

# Introduction

The purpose of this study, is to gauge the quality of the Gaffe Designer's user interface, and to identify parts that need improvement. Gaffe is a program made for my Masters thesis.

## Background

Z is a language used for writing specifications of systems (including software). Writing a specification in Z, instead of English, can help to find contradictions and ambiguities before work starts on creating the system; it can be used for creating test-sets, and also has many other advantages. Unfortunately understanding a Z specification takes some amount of experience. So if the client commissioning the system doesn't know Z, then it is not easy to tell if the client's vision for the system matches the specification. This is where Gaffe comes in...

Gaffe is a GUI front-end that sits on top of a Z animator. A Z animator is a tool that allows a user to try various operations described in the specification, to see what results can be produced; the animator sort of emulates the system described by the specification. Typically, though, animators require experience to use, which does not help with the client mentioned in the previous paragraph. Gaffe will present a more intuitive user interface. The interface could be generated from a Z specification using the Gaffe Generator, or designed by a person using the Gaffe Designer. This study is testing only the Gaffe Designer.

## This Session

The following will happen during this session:

- **Consent Forms:** You will be asked to read the 'Research Participant's Bill of Rights', and to read and sign the 'Research Consent Form' (there are two copies, one for you to keep).
- **Initial Questionnaire:** To determine how much experience you have with related tools.
- **Exercises:** A sequence of exercises working in the Gaffe Designer, using an interface based on one created by the Gaffe Generator. Please feel free to think aloud, ask questions, and describe what you are doing and why;

this will help with the study. While you are performing these exercises, I will be taking notes on how you carry out these tasks.

- **End Questionnaire:** To identify difficult parts in the exercises, and hence what parts of the Designer need improvement.

During the questionnaires and exercises, I may ask you questions to clarify your answers, and to understand the reasons for actions that you take.

**Remember, this session is not testing you, it is testing Gaffe.**

# Initial Questionnaire

## Contact Information

1. Name:
2. Email address:

## Experience with Z

1. How much experience (if any) have you had with the Z language?  
None 0 Very Little 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 I'm an expert
2. How much experience (if any) have you had with any other specification languages?  
None 0 Very Little 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 I'm an expert
3. How much experience (if any) have you had with Z animators (e.g. Jaza)?  
None 0 Very Little 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 I'm an expert

## Experience with Application Builders

1. How much experience (if any) have you had with graphical application/interface builders?

e.g. Glade or QT designer under Linux, or Visual Studio's facilities for visually designing windows.

None 0 Very Little 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 I'm an expert  
Please Explain.

2. How much experience (if any) have you had with JavaBeans?

None 0 Very Little 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 I'm an expert

# Exercises

The Gaffe Designer is used to lay out ‘beans’ on a window. The features of a bean can be changed by editing its properties. Some beans can join to a script using an ‘event link’. The bean (e.g. a button) can cause the script to run (e.g. when the button is pressed).

1. Go to the ‘BirthdayBook’ form, and highlight all event links.
2. Choose the panel containing just the ‘AddBirthday’, ‘FindBirthday’, and ‘Remind’ buttons, and change its background colour to something more festive.  
If you like, change the colour of some other panels.
3. Delete the Solutions part of the history navigation panel. The Birthday-Book specification is non-deterministic, so it will not be needed. Don’t forget to delete the scripts associated with the buttons you deleted.
4. Make a new form called ‘Test Window’. Put a label in it with some text (“Happy Birthday!”), and a button labelled ‘Close’.
5. Make a new button labelled ‘Test’ on the ‘BirthdayBook’ form in the panel containing the operation buttons. Make the new button open the new form you created. Make the close button in your new form close the form. (Hint: Any scripts you make will be almost identical to scripts attached to similar buttons).

If you want to test your changes at any stage, you can save then run

```
gaffe-animate
```

at the command line.

# End Questionnaire

1. How easy did you find it to change the properties of beans?  
Very Hard 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 Very Easy  
Please Explain.
2. How easy did you find it to design the new form?  
Very Hard 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 Very Easy  
Please Explain.
3. How easy did you find it to create the necessary scripts?  
Very Hard 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 Very Easy  
Please Explain.
4. How easy did you find it to hook up event links?  
Very Hard 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 Very Easy  
Please Explain.
5. How well do you feel you understood what was happening while you were using the Gaffe Designer?  
Didn't Understand 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 Understood Completely  
Please Explain.
6. How easy did you find it to use the Gaffe Designer?  
Very Hard 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 7 Very Easy  
Please Explain.

7. Which aspects of Gaffe did you find the easiest to use?

8. Which aspects of Gaffe did you find the hardest to use? Why?

9. Were you surprised by any part of the Gaffe Designer's behaviour? If yes, please explain.

Please, feel free to add any other comments you may have.