

# Evaluating a Sketch Environment for Novice Programmers

**Beryl Plimmer**

Department of Computer Science  
University of Auckland, Private Bag 92019,  
Auckland, New Zealand  
b.plimmer@auckland.ac.nz

**Mark Apperley**

Department of Computer Science  
University of Waikato, Private Bag 3105,  
Hamilton, New Zealand  
m.apperley@cs.waikato.ac.nz

## ABSTRACT

This paper describes the evaluation of an electronic sketch interface design tool for novice programmers. A comparative study was undertaken with small groups using two different shared space environments; a conventional informal design environment and the pen based digital whiteboard. The students reacted positively to the electronic environment, where they worked informally with their design ideas and checked them more carefully.

## Keywords

Tool evaluation, sketching, novice programmers

## INTRODUCTION

Hand sketching has long been the preferred method for rendering initial design ideas because its informality allows the designer to work with partly formed ideas and ambiguity. We have developed a form design sketch tool that is tightly integrated with a programming IDE and which utilises a pen-based digital whiteboard. The tool also facilitates using scenarios to check a design which is a powerful way to improve problem understanding. This paper describes the evaluation of this environment as a tool for novice programmers.

## BACKGROUND

A number of sketch design tools have been produced for different applications [1-3]. We developed a sketch environment (Freeform) that is tightly integrated into the Visual Basic© (VB) IDE, using a low-cost digital whiteboard to provide pen input.

The digital whiteboard is a large opaque glass screen with a Mimio© attached, on to which the standard computer screen is back projected. The software runs as a VB add-in; it provides a sketch space, storyboard, run mode, sketch recognition and sketch to VB form conversion.

In the sketch space users can draw, write and edit using the Mimio pen. Editing functions to move, resize, copy, paste, and undo are provided. Care has been taken to retain the informality and imagery of a whiteboard; we

chose to use pens as the only input device. Requirements for public space pen input are different from those for a private space such as a PDA; for example we do not use off focus writing. The storyboard provides a miniature view of all sketches and allows the user to draw navigational links between sketches.

In run mode the user can check the forms by filling in scenarios and following the navigational links between forms. When satisfied with the design the user employs the recognition engine to overlay the sketch with the VB control names and recognized words. The user can alter any recognition and then have the software automatically create a VB form.

This software went through two prototype cycles of development, usability testing and redevelopment before the evaluation study described below.

## APPROACH

This study evaluated Freeform as a tool for novice student programmers using a comparative study of two environments. The evaluation consisted of questionnaires, observations of the researchers, an evaluation of the design products and a review of the learning process.

Eight groups of two or three first year tertiary students completed two design tasks in two different environments, each for a different problem. In the 'natural' environment the groups sketched their designs on a whiteboard, checked them with the scenarios provided and then created the form on a standard computer using the VB IDE. In the Freeform environment described above the users were first trained on the environment and then instructed to design the form, check it using the run mode and use the software to convert it into a VB form.

The problems given to the student were a simplified book catalogue form, and a dog registration form. They were carefully constructed to be of equal difficulty yet to be sufficiently different so that the second was not simply a reconstruction of the first. For each problem we also provided two scenarios that the students could use to check their designs.

Four groups used the natural environment first, two doing the book catalogue problem and two the dog registration

problem. They then completed the other problem in the Freeform environment. The other four groups worked first with Freeform and then with the natural environment.

The students answered three questions on their current experience, practice and beliefs before the first task and then a set of questions on each task after it was completed and a further concluding question at the end of the study.

### FINDINGS

The students completed the questionnaires individually but we took the more conservative mean response for each group for the statistical analysis. The students were very positive about the Freeform environment. An analysis of co-variance for the task specific questions found that of the ten questions asked four were more positive for the Freeform environment at a greater than 95% confidence level. They were: This exercise was enjoyable; This exercise motivated me to find out more about programming; I would like to use this method of program planning in the future; This experience has made me value sketching interface designs more. A further two were more positive at a greater than 90% level: I feel prepared to complete the program; Checking the scenarios was easy.

Of the remaining four questions two were influenced by the order of the task, with the second task being easier, and the other two favoured Freeform but not at a statistically significant level. Seventeen of the twenty participants stated that Freeform would be their preferred design environment. Finally a comparison between their view of the value of sketching before the study and after each task showed that Freeform gave a significant boost to their rating of the importance of sketching designs.

The designs were independently evaluated by an expert HCI educator. In most cases the groups did a better job of the design in the Freeform environment.

We also observed that the groups made many more changes in the electronic environment. The mean number of changes in the natural environment was 1.12 compared with 3.5 in Freeform. Most changes were made after checking with the scenarios in run mode. In Freeform the students used the scenarios to fill in the form where as most groups only visually checked the form on the whiteboard. An example of a change that was made after trying the scenarios was the dog registration form address fields. Most groups initially drew a single-line textbox to hold the address, but the scenarios had multiple address lines. In the natural environment three of four groups finished form had a single-line textbox. In the Freeform environment three out of four groups finished form had space for multiple address lines.

This was not a result we had expected, we asked an educational psychologist to look at the videotapes of the sessions. He suggested that in the electronic environment there is lower cost for change because of the ability to

move and resize elements and lower risk because of the undo facility. The run mode encourages active participation in the checking process where the normal whiteboard checking was more passive. Along with this the immediacy of the sketch and check with the electronic environment provided quick feedback and completion of the learning cycle which is likely to result in more activity.

### DISCUSSION

This study was conducted with a small number of groups and the problems were relatively trivial, yet typical for novice programming courses. Nevertheless the study provided some interesting insights into how novice programmers attack interface design and problem solving. The enjoyment and enthusiasm at using such a tool in a small group was evident from the questionnaires. The better designs produced in the electronic environment were mostly as a result of active checking with the run mode. It is possible that using the same checking technique on a VB form in run mode would result in similar changes; however the work of others [4] suggests that we become more committed to a design that looks finished and are therefore less likely to change it.

### CONCLUSIONS AND FUTURE WORK

This study has suggested that a group informal design environment is likely to be useful to novice programmers because it encourages them to hand sketch interface designs and facilitates the easy checking and changing of the design before they commit to it. We plan further development of the environment and to test whether scenario checking of a completed VB form results in the similar sorts of changes.

### ACKNOWLEDGMENTS

The assistance of Matt Jones, Ray Littler and Tony Morrison for their help with respectively: the evaluation of the designs, statistical analysis, and comments on the learning processes, and the many students who have contributed to this and earlier studies is gratefully acknowledged

### REFERENCES

1. Gross, M. and E.Y.-L. Do. *Ambiguous intentions: a paper-like interface for creative design*. in *UIST '96*. 1996. Seattle Washington: ACM.
2. Landay, J. and B. Myers, *Sketching Interfaces: Toward more human interface design*. *Computer*, 2001. **34**(3): p. 56-64.
3. Damm, C.H., K.M. Hansen, and M. Thomsen. *'Tools support for cooperative object-oriented design: Gesture based modelling on and electronic whiteboard*. in *Chi 2000*. 2000: ACM.
4. Wong, Y.Y. *Rough and ready prototypes: Lessons from graphic design*. in *Human Factors in Computing Systems CHI '92*. 1992. Monterey.