero because even when all proposals disagree, each one trivially agrees with itself. For r, $[(15/20)^2 + (5/20)^2] / 1 \cdot 100\% = 62.5\%$ and $[(15/20)^2 + (3/20)^2 + (2/20)^2] / 1 \cdot 100\% = 59.5\%$.

Evaluating the Guessability of an Existing Symbol Set

One may also use the same participant data to evaluate the guessability of an *existing* symbol set S. Where a proposed symbol $p \in P$ used an existing symbol $s \in S$ that was correctly intended for s’s referent r, the proposal p is assigned to s. These proposed symbols (i.e. guesses) accumulate to form P_s in equation 3, the set of proposals using symbol s. Then equation 3 is applied, giving the percentage coverage of the captured symbols P by the symbols in the existing symbol set S. If all proposed symbols P are covered by S, the guessability G is 100%.

THE GUESSABILITY OF EDGEWRITE

As an example, we applied our approach to increase the guessability of the unistroke alphabet EdgeWrite [9]. Intuitively, one would not expect EdgeWrite to be highly guessable since its letters are made along the edges and into the corners of an area bounded by a physical square.

Method

Twenty participants, mostly staff and students from CMU, served as paid volunteers. None had prior experience with EdgeWrite or Graffiti. Participants were told they would be making unistroke gestures on a touchpad to indicate letters for a new alphabet. The unistroke concept was explained to prevent multi-stroke symbols. The importance of the four corners of the square input area was also explained, since EdgeWrite letters are defined not by their overall paths of motion but by their sequences of corner-hits [9]. No other constraints were in place and no examples were shown.

Participants were verbally prompted to enter each letter of the alphabet (*a..z*) and each number (0-9) by a Visual C# program that also recorded their gestures. An audio prompt was used to avoid biasing participants by the appearance of typeset letters. Participants were free to redo their symbols as often as they liked, but once a symbol was committed for a character, it could not be changed. To increase the variety of proposed symbols, participants were required to resolve conflicts among their own symbols before they were finished. Thus, each participant contributed 36 unique symbols, for $|P| = 20 \cdot 36 = 720$ proposed symbols in all.

Results

Corner sequences fully define an EdgeWrite gesture [9], so the 720 symbols were grouped by identical corner sequences in preparation for conflict resolution. The agreement of P was A = 34.9%, meaning about a third of the proposed symbols for a given referent agreed on average. After conflict resolution using the maximization scoring function (equation 1), the ensuing <u>u</u>ser-designed symbol set S_u accommodated 577 of 720 proposed symbols, for G_u = 80.1%. The <u>o</u>riginal EdgeWrite symbol set S_o was then evaluated for the proposed symbols P. It accommodated only 367 of 720, for G_o = 51.0%. Assuming the participants were representative, the improvement from S_o to S_u should generalize to larger populations of users.

THE IMMEDIATE USABILITY OF EDGEWRITE

In order to validate the improvement from S_o to S_u, we replicated a prior study of the immediate usability of Graffiti [7] for the two EdgeWrite alphabets. Recall that S_u was designed *only* by the proposed symbols of participants without designer intervention, and S_o was created by EdgeWrite’s designers over many prior studies [9]. Indeed, EdgeWrite’s designers were skeptical that an amalgam of uninformed participant symbols could actually be *more* usable than the product of many hours’ design work. They were further dubious that either alphabet would approach the immediate usability of Graffiti, since Graffiti had been shown to be “very respectable” in this manner [7].

Method

Our testing of immediate usability followed the prior study of Graffiti by MacKenzie and Zhang [7]. Twenty *new* participants served as paid volunteers. Like before, none of them had prior experience with Graffiti or EdgeWrite. The unistroke concept and importance of corner sequences were described to them. The same computer apparatus and touchpad were used as before.

As in the study of Graffiti [7], participants entered the alphabet (*a*..*z*) five times. This occurred twice in two separate phases of testing: the first after 1 minute of studying a 26-letter EdgeWrite character chart, and the second after 5 minutes of freeform practice with the same chart. Entered letters appeared in a Notepad document in Times 36pt font. Participants were not allowed to correct erroneous entries. Ten of the 20 participants used the original EdgeWrite alphabet S_o, and 10 used the user-designed alphabet S_u. Thus, for each alphabet, there were $26 \cdot 5 \cdot 2 \cdot 10 = 2600$ letters entered.

Results

As in the prior study, we measured the "accuracy attainable after minimal exposure" [7]. Figure 1 shows our results and those for Graffiti. After 1 minute of chart study, participants were 78.8% (12.6 stdev) accurate with the original alphabet S_o. This improved to 81.6% (12.8) for the user-designed alphabet S_u. This was very near the prior average for Graffiti of 81.8% (12.1). A one-way ANOVA shows no statistical differences for the three percentages ($F_{2,42}$=.23, *p*=.80), and no paired contrasts are significant.

After 5 minutes of freeform practice, S_o was 90.2% (11.0) accurate. S_u improved this to 94.2% (7.2). The latter was competitive with the prior result for Graffiti of 95.8% (4.0). A one-way ANOVA is nearly significant for the three percentages ($F_{2,42}$=2.43, *p*=.10). A paired contrast shows Graffiti was significantly more accurate than S_o ($F_{1,42}$=4.85, *p*<.05), but *not* significantly more accurate than S_u ($F_{1,42}$=.40, p=.53). S_u was not significantly more accurate than S_o, but the trend is in this direction ($F_{1,42}$=1.73, *p*=.19).

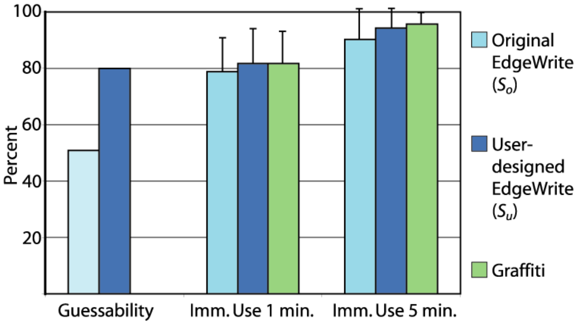

Figure 1. Guessability and immediate usability results. Error bars represent standard deviations. Graffiti data are from [7].

DISCUSSION

It was surprising that strict adherence to the guessability maximization procedure resulted in an alphabet (S_u) with higher average immediate usability than a highly iterated designer-made alphabet (S_o). Although this improvement was not quite significant after 5 minutes, that the average immediate usability increased at all shows the power of using participants to improve even refined symbol sets [5]. Furthermore, after examining the immediate usability data, we believe S_u could be improved even more by changing a few problematic symbols. For example, the "q" from S_u was only 57% accurate, while the "q" from S_o was 75% accurate. Other letters that were better in S_o than S_u were "t" (61% vs. 80%) and "y" (78% vs. 90%). It was also interesting that the standard deviations were similar for the three alphabets after 1 minute (≈12), but shrank considerably after 5 minutes for S_u (7.2) and Graffiti (4.0) but not for S_o (11.0), indicating more consistent performance for the more refined symbol sets. Finally, it was pleasing that EdgeWrite could be made competitive with Graffiti's laudable immediate usability [7].

CONCLUSION

High guessability is essential in today's computing systems, particularly for symbolic input. We have defined guessability and offered a procedure for its evaluation and maximization for symbolic input. The procedure described in this paper can be applied to a variety of domains, including gestures, voice commands, command keywords, text labels, and the design of icons. Through the application of this procedure, guessability can be quantified and compared, and the learnability of systems can be improved.

Acknowledgements

The authors thank Lisa Anthony, Darren Gergle, John Kembel, and Elaine Wherry. This work was supported by General Motors, Microsoft, Synaptics, and the National Science Foundation under grant UA-0308065. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NSF.

REFERENCES

[1] Brinck, T., Gergle, D. and Wood, S.D. (2001) *Usability for the Web*. San Francisco: Morgan Kaufmann.

[2] Fleetwood, M.D., Byrne, M.D., Centgraf, P., Dudziak, K.Q., Lin, B. and Mogilev, D. (2002) An evaluation of text-entry in Palm OS—Graffiti and the virtual keyboard. In *Proc. HFES 2002*. Human Factors and Ergonomics Society, pp. 617-621.

[3] Furnas, G.W., Landauer, T.K., Gomez, L.M. and Dumais, S.T. (1984) Statistical semantics: Analysis of the potential performance of keyword information systems. In *Human Factors in Computer Systems*, J.C. Thomas and M.L. Schneider (eds). Norwood, New Jersey: Ablex, pp. 187-242.

[4] Furnas, G.W., Landauer, T.K., Gomez, L.M. and Dumais, S.T. (1987) The vocabulary problem in human-system communication. *Communications of the ACM 30* (11), pp. 964-971.

[5] Good, M.D., Whiteside, J.A., Wixon, D.R. and Jones, S.J. (1984) Building a user-derived interface. *Communications of the ACM 27* (10), pp. 1032-1043.

[6] Költringer, T. and Grechenig, T. (2004) Comparing the immediate usability of Graffiti 2 and virtual keyboard. In *Proc. CHI 2004*. ACM Press, pp. 1175-1178.

[7] MacKenzie, I.S. and Zhang, S.X. (1997) The immediate usability of Graffiti. In *Proc. Graphics Interface 1997*. Canadian Information Processing Society, pp. 129-137.

[8] Wiedenbeck, S. (1999) The use of icons and labels in an end user application program: An empirical study of learning and retention. *Behavior and Information Technology 18* (2), pp. 68-82.

[9] Wobbrock, J.O., Myers, B.A. and Kembel, J.A. (2003) EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proc. UIST 2003*. ACM Press, pp. 61-70.