EVALUATING A NOVEL CALCULATOR INTERFACE

Paul Cairns **Sameera Wali** **Harold Thimbleby**
UCL Interaction Centre
University College London
31-32 Alfred Place
London WC1E 7DP, UK
p.cairns@ucl.ac.uk

ABSTRACT
Calculators have very specific and recognisable interfaces to such an extent that the physical interfaces are reproduced even in computer-based calculators. However, these interfaces are not without fault. In previous work, Thimbleby proposed and implemented a calculator with an interface more in keeping with mathematical style. This paper describes a comparative evaluation of this interface. The results show promise for the new interface but there are still some usability problems that need to be ironed out.

Keywords
calculator; declarative calculator; interface evaluation

1. INTERFACES FOR CALCULATORS
Calculators are widely used in everyday life: they appear on shop counters, office desktops and even mobile phones. Their importance is recognised in that they are specifically included in the National Curriculum for UK school children. However, despite being a significant part of life, design of basic calculator interfaces seems to have frozen over the last thirty years. For example, the Casio 101F from 1975, shown in Figure 1, has essentially the same interface as the more modern Casio HS-8V still currently on sale. Indeed, somewhat surprisingly, it bears more than a passing resemblance to the calculator program that comes as part of the Windows XP OS. If these designs were without problems then this would be fine but this is not the case. In this paper then, we consider basic four function calculators and empirically evaluate a new design for a calculator interface against existing designs. A challenging part of the evaluation is accounting for differences in mathematical ability between different users. Our method seems to give clear results that do account for these differences. Before giving details of the evaluation, we consider existing calculator problems and briefly describe the new design.

Fig. 1: The Casio 101F released in 1975

2. CALCULATOR PROBLEMS
Using the HS-8V as a (representative) example, there are several immediate problems with the interface. Contrary to a simple usability criterion, the calculator has many states that affect interaction but are not at all visible to the user [2]. For example, consider a calculator simply showing the display:

1.

A user cannot tell whether pressing “2” would result in the display showing “12”, “1.2” or “2” and if the last then what is the operation that was or will be performed, if any? The simple operation of swapping over the contents of the memory and the display, which is really useful in slightly longer calculations, requires many key presses which can only be found after some very careful thinking [6]. These problems could just be quirks of that particular model but they are not. In fact, these ‘quirks’ are faithfully reproduced in the Windows calculator.

Such calculators also impose extensive cognitive loading (or pencil and paper) to do all but the simplest of calculations. For instance, suppose you want to buy stamps at Christmas and you want to know how many 1st class stamps (27p each) you can get for your £5 note not forgetting that your aunt’s birthday present also needs postage of £2.30. This is not a hard calculation but the calculator in no way helps you with it. It is up to you to work out that the correct calculation, as required by the calculator, are the key presses: “5 – 2 . 3 0 / 0 . 2 7 =”. And actually, you would probably want to press “=” before “/” just in case you are using a calculator that does not take notice of the precedence of operators.

In this example, the calculator is not helping the user to calculate. It is simply acting like a programming language – give it the right instructions and it will give you the right answer – but you had better make sure you know how to program it. In the context of children’s learning, these design issues may have a real, detrimental impact in what is already considered to be a tricky subject. In a computer-based calculator it must be possible to do better than this.

3. A SOLUTION?

In an attempt to improve matters, Thimbleby developed the declarative calculator [4,5]. Briefly, the conceptual model for this calculator is one of a whiteboard where the user and the computer both write on the board. The computer’s contribution ensure that the board is always showing a correct equation. To make this clear, the computer writes on the board in a different coloured ink. For the above example, the user might write on the board, “5 = 2.30 + 0.27 ×”. As they write, the computer would also be writing so that when the final blank is left, the computer fills this in with the required answer. The reader is referred to the earlier papers for a fuller description but the intended advantages are that: no device specific “progammable” version of the problem is required; there is no distinction between memory and display; there are no modes; the input is more like the structure of the problem and more like traditional mathematical style.

Though extensive analysis and implementation effort has gone into this design, there may still be usability problems that prevent this from being an acceptable design. The following evaluation is intended to elicit what these problems might be.

4. MEASURING IMPROVEMENTS

There are many factors to consider when trying to measure performance and, specifically, performance improvements for calculators. The declarative calculator has been designed to overcome identified problems, such as feedback and memory, so it would not be fair only consider tasks that examine these aspects. Simply giving users specific calculations to perform could bias towards one particular type of interface. To avoid these biases, the tasks need to be realistic and representative of typical tasks with these sorts of calculators. Rather than re-invent the wheel, it seemed sensible to use exam questions from GCSE mathematics papers that were specifically made to test people’s abilities to use calculators. At the very least, these are valid tasks for many 16 year old calculator users in the UK!

The interface should allow the user to perform the calculations quickly, without making mistakes and that the experience should not be dissatisfying [3]. Accordingly, we aimed to measure these factors. However, it is immediately obvious that there are individual differences in how well people perform mathematics problems and these might confound any timings. Users’ performances need to be measured against some baseline of their own individual abilities. One way would be to give users some trial questions and then measure their performance on those without a calculator. However, as the tasks are built to require a calculator, it would seem that they would either be too hard to do without one or they could be simplified to be achievable and then not representative of the actual tasks.

In our design, the time taken to understand a question was taken as the baseline measure as it would (in some way) reflect a person’s ability to comprehend and possibly plan a solution to a problem. These seemed sensible aspects of a person’s mathematical ability. Individual performance times would then be correlated with reading times rather than taken as absolute values. The effect of the calculator in helping or hindering the user would then be given by the steepness of the regression line. The intercept would reflect some measure of the basic ‘cost’ of using the device.

As will be seen, this approach did indeed give appropriate correlations and hence useful results.

5. EXPERIMENTAL METHOD

The declarative calculator was to be compared to a standard four function calculator. The declarative calculator has been implemented on an Apple Mac. To avoid issues of physical versus computer-mediated interaction, the HS-8V was also implemented in software. It was used rather than the built-in computer-based calculators because it reflected a slightly more constrained set of functions typical to handheld calculators and also because we had control over the implementation if there were any bugs.

Twelve subjects took part and each used only one interface so six people used each interface. The subjects were predominantly Masters students from our department aged between 22 and 35. All had used calculators at school and since then, though none of them had gone on to do a numerate discipline for their first degree.

Subjects were asked to answer five GCSE mathematics exam questions taken from the calculator paper that explored the range of basic calculator functions. The questions were chosen to avoid close matching with the strengths and weaknesses of the two interfaces used. Observation was done by an experimenter present in the room. Video observation was considered but it was felt that it might add to the stress and formality for subjects who might already be stressed by the thought of doing maths exam questions. Two major timings were recorded, being the time for the subjects to understand each question and the time to complete each question. Subjects had to declare when they had understood the question and then when they had completed the task to avoid any second guessing by the experimenter. The experimenter also observed any errors, being one of a restarted calculation, using incorrect values or using an incorrect method.

In a pilot study, users were initially given an example task to work through in order to familiarise themselves both with the interfaces and with the styles of interaction. However, the results from this were poor and subjects indicated that this might be because the questions were too hard particularly as it was a long time for some of them since they had worked on such questions. Easier questions were selected for the main tasks and, in addition, extra example tasks were given to the subjects before the main tests. Subjects performed these pre-test examples with pen and paper. Accordingly, the examples used simpler numbers than the calculator tasks but were intended to familiarise the subjects with the sorts of tasks that would appear in the actual tasks.

To measure user satisfaction with the interfaces, the SUS scale was used [1]. Although this is a "quick and dirty" measure, it is useful for getting some insight into how users feel about the interfaces and also for prompting for more detailed feedback on the user experience.

6. RESULTS

The total times for subjects are given in Fig. 2. There are no significant differences between the understanding times and the calculation times of the two different interfaces. This suggests that the two groups represent the same ranges of abilities. The correlation between the subjects' calculation time and understanding time are both significant with $r = 0.899$ ($p<0.01$) for the HS-8V and $r = 0.789$ ($p<0.05$) for the declarative interface.

The slopes of the regression lines for these data are 1.80 for the HS-8V and 2.67 for the declarative interface. The intercepts are 219 and 191. These figures mean that the basic cost of using the calculator is slightly lower for the declarative interface but that, after that is accounted for, a user would take a third longer to use the declarative interface to perform the calculation than the HS-8V.

The number of errors recorded were 9 for the HS-8V users and 5 for the declarative interface users. These were not large enough to perform any further analysis.

	Interface	**Total understanding time (s)**	**Total calculation time (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

Fig. 2: Total times for completing tasks

The SUS questionnaire scores came out at a mean of 47.1 for the HS-8V and 50.4 for the declarative calculator. The difference is not significant though interestingly the HS-8V had a much higher spread with SD of 10.1 against the SD of 2.9 for the declarative interface. It seems the subjects were more provoked one way or the other by the HS-8V however neither score indicates a particularly strong satisfaction with the interfaces.

Turning to specific user feedback, 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 write on the whiteboard 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 "=" to get an answer. This, however, seems like a carry-over from existing calculators as that is traditionally used to indicate 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.

7. DISCUSSION

The steeper slope of the declarative interface regression line over that of the HS-8V suggests that the declarative interface is not as quick at turning understanding into completed calculations. Or, put more simply, subjects performed less well with the declarative user interface. In many ways, this was to be expected. The subjects all had at least ten years of experience with existing interfaces and probably some of that time involved formal training on how to use them. The positive side though is that subjects did not perform so much worse with the declarative interface – it is not even a factor of two in reduced performance. A natural follow up might be to take a longer time period and to test subjects several times with both interfaces to look for both learning and improvement effects over a period of a few weeks.

Errors were surprisingly few with both interfaces. This may be due to the preparation time both before the tasks and for each individual task. If subjects are using the preparation time to devise essentially complete plans then they should largely be following that plan and only making slips rather than more substantial errors. It implies (life long) training and its associated assumptions are important, and perhaps disadvantaged the novel user interface evaluation.

The new declarative calculator increased user choice, and this may have increased task times. For example, the calculation described in Section 2 and 3 can also be solved on the declarative calculator in the *same* order as the English scenario by entering the sum '$\times$ 0.27 = 5 – 2.30' This additional flexibility, we know is appreciated by expert users, causes problems when users only have brief exposure to the approach.

The subjects did not seem particularly satisfied with one interface or the other. The main problem with the declarative interface seemed to be its blankness – there were no cues such as buttons or operations that suggested what the user could do. Also, users needed to hunt around the keyboard for some operations such as "." and "+". This may be to do with unfamiliarity with a keyboard (or those keys) than any feature of the calculator. However, it does suggest that adding buttons or cues of what to type might help the user. It may be interesting to try this calculator with a button interface of the sort used on the HS-8V.

The other cause of problems to subjects was the constant movement and updating of the interface. As they type, the computer is also filling in extra characters to make the equation correct. The user then is effectively working in a constantly shifting area which possibly undermines the blackboard style of the interface. It may be that some sort of button to relinquish user control may be required. Perhaps there should be some latency between the user typing and the computer taking its turn. What is positive though is to know that users appreciated the editor style interface. It would be interesting to find what users think when specifically memory intensive calculations were required and whether the memory advantages would outweigh the disconcerting behaviour of the interface.

These then are useful insights into the interface and suggest new elements that could be added. They also suggest obvious ways to re-test the interface as well.

8. CONCLUSIONS

The declarative calculator did not perform as well as hoped but then given the years of experience with traditional calculators, there is room to be optimistic (particularly with the lower error rates). The experiment suggests simple changes to the interface that might improve its acceptance and user satisfaction. Also given the matching with traditional pencil and paper mathematics, it may be that with the declarative calculator students can move away from being taught how to use calculators.

The regression applied between training tasks and experimental tasks is sound, and appears to be a new and promising approach for usability evaluation.

Finally, given that the declarative calculator interface is so different from a conventional calculator, the equivocal usability evaluation results suggest that great care must be taken in any usability evaluation.

9. REFERENCES

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

[2] Nielsen J. (1993) Usability Engineering. Morgan Kaufmann

[3] Preece J, Rogers Y. and Sharpe H. (2001) Interaction Design. Addison-Wesley

[4] Thimbleby H. (1986) The Design of Two Innovative User Interfaces. In Harrison D. and Monk A. F. (eds) Proc. of BCS HCI, CUP, 336-351

[5] Thimbleby H. (1996) A New Calculator and Why it is Necessary. Computer Journal 38(6), 418—433

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