A method for evaluating calculator interfaces

S. Wali P. Cairns H. Thimbleby

UCLIC, UCL Interaction Centre
University College London
26 Bedford Way
London
WC1H 0AP

24th October, 2002

Abstract

Calculators have traditional user interfaces that no longer fit with acceptable practice in user interface design. This paper proposes a method for evaluating calculator user interfaces and demonstrates its value on a radically different style of calculator. The method produces a strong correlation between prior competence of the user and the time taken to achieve tasks with a given calculator despite differences in interfaces. Using this method, new calculator designs could be effectively evaluated and hence lead to long term improvements in their design.

1 The problem with calculators

In this paper, we consider basic four function calculators, that is, ones with the four arithmetic functions together with one or two extra facilities such as a memory, percent and square root. The user interfaces for such calculators has remained reasonably static for the last thirty years. This is exemplified by the two Casio calculators 101-F and HS-8V showing in Figure 1. The former was released in July 1975 [3] and the latter is still on sale at the time of writing (late 2002). Both have the four arithmetic functions, square root, memory facility working with the MC, MR, M+ and M− keys and a % key.

With such a long history and the huge advances in both technology and in understanding usability, it would be hoped and expected that the calculator interface was well matched to usability requirements. This is not the case and we consider some examples from the HS-8V.

A simple usability criterion [5] is that the user receives feedback on the state of the system. On picking up the HS-8V calculator when it shows

Figure 1: The Casio 101F (left) and the Casio HS-8V.

it is impossible to say whether the next digit press will add another digit to the right of the '1.' (e.g., to display 1.2 say), whether the next digit will be inserted between the 1 and the decimal point (e.g., to display 12.) or whether the new digit will entirely replace the '1' on the display (e.g., to display 2.). A related problem is that it is impossible to say whether or not a function key (plus, minus etc) has been pressed and if so which one. Thus, a slight slip of the finger can result in a [−] instead of a [+] press, which over a long addition may go unnoticed. Also in a long sum, users may lose their place and not know what they last entered. There is no history of entries made available by the calculator. Furthermore, a double press of a function key — deliberately or as a slip — will allow you to do a "constant" function, for example, to subsequently multiply by the same multiplicand. There is again no indication that the calculator is in this state. Interestingly the older 101F does indicate this (to an observant and knowing user) with a small dot coming on when the constant operation has been activated.

There are also deeper functional problems. If in the course of doing a calculation you have one number in memory and another on the display, you may wish to swap the numbers. This is particularly necessary as the calculator does not obey the usual mathematical priorities of multiplication and division over addition and subtraction. There is no obvious way to do this. It requires at least six key presses, for example the key sequence [M+], [−], [MRC], [=], [M+], [±] will do.[1] Obviously. In fact, there are an infinite number of ways to swap the display and memory[2] — and not one method is remotely obvious. It's easier to use paper than use the memory.

Percentage keys also pose problems. Once mastered for a particular calculator, there is no guarantee that the percentage key will work the same on other calculators even those of the same manufacturer. Details of this and other problems are given elsewhere [8, 10].

These are issues inherent to the specific device but not to calculators *per se*. There are also concerns with how the users trust calculators. Increasingly, calculators are used instead of mental arithmetic. MacKenzie gives the example of doing the simple sum, $(400/1200) \times 588$, which when given to a class, they all immediately reached for their calculators [4], rather than simplifying the simple 4/12 part of it first. We have had similar experiences. This sum on the sorts of calculator discussed will result in a rounding error and give the answer 195.99998, for instance. Collins and Kusch regard this as acceptable [2] — the user engages in a repair and knows the correct answer to be 196. But MacKenzie points out that, in practice, this does not happen. Instead, calculators are believed to be authoritative and accurate. If so, then manufacturers should be making every effort to ensure that they are.

[1] ... provided the memory does not overflow as a result of the additions caused by [M+].

[2] For example, once you know how to do a swap, doing a swap three times, perhaps in different ways, is yet another way of doing a swap. There are more interesting ways of finding new ways of swapping than this, too, but exploring them here will take us beyond the concerns of this paper [11]. However, on similar calculators without a [±] key (and without a swap key), the swap problem is insoluble.

2 The purpose of the paper

A readily-available Casio calculator (the HS-8V) was used to demonstrate problems that seem common to all calculator manufacturers. The poor quality of interfaces across many current models, from many manufcaturers, indicate the need for a better design more suited to the needs of users. This has a particular moral obligation because calculators are considered an essential part of a child's education. They are in the National Curriculum in the UK. This means we are obliged to teach our children mathematics using tools that are inconsistent and confusing. This is not going to help a child with what is already perceived to be a difficult subject.

Of course, unless we can propose better user interfaces and show that they are better, then we are merely being churlish. This paper considers a very different, new design of calculator already proposed [7, 8] in 1986 and shows that a single method of evaluation can give a high correlation between user competence and user performance. This is used to compare the calculators. Though both calculators seem to perform to about the same standard, that is a source for hope: all evaluators had at least ten years' experience with existing designs yet performed just as well with a calculator with which they had less than an hour's experience. We conclude the paper with a discussion for future developments in calculator design and evaluation.

3 Measuring improvements in calculators

Clearly, calculators could be made easier to use by solving the problems already highlighted without a radical re-design. It seems probable, though, that we could do a lot better given the advances in user interface technology over the past twenty five to thirty years. However, before expelling a long tradition of calculator user interfaces, there should be worthwhile gains to be had from new interfaces. After all, if all that is achieved is a fractional speed up in calculation time, it is hard to justify entirely replacing all of the existing calculators in use.

The question, then, is what makes a better calculator interface. There are many possible measures but we chose the following as a reasonable starting point [6]:

- Reduced number of user errors
- Increased speed of performing calculations
- Increased user satisfaction and confidence

If there are measurable improvements in these factors and the interface also provides better feedback, simpler memory facilities and more accurate answers then there are good grounds for judging the interface to be superior.

The biggest challenge is how to reliably measure the user errors and speed. There are many potentially confounding factors. If the tasks involved solving mathematics problems that require a good mathematical ability, some users would be unable to do them. If the tasks were simply entering calculations into a calculator, this is not a realistic use of calculators and may also favour one sort of interface over another.

The tasks therefore were based on UK GCSE exam questions from the calculator section of the exam. Thus, the tasks would be problems deemed by examination boards to be appropriate to challenge calculator users without overtaxing other mathematical abilities. Furthermore, our evaluators were required to have passed GCSE mathematics or equivalent so that they would at some point have been able to satisfactorily do such questions.

This still does not counter the natural variation in individuals. We would not expect to see all users performing calculations within a similar timescale. To obtain a measure of user ability, we asked users to read each task before attempting to use the calculator and to only start the calculation once they had understood the task. The time to understand the question was plotted against the time to complete the calculation. This produced a scatter graph which, if there was strong correlation, would show how user performance related to their ability to understand mathematical tasks. We naturally expect a positive correlation, that is, users that were slower at understanding the tasks would take longer to complete them. By comparing the slopes of correlated data for different interfaces, it would be possible to make some sort of comparison between the interfaces. A steeper slope would indicate that, in general, users did not speed up as much by using that interface as much as using an interface that produced a shallower slope.

If the methodology were tried on similar interfaces, it is likely to produce similar results. Accordingly, two interfaces of quite different design were used. The first was based on the HS-8V described above and taken as representative of a broad set of calculators. The other was a new design [10] that uses a declarative interface more like the declarative style of written mathematics. Although the declarative calculator is described in the original paper, we give some brief details here to illustrate how different it is from traditional calculators.

The declarative calculator is implemented on an Apple Macintosh [12]. The user presses digits and functions as for a usual keyboard entry and the corresponding symbols appear in a window. Each line of the window corresponds to an single calculation and is editable like a word processor. Thus, there is no need for memory facilities common to traditional calculators. Also, as the user types in a calculation, the calculator automatically "completes" the calculation to make it correct. For example, if the user types in 3×2 the calculator will complete the equation as it is entered. So initially, on typing the 3, the calculator makes a true equation by completing it with $= 3$. Typing $\times$ the calculator fills in a 1 so that $3 \times 1 = 3$ is the new true equation and finally on typing 2, the calculator will complete the equation to $\ldots = 6$. Table 1 shows the successive displays as the user types.

If the user leaves a gap in the calculation corresponding to an unknown in an algebraic equation, the calculator fills in the gap with a value that preserves equality. In the example above, if the user had entered $3\times = 12$, the calculator would have inserted 4 at the correct place. The advantage of this is more apparent with calculations like $\sin = 0.5$, $\log 4 + \log = \log 20$, and even $5040 =!$ — which asks for the factorial 7!, a calculation that is next to impossible on an ordinary calculator. Thus, unlike in a conventional caclulator, a user does not need to re-arrange equations to find solutions to problems, they simply need to be able to express the problem as an equation and enter that. More realistic examples are given in [8, 9], however no such complicated examples were used in our experiments.

Key press	Display after each key press
	$_{\Delta}0 = 0$
$\boxed{3}$	$3_{\Delta} = 3$
$\boxed{\times}$	$3 \times {}_{\Delta}1 = 3$
$\boxed{2}$	$3 \times 2_{\Delta} = 6$

Table 1: Example use of declarative calculator. The $_{\Delta}$ symbol indicates the cursor position on the display. The actual calculator uses two colours in the display to disambiguate the typing from the answers: this detail is not shown here.

To distinguish what the user has entered and what the calculator has completed, different coloured fonts are used, but at any time the user can choose to take over the calculator entries and the calculator will then make its completions elsewhere in the equation.

There is little overlap, then, between the *modus operandi* of the two calculators. This provides a good opportunity to see whether our proposed method is really evaluating the effects of interfaces on user performance.

4 Evaluations of Interfaces

Two experiments were performed to fine tune the method. In each case, the tasks were five GCSE mathematics examination questions from the calculator papers. They had been chosen to avoid direct matching with either calculator interaction style and tested in a pilot study to ensure this.

As the declarative calculator ran only through an Apple there could be problems in comparing it with the physical device of the Casio. Instead, a complete simulation of the HS-8V was implemented in Visual Basic and run on a PC running Windows NT. At the start of each test, the user was presented with one or other of the calculators already running on the computer so there were no real differences resulting from the differences in operating system.

During the experiment users were observed by the experimenter present in the room. Video observation was considered but found unnecessary. The aim of the experiment was to measure user performance in a mostly quantitative format. Video recording offered no advantages over direct observation in this case. Also, the experimenter would have to spend some time with the users familiarising them with the interfaces. Subsequently removing the experimenter is more likely to increase user feelings of formality and possibly adversely affect performances.

User errors were recorded by the observer. A user error was one of the following:

- Calculation restarted from the beginning
- Incorrect values used in a calculation
- Incorrect method such as inappropriate button presses

In addition to errors, the observer also noted any comments that the user made. Unexpected behaviour was also noted.

Timing of various phases of the experiment were made using a digital stopwatch. As the stopwatch was operated by the observer whilst the experiment was in progress, timings were only taken to the nearest second. Two major types of timing were taken: the time a user took to understand the task; the time a user took to perform the calculation having understood the task. From the pilot study, it was clear that waiting for the user to declare when they had completed the task was not always useful. Some users would feel they might have the wrong answer and go back and start again even if they did not. [Sameera: did you take the timings differently as a result in the experiments?]

The pilot study also helped to refine the tasks themselves. The final tasks used were chosen at a simpler level of mathematics. They were still GCSE mathematics questions, though, so they retained their relevance as realistic, appropriate tasks to perform with a calculator.

At the end of each experiment, the user was asked to fill out the Subjective User Satisfaction (SUS) questionnaire [1]. This is a quick and dirty instrument but would be useful to give some feeling for how satisfied users were with the interface. Also, it provides the opportunity for more open feedback which helps in more qualitative assessment of the interfaces.

4.1 Experiment 1

Experiment 1 was a between subjects test with 16 subjects in total. Eight used the HS-8V simulation running on a PC with Windows NT operating system. The other eight used the declarative calculator running on an Apple iMac. Both were given the same introductory briefing except that certain interface specific differences. These were written examples of how to do square roots and percentages appropriate to each calculator.

Each subject was given an example task in order to familiarise themselves with the calculator. In this time, they were allowed to ask the experimenter for help or clarifications on the interface.

Having completed the example, the subject was given five tasks, one at a time. When the subject was satisfied that they understood the task, they said "stop." They were allowed to use pen and paper to help them understand the task. The time from being given the task to the time they said stop was measured and recorded as the time to understand the task.

Once the subject felt they had understood the task, the experimenter told them to solve the task using the calculator. Again, the subject said "stop" to indicate when they had completed the task. The time from being told to start and the subject saying "stop" was measured and recorded as the time to complete the task.

On completion of all tasks, the subject was given the SUS questionnaire to fill out and the opportunity to make any other remarks, critical or otherwise, about the calculator that they had been using.

During the course of the experiments, the experimenter noticed a bug in the HS-8V simulation that affected what was displayed and hence gave incorrect values. Only one subject out of the eight noticed this bug.

In each condition, the total time to understand the tasks and to complete the tasks for each subject is given in Table 2.

Subject	Calculator	Total time to understand (s)	Total time to complete (s)
1	HS-8V	52	231
2	HS-8V	57	300
3	HS-8V	57	280
4	HS-8V	146	516
5	HS-8V	104	201
6	HS-8V	163	262
7	HS-8V	74	163
8	HS-8V	48	326
9	Declarative	286	469
10	Declarative	116	526
11	Declarative	115	375
12	Declarative	107	372
13	Declarative	84	350
14	Declarative	411	569
15	Declarative	134	510
16	Declarative	35	268

Table 2: Times taken to understand and to complete all tasks for all subjects.

Figure 2: Total times to complete all tasks plotted against total times to understand all tasks in Experiment 1.

The Pearson's correlation coefficient (r-value) was calculated for both conditions. For the HS-8V, $r = 0.355$ (3sf) and for the declarative calculator, $r = 0.725$ (3sf). For the sample size of eight and a one-tailed test at 95% significance, the r-value is 0.622. Thus, there is no significant positive correlation between the time to understand and complete the task for the users of the HS-8V. But for the declarative calculator, there is a significant correlation. The scatter graph together with the linear regression lines is show in Figure 2.

It is worth noting that between the two calculators, the mean times to understand and to complete each task were in fact generally larger for the declarative calculator than for the HS-8V. However, there was no significant difference in the understanding times and only two of the five tasks were the completion times significantly different. This indicates that the populations were broadly similar in terms of ability to understand problems but that the declarative calculator made certain tasks slower.

The results of the SUS questionnaire are that the HS-8V scored a mean score of 40.6 and the declarative calculator 48.1. Though this indicates greater satisfaction with the declarative calculator the difference is not significant and also even if it were, the SUS does not really merit such a comparison. Rather, it suggests that both calculators are reasonably satisfying to use but that there are probably some serious problems with each.

The counting of errors gave that subjects using the HS-8V produced a total of 14 errors whilst the declarative calculator users made only eight errors in total. As errors are so few, further analysis was performed.

Subject	Calculator	Total time to understand (s)	Total time to complete (s)
1	HS-8V	28	219
2	HS-8V	60	344
3	HS-8V	95	290
4	HS-8V	64	327
5	HS-8V	82	358
6	HS-8V	168	507
7	Declarative	33	393
8	Declarative	51	322
9	Declarative	64	366
10	Declarative	34	249
11	Declarative	90	449
12	Declarative	88	484

Table 3: Times taken to understand and to complete all tasks for all subjects.

The lack of significant correlation means that it is not possible to compare the calculators. A possible cause might the lack of recent practice of the subjects at both mathematical problems and calculator use. However, given the generally better times to use the HS-8V, the lack of practice at calculator use may not be the significant factor. The second experiment was devised in order give some training at mathematical problems through pen and paper exercises first.

4.2 Experiment 2

The experimental conditions were largely the same as Experiment 1. Twelve subjects were used, six on each calculator with the tasks performed on the calculator being the same as the tasks performed in Experiment 1. The same timing measures were used as well. Before these tasks, each subject was given five other tasks to perform with pen and paper. The subject was asked first to read and understand the test before attempting calculate a solution. The times to understand and complete the pen and paper tasks were also recorded as for the calculator tasks. The subject was then trained on how to use the calculator using tasks from the pen and paper exercise as training examples.

As previously, when the subject had completed all the tasks on the calculator they were given the SUS questionnaire to fill out and the opportunity to make comments.

The mean times to understand and complete each individual task, be it with pen and paper or with a calculator, was not significantly different for subjects using the HS-8V from those using the declarative calculator. Thus, this time, the abilities of the two sets of subjects seem more evenly matched. Also, there was not the trend that the subjects using the declarative calculator took longer on average to complete the tasks than those using the HS-8V.

In each condition, the total time to understand the tasks and to complete the tasks for each subject is given in Table 3.

The Pearson's correlation coefficient (r-value) was calculated for both conditions. For the HS-8V, $r = 0.899$ (3sf) and for the declarative calculator,

Figure 3: Total times to complete all tasks plotted against total times to understand all tasks in Experiment 2.

$r = 0.789$ (3sf). For the sample size of six and a one-tailed test at 95% significance, the r-value is 0.669. (In fact, the HS-8V correlation is significant at 99%.) Thus, there is a significant positive correlation between the time to understand and complete the task for the users of both calculators. This means that it is legitimate to compare the calculators in terms of how much the completion time is affected by the calculator being used. That is, the gradient of the slopes of the regression lines gives an indicator of the extent to which the calculator speeds up or slows down the users ability to do a calculation. The scatter graph together with the linear regression lines is show in Figure 3. As can be seen, both lines have a similar gradient indicating that there is no particular difference in the effective of the calculators on the efficiency of the users.

This time for the SUS questionnaire, the HS-8V scored a mean score of 47.1 and the declarative calculator 50.4. The difference is certainly not significant. Interestingly, the spread of the scores is quite large — the declarative calculator has a spread of only 2.9 in comparison to 10.1 for the HS-8V. The latter calculator seems to evoke a wider range of satisfaction (or dissatisfaction).

The number of errors again showed a similar trend as in Experiment 1. The HS-8V users produced a total of 9 errors and the declarative calculator users only 5. No further analysis was made as there were so few errors to work with.

4.3 Some user feedback

On the whole, the HS-8V elicited little feedback as it conformed to the users' expectations of what a calculator should be like. The only criticisms were the function of the [%], memory and [±] buttons which one or two users found counter-intuitive. One user commented that they would have preferred a keyboard rather than mouse interface.

The declarative calculator elicited more comments because it was so unfamiliar. The fact that it immediately starts to correct equations was commented on. Some thought it was confusing and distracting. One person liked it because of the immediacy of the feedback.

Because the calculator fills in answers as the calculation is entered, some users were initially disconcerted by not needing to use a press of [=] to get an answer. This, however, seems like a carry-over from existing calculators as that is traditionally inidicate to a calculator that you have finished a calculation. It is certainly not the last symbol put down when you do such calculations by hand!

The property of the calculator that it behaves like an editor was considered as favourable. It allowed users to edit calculations if they made mistakes as opposed to having to restart calculations as with a more usual calculator. Also, they did not need to explicitly use a memory function — everything they entered was automatically visible unless they actively deleted it.

The interface to the declarative calculator seemed to offer some problems. Some users said that they would prefer a mouse input rather than having to "hunt and peck" for functions as was observed by the experimenter. Also, the

square root formula was found to be cumbersome and maybe a single button would have been more appropriate. This fits with traditional calculators but may also match with the concept of square rooting as a special function rather than an instance of taking powers.[3]

5 Discussion and further work

The primary aim of these experiments was to see if it was possible to find some reliable measure of the usability of calculators. Experiment 2 indicates that after a training/practice period, there is a strong correlation between the time a user takes to understand a task and the time to complete it on a calculator. Inasmuch as time to understand indicates the fluency of the user at mathematics and that speed to achieve tasks is a measure of usability, this correlation provides an indicator of usability independent of individual differences in user ability. The slope of the linear regression can be compared to show which calculator is providing better "speed up" on users' abilities and thus it is possible to get some indication of calculator usability.

Of course, there are other measures of usability such as frequency of errors and user satisfaction but these can be measured along side or independently of these measures and so provide a cumulative view of the usability of calculators.

This trial is clearly only small scale so further work would necessarily repeat the experiments with a greater number of subjects and variations in interfaces tested. In particular, it might be interesting to see how moving from four function to trigonometric functions and beyond may affect usability. The declarative calculator can easily do such calculations and there is much greater variation between interfaces of calculators currently sold. This would require a broader set of tasks. The GCSE exams may be biassed in favour of conventional calculator user interfaces (e.g., requiring only trivial manipulation of the equation to get the answer last, which the declarative calculator does not require).[4] It would be a challenge to devise measures that still took account of individual variations in ability when the spread of ability is likely to be much larger.

A secondary effect of these experiments is that they gave some measure of the relative usabilities of a basic four function calculator and the declarative calculator. By the several measures taken and the new measure developed, the declarative calculator was not obviously more usable. Users seemed more satisfied and made fewer errors but longer, more comprehensive trials would be needed to see if this difference is real. The new measure devised here also showed that the speed up of users was less than with a traditional calculator.

Additionally, there are other considerations. Most users were familiar with the traditional four function calculator from several years of training and use in schools. A longer trial, involving possibly several testing sessions over a substantial period of time might help counteract the prior learning effect of the traditional calculators. Such a trial is currently being planned.

That traditional calculators clearly flaunt both usability and correctness criteria is lamentable having not substantially progressed their interfaces in over twenty five years of development. The declarative calculator has the more

[3]The declarative calculator has no square root button; instead, numbers are raised to the power 0.5, which is a more general approach but was unfamiliar to our subjects.

[4]WHAT WERE THE QUESTIONS? IS THIS PLAUSIBLE?

usable aspects founded on good usability principles implemented in modern technology.

So taken all in all, the declarative calculator is certainly not out of the picture. It fairs about as well as a traditional calculator and clearly improves upon basic usability features.

There is also a deeper issue of the matching between the declarative calculator and how mathematics is done by hand. Currently, children are taught specifically how to use calculators in addition to the maths that requires calculators. The complexity of this is clearly seen in that most schools specify which calculators to use so that pupils can all be taught the same way. Using a more declarative style of calculator, users could move from the pencil and paper style of mathematics to a calculator without a significant learning overhead. It would mean an end to learning how to use this calculator or that one and everyone would simply be doing mathemtics.

Acknowledgements

Harold Thimbleby is a Royal Society Wolfson Research Merit Award Holder, and acknowledges the support of the Royal Society.

References

[1] J. Brooke, "SUS: A quick and dirty usability scale," 1996.
`http://www.cee.hw.ac.uk/~ph/sus.html`

[2] H. Collins & M. Kusch, *The Shape of Actions*, MIT Press, 1998.

[3] R. MacKay, "Old Casio Calculators," October, 2002.
`http://www.dotpoint.com/xnumber/casio.htm`

[4] D. MacKenzie, *Mechanising Proof*, MIT Press, 2001.

[5] J. Nielsen, *Usability Engineering*, Morgan Kaufmann, 1993.

[6] J. Preece, Y. Rogers & H. Sharp, *Interaction Design*, Addison-Wesley, 2001.

[7] H. Thimbleby, "The Design of Two Innovative User Interfaces," *Proceedings British Computer Society Conference on Human Computer Interaction*, HCI'86, M. D. Harrison & A. F. Monk (editors), York, pp336–351, Cambridge University Press, 1986.

[8] H. Thimbleby, "A New Calculator and Why it is Necessary," *Computer Journal*, **38**(6):418–433, 1996.

[9] H. Thimbleby, "A True Calculator," *Engineering Science and Education Journal*, **6**(3):128–136, 1997.

[10] H. Thimbleby, "Calculators Are Needlessly Bad," *International Journal of Human-Computer Studies*, **52**(6):1031–1069, 2000.

[11] H. Thimbleby, "User Interface Design With Matrix Algebra," Working Paper, 2002.
`http://www.uclic.ucl.ac.uk/harold/srf`

[12] W. J. Thimbleby, 2002.
http://