EFFECT OF LEVEL OF PROBLEM DESCRIPTION ON PROBLEM DISCOVERY RATES: TWO CASE STUDIES

James R. Lewis
IBM Conversational Speech Solutions
Boca Raton, FL

The primary purpose of this analysis was to investigate the effect of changing the level of description of usability problems on the estimate of the problem discovery rate (*p*). A secondary purpose was to describe a method for using *p* to estimate the number of problems remaining available for discovery given the constraints associated with a particular participant population, application, and set of tasks. The level of problem description influenced estimates of *p*, and the direction of influence was predictable, with higher levels of description producing higher estimates of *p*. Practitioners need a level of description that flows easily into recommendations for redesigning products, but to keep usability studies as efficient as possible, practitioners also need to seek a level of description that takes advantage of common patterns in observed usability problems. Managing this tradeoff is only one of the challenges of usability evaluation, but it is an important one.

INTRODUCTION

Motivation

A current area of usability research is to develop an understanding of the fundamental properties of usability problems. For example, one important outstanding issue is the development of a definition of what constitutes a 'real' usability problem with which a broad base of usability scientists and practitioners can agree (Cockton and Lavery, 1999; Connell and Hammond, 1999; Lavery, Cockton, and Atkinson, 1997; Lee, 1998).

Included in this issue is the appropriate level of description of usability problems. If the purpose of a usability study is to discover and fix problems during system development, then it might be necessary to describe the problems at a fairly low level to ensure a specific enough description to guide efforts to redesign the system. Alternatively, if the purpose is to map problems onto a theoretical or heuristic framework (such as that of Nielsen, 1994, or Limin, Salvendy, and Turley, 2002), it might be necessary to describe the problems at a higher level.

Another area of research in the properties of usability problems is the estimation of the problem discovery rate *p* (Lewis, 1992, 1994, 2001, 2006; Nielsen and Landauer, 1993; Virzi, 1990, 1992). This estimate is useful in planning sample size requirements for formative (diagnostic) usability studies and, after completing a study, in assessing the adequacy of the sample size.

The primary purpose of the current study was to investigate the effect of changing the level of description of usability problems on the estimate of the problem discovery rate. A secondary purpose was to describe a method for using the problem discovery rate to estimate the number of problems (rather than just the percentage) remaining available for discovery given the constraints associated with a particular participant population, application, and set of tasks.

Sample size estimation for problem-discovery studies

Estimating sample sizes for studies that have the primary purpose of discovering the problems in an interface depends on having an estimate of *p*, the average likelihood of problem occurrence (which is also an estimate of the problem discovery rate). This estimate can come from previous studies using the same method and similar system under evaluation, or can come from a pilot study. For standard scenario-based formative usability studies, the literature contains large-sample examples with *p* ranging from .16 to .42 (Lewis, 1994, 2001). For heuristic evaluations, the reported value of *p* from large-sample studies ranges from .22 to .60 (Nielsen and Molich, 1990).

When estimating *p* from a small sample (say, fewer than 20 participants), it is important to adjust its estimated value because small-sample estimates of *p* have a bias that results in substantial overestimation (Hertzum and Jacobsen, 2003).

A series of Monte Carlo experiments (Lewis, 2001) demonstrated that a formula combining Good-Turing discounting with a normalization procedure provides a very accurate adjustment of initial estimates of *p*, even when the sample size for that initial estimate has as few as two participants. This formula for the adjustment of *p* is:

$$adjp = ½[(estp – 1/n)(1 – 1/n)] + ½[estp/(1+GTadj)] \qquad [1]$$

where *GTadj* is the Good-Turing adjustment to probability space (which is the proportion of the number of problems that occurred once divided by the total number of different problems). The *estp*/(1+*GTadj*) component in the equation produces the Good-Turing adjusted estimate of *p* by dividing the observed, unadjusted estimate of *p* (*estp*) by the Good-Turing adjustment to probability space. The (*estp* – 1/*n*)(1 – 1/*n*) component in the equation produces the normalized estimate of *p* from the observed, unadjusted estimate of *p* and *n* (the sample size used to estimate *p*). The reason for averaging these two different estimates is that the Good-Turing estimator consistently tends to overestimate the true value of *p*, and the normalization consistently tends to underestimate it (Lewis, 2001).

Once you have an adjusted estimate for *p*, you can use the formula $1-(1-p)^n$ (derived from the binomial probability formula, Lewis, 1994, or, alternatively, from the Poisson probability formula, Nielsen and Landauer, 1993) with various values of *n* (say, from 1 to 20) to generate the curve of diminishing returns expected as a function of sample size (illustrated in Figure 1 for a range of values of *p*).

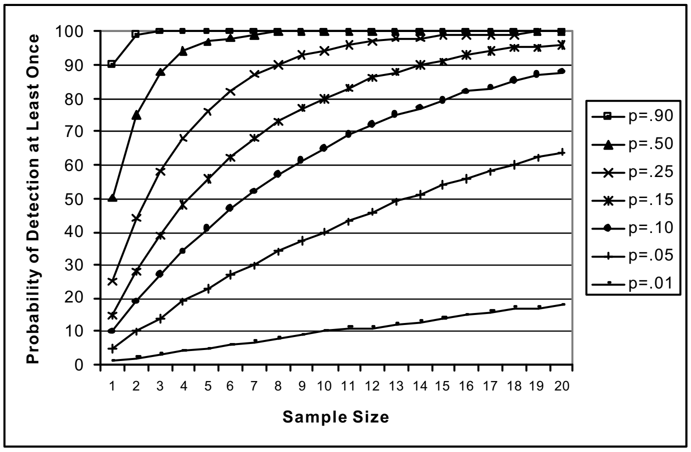

Figure 1. Problem Discovery Curves

Usability practitioners can use these curves to estimate sample size requirements for a usability study, typically by selecting some goal for percentage of problem discovery (say, 90%) and checking to see at what sample size for a given problem discovery rate the curve crosses 90%. Practitioners who must use a given sample size can use a variation of this technique to assess the effectiveness of the sample size by determining the percentage of problems discovered with the given sample size for the adjusted estimate of *p*.

It is also possible to solve the equation $Goal = 1 – (1 – p)^n$ for *n* (Lewis, 2006), which produces:

$$n = \log(1 - Goal)/\log(1 - p) \qquad [2]$$

For example, suppose a practitioner has decided that an appropriate problem-discovery goal is to find 97% of the discoverable problems, and has estimated *p* to be .28. The computed value of *n* is 10.6 (log(.03)/log(.72), or -1.522/-.143). The practitioner can either round the sample size up to

11 or adjust the problem-discovery goal down to 96.3% (1-(1-.28)^{10}).

Estimating the Number of Remaining Usability Problems

Because a practitioner knows the number of problems discovered for a given sample size in any specific usability study (given that practitioner's criteria for the identification and classification of usability problems), it is possible to use the estimates for the percentage of discovered problems to calculate the number of remaining problems and their likely pattern of discovery.

For example, suppose a practitioner has collected data from three participants, observed 12 distinct usability problems, and determined that the adjusted value of p was .25. From Figure 1, the percentage of problems discovered with three participants when p = .25 should be about 58%. If 12 problems are 58% of the total available for discovery, then the total is 12/.58, or 21.

CASE STUDY 1

The prototype in the first case study simulated a speech recognition application that had Weather, News, and E-mail/Calendar applications. Usability testing with three participants revealed six usability problems using a low-level of description, summarized in Table 1.

Table 1. Summary of Observed Usability Problems in Study 1 (Low-Level Description)

Problem Description	Part 1	Part 2	Part 3	Freq
1. Task 2: False stop	0	1	1	67
2. Task 4: Nomatch – back on task with Help 1	0	1	0	33
3. Task 5: Misleading prosody on prompt	0	1	0	33
4. Task 1: Nomatch – back on task with Help 1	0	0	1	33
5. Task 4: Exploring – searching for good option	0	0	1	33
6. Task 5: Nomatch – back on task with Help 1	0	0	1	33

Note: *Freq* = Percentage of participants who experienced the problem.

The six observed problems had low frequency except for accidental stopping of system playback of a news story in Task 2. The three nomatch events (failures to match what the user said with anything in the active grammar) occurred in three different tasks. In each case, the first level of help that played in response to the nomatch event put the participant back on task.

Using the formula given in [1], the adjusted value of p (*adjp*) is .125 (less than half the initial estimate). Projecting to a sample size of 3 with the cumulative binomial probability formula (1 - (1 - p)^n) with p = .125 and n = 3, the probable proportion of discovered problems is about .33. Therefore, the total number of problems available for discovery in this problem space (at the given level of problem definition) is about 18 (6/.33).

One way to shift to a higher level of description for the usability problems is to collapse problem types over task scenarios. At this level of description, the usability study uncovered four types of problems, summarized in Table 2.

Table 2. Summary of Observed Usability Problems in Study 1 (High-Level Description)

Problem Description	Part 1	Part 2	Part 3	Freq
1. False stop	0	1	1	67
2. Nomatch - Help 1	0	1	1	67
3. Bad prosody on prompt	0	1	0	33
4. Exploring	0	0	1	33

Note: *Freq* = Percentage of participants who experienced the problem.

For this level of description, the initial estimate of p (*estp*) was .50, and the adjusted estimate (*adjp*) was .22. Given these values, a sample size of 3 should uncover about 53% of the problems available for discovery (which, with four problems at this level of description observed, suggests that three or four problems remained undiscovered).

CASE STUDY 2

The prototype speech recognition application in the second case study also had

Weather, News, and Lotus Notes applications, but had design changes intended to eliminate or reduce the usability problems observed for the first prototype. It also had a natural command grammar for the e-mail/calendar application and other new functions, which were tested with new tasks.

Usability testing with seven participants revealed 33 usability problems (using a low-level of description). Analysis of the low-level problems provided an initial estimate of *p* (*estp*) of .27, and the adjusted estimate (*adjp*) was .15. Given these values, a sample size of 7 should uncover about 68% of the problems available for discovery (which, given 33 observed problems at this level of description, suggests that there were about 49 problems available for discovery in this problem space, with about 16 undiscovered problems.

At the higher level of description, the usability study uncovered 18 types of problems. The initial estimate of *p* (*estp*) was .36, and the adjusted estimate (*adjp*) was .235. Given these values, a sample size of seven should have uncovered about 85% of the problems available for discovery (which, with 18 observed problems at this level of description, suggests that there were about 21 problems available for discovery, with 3 remaining undiscovered).

DISCUSSION

The goal of formative (diagnostic) usability testing in industrial practice is the discovery, prioritization, and resolution of usability problems (Lewis, 2006). It is possible, however, that research conducted in the emerging field of usability science (Gillan and Bias, 2001) could have different goals that necessitate different levels of problem description.

These case studies indicate that the level of problem description influences estimates of problem discovery rates and that the direction of influence was predictable, with higher levels of description producing higher estimates of *p*.

Consider the highest possible level of description, which is simply that a problem exists. This has the effect of collapsing all problems into a single row. For this general situation, it is very unlikely that the value of *GTadj* (the Good-Turing discounted adjustment for *p*) would be anything other than 0, removing any effect of Good-Turing discounting from the adjustment of *p*.

The opposite extreme is to consider every observed problem to be completely unique. For Case Study 1, there were a total of 7 observed problems at the low level of problem description. In Case Study 2, the total at that level was 63.

If every problem is unique, then the value of *GTadj* is necessarily 1. The value of *estp* is the number of observed problems divided by the number of observed problems times the number of participants observed, which reduces to $1/n$. For Case Study 1, the value of estp is .33, and for Case Study 2 was .14. Because the numerator of the normalization portion of Equation [1] is $estp - 1/n$, and in this situation, $estp = 1/n$, this part of the computation is 0. Because the Good-Turing and normalization components are averaged in the equation, the effect is to make *adjp* equal to *estp*/4.

Thus, for the problems observed in Case Study 1, the adjusted estimate of *p* could range from .08 to .45, depending on the level of problem description. For Case Study 2, the value could range from .04 to .87.

Practitioners need a level of description that flows easily into recommendations for redesigning products. Any other level of description places severe limitations on the practical utility of a usability study. On the other hand, to keep usability studies as efficient as possible by maximizing adjusted values for *p*, practitioners need to seek a level of description that takes advantage of common patterns in observed usability problems. Managing this tradeoff is only one of the challenges of usability evaluation, but it is an important one.

REFERENCES

Cockton, G., and Lavery, D. (1999). A framework for usability problem extraction. In *Human-Computer Interaction -- INTERACT '99* (pp. 344-352). Amsterdam: IOS Press.

Connell, I. W., and Hammond, N. V. (1999). Comparing usability evaluation principles with heuristics: Problem instances vs. problem types. *Human-Computer Interaction -- INTERACT '99* (pp. 621-629). Amsterdam: IOS Press.

Gillan, D. J., and Bias, R. G. Usability science I: Foundations. *International Journal of Human-Computer Interaction*, *13*, 351-372.

Hertzum, M., and Jacobsen, N. J. (2003). The evaluator effect: A chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction*, *15*, 183-204.

Lavery, D., Cockton, G., and Atkinson, M. P. (1997). Comparison of evaluation methods using structured usability problem reports. *Behaviour & Information Technology*, *16*, 246-266.

Lee, W. O. (1998). Analysis of problems found in user testing using an approximate model of user action. In *People and Computers XIII: Proceedings of HCI '98* (pp. 23-35). Sheffield, UK: Springer-Verlag.

Lewis, J. R. (1994). Sample sizes for usability studies: Additional considerations. *Human Factors*, *36*, 368-378.

Lewis, J. R. (2001). Evaluation of procedures for adjusting problem-discovery rates estimated from small samples. *International Journal of Human-Computer Interaction*, *13*, 445-480.

Lewis, J. R.. (2006). Usability testing. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (pp. 1275-1316). New York, NY: John Wiley.

Limin, F., Salvendy, G., and Turley, L. (2002). Effectiveness of user testing and heuristic evaluation as a function of performance classification. *Behaviour & Information Technology*, *21*, 137-143.

Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods* (pp. 25-61). New York, NY: John Wiley.

Nielsen, J., and Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Conference Proceedings on Human Factors in Computing Systems – CHI93* (pp. 206-213). New York, NY: Association for Computing Machinery.

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces. In *Conference Proceedings on Human Factors in Computing Systems – CHI90* (pp. 249-256). New York, NY: ACM.

Virzi, R. A. (1990). Streamlining the design process: Running fewer subjects. In *Proceedings of the Human Factors Society 34th Annual Meeting* (pp. 291-294). Santa Monica, CA: Human Factors Society.

Virzi, R. A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors*, *34*, 443-451.