A novel pen-based calculator and its evaluation

William Thimbleby
Department of Computer Science
The University of York
YORK
YO10 5DD
will@thimbleby.net

ABSTRACT
A novel calculator, ideal for interactive whiteboards and pen-based devices, is introduced and evaluated. The calculator provides a natural, dynamic method of entering conventional expressions by handwriting and provides continual feedback showing the calculation and results. The user interface adjusts and copes with partial expressions, morphing the expressions to correct position and syntax. Gestures are also used to edit and manipulate calculations. The user interface is declarative, in that all displays, even with partial user input, are of correct calculations.

The new calculator is faster for more complex expressions and importantly, gives users more confidence in its results. The majority of users said that they would prefer to use this calculator rather than their conventional calculator.

Author Keywords
Handheld calculators, gesture input, novel interfaces.

ACM Classification Keywords
H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

INTRODUCTION
Imagine writing a calculation down on paper, and the paper magically working out the answers. We have built a calculator that works like this, which is ideal for pen based user interfaces, or for interactive whiteboard use in classrooms. This paper discusses the design and its evaluation. The calculator is written in Java and is available with tablet drivers for use under MacOSX. (It will be demonstrated in the conference presentation.)

Overview
Refer to Figure 1 at the top of the next page, which shows screen snapshots of the new calculator in use. We first show a user doing the sum 3+6. In the first screen shot, they have written 3+6, in the next the calculator is catching up with them and has already rendered the 3 and + in a printed font. In screenshot 3 the calculator has morphed the input into a nicely typeset equation. The user then clears the screen, screenshot 4, using a cross-out, X, gesture; the feedback from deletion, the 'smoke poofs' are also visible. The final two screenshots show the user entering 2/3×=4, the declarative calculator ensures the answer is correct, and the interface morphs the answer into a nicely typeset and readable equation.

History and Motivation
Somewhere around 200AD, the abacus was invented. Since that time we have always used instruments to aid our mental arithmetic and to help us with mathematics. In the 1970s with the development of electronics, electronic calculators became popular. For the most part their design copied earlier mechanical calculators. Now, thirty years later, when desktop and handheld computers can do almost anything, today's calculators merely imitate early electronic calculators. The calculator provided in the Start menu by Microsoft is less powerful, and less expressive, than a 10 year old handheld calculator! Yet computers today could do a lot better than just simple imitations of mechanical calculators.

The majority of current research on expression recognition has been directed towards that of expression entry [1,2,3], although there have been attempts to marry expression entry with calculation (for example, the PenCalc project [4]. Yet, none of the existing implementations have attempted to use expression recognition itself as a user interface for a calculator.

The calculator presented here, and its design extends the domain of calculator user interfaces into the 21st century. Rather than relying on obsolete metaphors that dictate awkward and unnatural mathematical entry, this calculator provides a natural interface that is designed to (and does) function like pen and paper — or, rather, paper that does mathematics magically.

PEN-BASED USER INTERFACES
Few papers have been published regarding pen-based interfaces. PenPoint [5] is one of the few complete systems de-

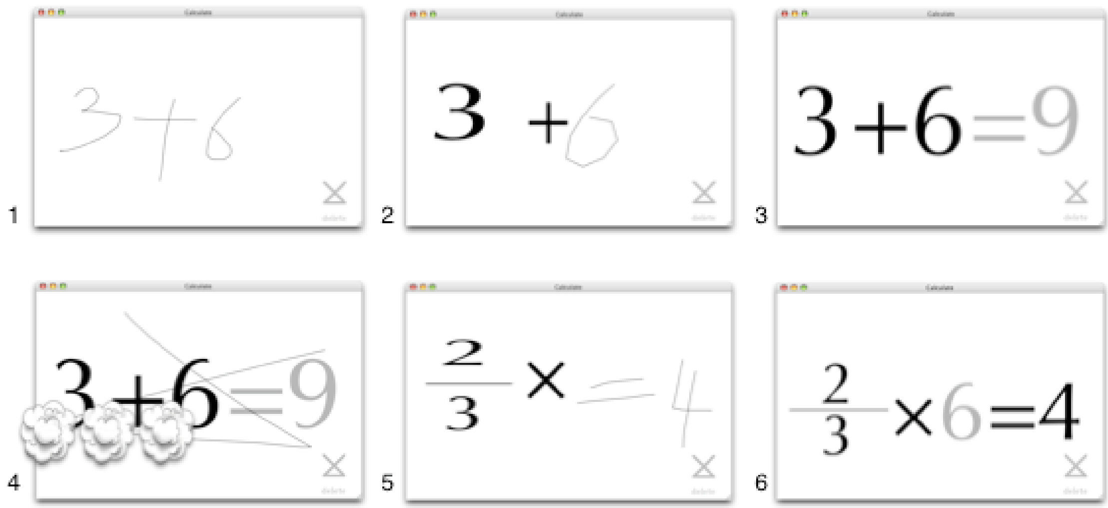

Figure 1. Screenshots of the calculator's progress computing, 3+6 and then 2/3×?=4.

scribed. Research on specific areas of pen based interfaces has been conducted, notably text entry, editing, and the use of special gestures to control additional functions.

Meyer [6] gives a good detailed overview of the whole technology, including both a history of pen based computing and more technical aspects of the hardware and software.

The main advantage of a pen-based system is the familiarity of the interface. The majority of users are already competent at writing with a pen. This advantage is greater with mathematical expressions, because the majority of mathematical work is still done on paper with a pen or pencil. Using a pen-based system to enter expressions is a natural progression, as it means that anyone can use it with little or no training.

Another advantage is there is no need for any other interface. Pens can replicate the complete functionality of both keyboard and mouse.

USER INTERFACE REQUIREMENTS

Ideally a user interface for mathematical input should provide a superset of paper's functionality, allowing a user to use the interface in the exact way they would use paper. The key features of the paper metaphor are outlined below.

Sketching

Ideally a user could draw rough sketches. Often when solving a problem a user will not jump straight onto her computer and solve it, but jots down diagrams or figures first. The system presented does not implement sketching, however a solution is to specify areas for diagrams and mathematics.

Expressions

The system should allow the user to enter expressions as they would on paper, without unnatural restrictions. For instance, the user should not be forced to enter the expression in a linear fashion.

Editing

It should be possible to edit expressions at any time. Both input and output expressions (that is, an expression just entered, and one that has been computed) should be treated in the same manner. High-level editing, such as rearranging, insertion and deletion should be possible. To implement these with a pen-based interface, without leaving the paradigm of pen and paper, requires special gestures that are assigned to each of the editing functions. For example, a scribble is used for deletion. Character editing is different and involves correcting the computer guess at the semantics of a set of strokes.

Feedback

The system should always keep the user informed about what is going on, providing appropriate, timely feedback.

THE DESIGN

Our new calculator uses a single canvas for mathematical expressions, which enables us to create a completely modeless interface that is intuitive and natural. The user interface is shown in Figure 1. The operation of the user interface was developed from Thimbleby [7].

The one adornment of the interface is a delete gesture reminder in the bottom right corner.

Expressions

The system imposes small timing constraints. It requires the strokes of contiguous symbols to be written sequentially within a small amount of time. This allows the interface to recognise the user's input on-line as they enter it. These small restrictions were found to be unobtrusive, and do not affect the user's writing style.

Editing

After testing several different ways of editing expressions, it was found that for the majority of mathematical expressions, the easiest way to edit them was to delete and rewrite portions of the expression. This keeps the interface very simple. By allowing dragging or pop-up menus creates areas of the screen that function differently from each other. By allowing only a simple delete gesture, no mode changes are necessary. Every part of the screen or virtual paper acts like paper: every click and drag draws.

Feedback

The visibility of the system's status is provided through two kinds of feedback: annotation and morphing. As the user is writing, the system can process in the background. As a symbol is recognised the user is informed of this recognition by visual feedback: a typeset character stretched to the stroke's bounding box replaces the written strokes.

Morphing starts after a short time delay from when the user stop writing, and halts when the user starts writing again. This stops a morph from distracting the user and from rearranging expressions as they are trying to enter them. The morph formats the entered expression into a correctly typeset equation by moving the symbols as little as possible from the user's writing. The morph provides continuity between the user's input and the typeset equation that allows them to continue to edit and use it.

EVALUATION

A total of nine participants took part in the usability testing. The testing comprised of a number of mathematics questions based on old GCSE papers (i.e., exam papers for 16 year olds). Users were given time to familiarise themselves with the interface. When they were happy they were observed and recorded whilst attempting to complete the questions using the new system and their own pocket calculator or one provided (a Sharp EL-531GH DAL).

An observer was present and users were encouraged to discuss problems with them, and afterwards the users filled in an anonymous questionnaire.

Measurements were recorded of the time taken and the number of errors or problems encountered entering expressions. The questionnaires provided a better general impression of the ease of use.

RESULTS

Time on Task

For the simpler sums, like 9×2/3, a handheld calculator was much faster than the new system. This was expected. The majority of users were familiar with handheld calculators, and had used them over many years.

Two of the tasks were actually faster on the new system:

Calculating $\frac{2^2}{21-4}$ was faster because users could enter it "as they saw it" rather than having to search for buttons on a calculator.

For the task "What power of two is 28?" *every* user was able to complete the task on the new system, yet most struggled to solve it on a handheld calculator.

Thus the new calculator enables users to perform mathematics that they could not do before. Furthermore, it is faster for more complicated expressions because users do not have to rearrange the expression in their head. This was the case *even* for users who knew the rearrangement log 28÷log 2 finds the power of 2 equal to 28.

Accuracy

A large part of the time taken to complete the tasks was taken up with recovering from symbol recognition errors. Currently the accuracy percentage of 81.1% is poor. This significantly lowers the usability of the system. Expression recognition caused only a very few errors, mostly caused by short divisor bars.

However, when calculating mathematics, input accuracy is not the most important consideration; output accuracy is. No user got the wrong answer for any question with the new system.

By displaying the computed equation in an easily understandable two-dimensional format, it provided the feedback necessary to understand what was being computed. Importantly, users *never* arrived at an incorrect answer with the new calculator, compared to several simple mistakes that went unnoticed when using traditional calculators. Evidently users knew if and when their calculations were wrong and when they had to be corrected.

Ease of Use

The overall impression from users was that the new calculator was *easy* to use. Typesetting and feedback through morphing successfully allowed the user to understand what the calculator was doing.

No user had trouble editing expressions using the delete gesture. Other editing functions like cut and paste were never missed and users liked the modeless interface and the simplicity of one function.

FURTHER WORK

During both testing and design many ideas were developed that provide possible avenues for further development. The more interesting are outlined here.

Extended Vocabulary

Expanding the number of symbols recognised to include symbols like π, letters, and other Greek characters, would enable the system to handle more complex expressions.

Additionally, extra functional vocabulary would allow the system more power and expressiveness. For instance trigonometric functions, user defined constants, logarithms, and factorials.

The User Interface

Further additional features of paper (for instance, sketching) would add to the usability.

Users specifically requested two additional features. A clear button clears the whole screen, a similar metaphor to starting a new page. This could be provided as a simple gesture or an external button.

Secondly, users found that there was sometimes not enough room to enter their additional symbols into an existing expression. Two solutions for further work would be the addition of an insert space gesture that adds in a gap into an expression and the re-morphing of an expression as a user writes to accommodate the user's input.

Hardware

The most obvious, and possibly effective, development of this system is to build it into existing systems: PDAs, tablet computers and smart whiteboards

CONCLUSIONS

At its most abstract, this paper described a novel pen based interface for any application. This paper described and evaluated the pen-based interface for a dynamic, on-line mathematical calculator.

New user interface concepts for the computation of mathematics were introduced, including:

- Automatic annotation and morphing of symbols into correct and pleasantly arranged calculations after the user has finished writing.
- Extra space added to calculations, like longer division bars, to aid the addition of more symbols.
- The use of a single delete gesture to edit expressions, making the calculator completely modeless and providing the user with an extremely simple interface.
- The use of a declarative engine to ensure that calculations are always correct and that the expression being computed is always displayed. This enables the user to have complete confidence in their answers.

The comparative user testing comparing the new system with traditional calculators showed that:

- Answers produced by the new calculator were more accurate. In contrast, users failed to notice when a traditional calculator gave them a wrong answer — errors that they noticed when using the new calculator.
- Users were able to calculate the answer to problems using the new calculator that they could not solve using traditional calculators.
- Users are able to obtain accurate answers and have greater confidence in those answers compared to results from traditional calculators.
- A pen based calculator is more intuitive, fun, and easy to use than traditional calculators.
- The pen is a suitable device for entering and editing mathematical expressions. Additionally, more complex editing operations than delete are neither necessary nor missed.
- For complex calculations, the new design was faster than using traditional calculators.

It is hoped that the creation of this new calculator will prompt people to rethink the methods by which we do mathematics. (The calculator is available on the web for others to build on.) Calculators are currently restricted by obsolete metaphors, as the testing and creation of this new calculator has shown.

Ultimately, the calculator should be ported to and tested on pen based, handheld computers and tablet PCs, as well as in school classrooms (e.g., using projectors and touch-sensitive whiteboards) where they would be an ideal way of teaching mathematics to children.

We are confident that the prototype described in this paper charts a course in the right direction.

AVAILABILITY

A movie of the calculator and the Java application are available at http://www.uclic.ucl.ac.uk/will/

REFERENCES

1. Dorothea Blostein and Andy Sch uerr. Computing with graphs and graph transformation. Software Practice and Experience, 29(3):1-21, 1999.
2. Yuko Eto and Masakazu Suzuki. Mathematical formula recognition using virtual link network. In Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR '01), page 762. IEEE Computer Society, 2001.
3. Steve Smithies. Freehand formula entry system. Master's thesis, University of Otago, Dunedin, New Zealand, May 1999.
4. Pencalc http://www.cs.ust.hk/pencalc/.
5. R. Carr and D. Shafer. The Power of PenPoint. Addison-Wesley, 1991
6. A. Meyer. Pen computing: a technology overview and a vision. SIGCHI Bull. 27(3):46–90, 1995.
7. H. Thimbleby. A new calculator and why it is necessary, 1996.