

Teaching the Empirical Approach to Designing Human-Computer Interaction via an Experiential Group Project

Matthew C. Clarke

Computer Science and Information Systems

University of Natal, Pietermaritzburg

South Africa

Abstract

Empirical research plays an important role in the design of user-interfaces and is frequently included in university courses on human-computer interaction. For instance, the ACM SIGCHI guidelines refer to the importance of empirical research, although they do not specify how this approach to user-interface design should be taught. In an Honours (fourth-year) course at the University of Natal, Pietermaritzburg, the theoretical foundation of empirical research is augmented with a real experience of running a simple experiment. This experiment is planned, executed and analysed by the class as a whole. This paper describes the type of empirical studies carried out and discusses the benefits and limitations of such studies in this educational context.

1. Curricula for Human-Computer Interaction

Eberts and Eberts [5] identify four strategies for making decisions about user-interface design issues — the empirical, cognitive, predictive modelling and anthropomorphic approaches. Theoretical aspects of each of these often appear in human-computer interaction (HCI) curricula and some curricula even include project work to add a practical experience to the theory. There is a great benefit to be gained by students actually experiencing these design strategies rather than just learning about such research methods theoretically and it is the thesis of this paper that such experiences can and should be included in HCI courses. The paper suggests the sort of project which would provide the experiential element for the empirical approach.

The ACM SIGCHI has suggested four curricula for HCI, namely *The Human Aspects of Information Systems*, *User Interface Design and Development*, *Psychology of Human-Computer Interaction* and *Phenomena and Theories of Human-Computer Interaction* [1]. The last of these includes some exposure to empirical methods, and suggests additional readings to support this topic [2, 12]. But the curriculum in which empirical methods play a larger role is *Psychology of Human-Computer Interaction*. This curriculum assumes a background in applied statistics and experimental methods, and suggests that a later HCI Laboratory course might be the appropriate place in which

to put the theory into practice. None of the four curricula suggested by the SIGCHI include practical experience of empirical research methodology.

There are at least the following two reasons for thinking that a greater emphasis on experiential exposure to empirical methods would benefit the student of HCI rather than just giving them a theoretical exposure to these ideas. Firstly, a well-established principle in much recent educational theory indicates that learning is constructed on experience. That is, experience provides the concrete foundation for the learning process. Understanding of abstract concepts is built through observation and reflection on experience [7]. One needn't be a Constructivist to recognise the difficulty of teaching abstract concepts when the learner has no experience of any instances of those concepts. In an HCI course which covers empirical methodology, the theoretical issues are more likely to be properly understood if the learner has seen examples of actual empirical research and even more so if they have participated in such research.

Secondly, a wealth of research shows that students have difficulty transferring concepts learnt in one domain to a new domain (see, for instance, [11]). They may learn of empirical research methods in another course, but not be able to apply these methods to user-interface design. Unless connections are explicitly made during the teaching process, it is likely that students will not make a clear connection between what they learn in an applied statistics course and what they learn in an HCI course. Participating in an experiment related to user-interface design is an effective technique for overcoming the difficulties of transfer.

2. Outline of the Teaching Approach at the University of Natal, Pietermaritzburg

At the University of Natal, Pietermaritzburg, Honours students (that is, students in their fourth and final year of a Bachelor's degree) may enrol in the course Human Factors of Computers which includes a substantial section on user-interface design. Various approaches to user-interface design are studied, supported by readings such as [4], [5], [6] and [10]. The theory presented in the lectures and readings is augmented with a major project which is completed by the class as a whole. The project requires the class to work together on the design, execution and analysis of some actual, though un-original, empirical research, typically in the form of a single-factor experiment.

Over an eight-week period, the Lecturer guides the group through the standard process of the experimental method. Some class time is spent on this project, but the majority of work is carried out by the students outside class time. The phases of the experimental method — conception, design, execution, analysis, and dissemination and decision making — are well described in Chapter 4 of [6], adequate extracts of which were previously published as [9]. These six phases are applied to the HCI project as follows.

2.1. Conception

To begin with, the Lecturer presents several possible topics for study but the class should be allowed to suggest other topics. Through negotiation, the class chooses a topic which is both interesting and feasible. The chosen topic must be simple enough to provide a meaningful experience within the constraints of time and student competence. Rather than engage in pioneering work, we have always chosen topics which have been previously researched — the class's objective is then to attempt to confirm previously reported conclusions. The following topics have been examined over the past several years —

- *The effect of keyboard layout on novice typing speed.* The class set up four types of keyboard — Qwerty, Dvorak, alphabetic and one on which the keys were arranged to read “The quick brown fox jumps over the lazy dog”. Subjects with no prior typing experience were set some simple typing exercises and their speed (and incidentally their error rates) was measured over a twenty minute period. The data allowed some discussion of which layout was best suited to novice typists.
- *The effect of screen colours on time and error rate.* In response to some earlier theoretical discussions on the importance of choosing appropriate colour combinations in computer software, the class in one year decided to study which colour combinations really did enhance or detract from efficient task completion. All subjects were assigned the same sequence of object-counting tasks but the foreground and background colours varied for each treatment group. By comparing the time taken to complete the tasks and the extent of mis-counting in

each treatment group, some conclusions could be drawn about the visual efficacy of the tested colour combinations.

- *Intuitive interpretations of icons.* One type of icon commonly used in graphical user-interfaces is a button which represents the status of some feature. Such a button has two positions — in and out — which correspond to the feature being on and off. Without additional cues, the setting of a button is ambiguous — does the *in* position represent on and *out* represent off, or vice-versa? Even when labels such as “On” or “Off” are added, an ambiguity remains — does a button labelled “On” mean that the feature is currently switched on, or that the button should be pressed in order to turn the feature on? The class created an artificial task in which fictional room lights were to be turned on and/or off using simple button icons. One treatment group was given buttons with labels and another without. The way the buttons were configured by the subjects in order to fulfil the set tasks indicated the subjects' natural interpretation of the buttons' two positions.
- *The effect of font on reading speed.* According to received wisdom, large amounts of text are more easily read when typeset in a serif rather than sans-serif typeface. The class decided to test whether this was still the case when text was being read from a computer screen rather than from paper. Two treatment groups were given the same text to read, but presented in different fonts. Three simple comprehension questions were asked at the end to ensure that the automatically-recorded reading times were not invalidated by subjects who didn't actually read the text.
- *Technophobia.* In the latest version of the course, the class steered away from the previous sort of experimental studies and opted to replicate a Technophobia survey which had previously been undertaken in 23 other countries [13]. Although this study did not involve much design since the survey instrument was identical to that used in previous studies, this still provided a rich experience of data collection, analysis and report writing.

After a topic has been selected, the students are required to read previous research in the field in order to gain an understanding of the issues they will address in their own experiment. This background reading may include references supplied by the Lecturer, but should also require the students to search for references themselves through information resources in libraries and on the Internet.

2.2. Design

As a group, the class and Lecturer need to precisely define the goal of the research in the form of null and alternate hypotheses. These terms need to be properly theoretically grounded through readings and input from the Lecturer, but the hypotheses should not be given to the class by the

Lecturer. Rather, the students should show their understanding of the theoretical issues by coming up with an appropriate wording themselves.

The construction of hypotheses will also require an understanding of the dependant and independent variables. Once again, these and other potentially confounding variables should be identified by the class with the Lecturer playing a facilitative role.

Once the goal has been clearly specified, the class determines which experimental structure will best test the hypothesis. The best approach may be to replicate a previous experiment, but whatever the decision, it will be based on the literature review already completed. Associated with the experimental structure, further decisions are made about what data needs to be collected and how will it be analysed.

Apart from the design of the experiment, this phase also includes the specification of any software (and perhaps hardware) required for the experiment. It is helpful at this stage to split the class into task groups to subdivide responsibility for the hardware/software preparation, and for each of the following phases.

2.3. Preparation

The Lecturer will probably be required to assist with logistical matters such as booking a venue for the experiment at appropriate times. Some arrangement must be made for enlisting subjects, typically student volunteers.

Appropriate hardware must be configured and the specified software must be coded and tested.

2.4. Execution

Once the preparation is complete — the experimental apparatus set up and tested and the subjects scheduled to turn up to the appointed venue — the task group responsible for running the experiment takes over. In the first four examples given in Section 2.1, the experiment entailed subjects interacting with PCs in batches of 30 (dictated by the capacity of the computer lab). Each experimental session follows the same pattern — the subjects are seated; instructions are given by one of the students; the subjects complete the required tasks while the computer automatically records the necessary measurements; the subjects are thanked for their time as they leave.

To maximise participation, we have usually allowed several students to instruct and supervise the subjects during the experiment. However, a script is prepared beforehand so that the instructions are consistent across multiple experimental sessions.

2.5. Analysis

The data from the numerous experimental sessions is collated and analysed by the next task group. The analysis is reasonably simple given the simple structure of the

experiment and may be undertaken with a standard statistical package such as Statgraphics or SPSS. The analysis typically requires hypothesis testing based on a single-factor experimental design.

2.6. Dissemination and decision making

The final task of the project is to write up the experimental findings in an acceptable academic style. This report should include the background to the research, the goal and methods used, the results of the data analysis and a discussion of their implications for future user-interface design. The report should also show a clear understanding of the limitations of the research and describe the lessons which the class learnt from the experience. One task group may be responsible for collating and formatting this report, but all task groups should write up their own section.

3. Benefits and Limitations

As indicated in Section 1, the two intended benefits of project work such as that suggested here relate to the educational value of building theory on experience and of making explicit the application of previous knowledge to the current context. However, there are benefits other than these, as well as some limitations.

One notable result of providing a practical experience of empirical research is that students' enthusiasm for the course seems to be increased. The practical experience helps to make the theory real for them. Working through the practicalities of an actual experiment motivates the student to understand the theory. Consequently, the students complete the course with a much greater understanding of the benefits and limitations of the empirical approach.

Apart from learning about the content of the project (i.e. the specific empirical issue being studied) and learning about the process of empirical research, these sort of group projects also promote learning about how to manage the logistics of venues, resources and people (both team members and experimental subjects), about written and verbal communication, and about teamwork and interpersonal interaction.

Computer science students are not known for their abilities in written communication, however, they frequently find themselves in situations requiring such abilities after they graduate. In academia, industry and commerce computing professionals need to be able to write well for technical documentation, user guides, tenders and tender responses, funding proposals, project progress reports etc. A computing degree which does not include some substantial writing does not equip graduates for these needs.

Projects of this sort will rarely yield significant results. Since the project has to be completed within the time and resource constraints of a one-semester course, the sample size is likely to be fairly small. There are likely to be unforeseen problems when it comes to actually running the experiment but there is unlikely to be sufficient time to rectify any

mistakes or repeat the experiment¹. This may be seen as a limitation, but it must be kept in mind that the aim is not to produce novel research, but to use a simple research topic for the purpose of educating the students. The students are novices and should not be expected to execute the research perfectly. Nevertheless, they should not be seen simply as research assistants to the Lecturer — the class needs to take responsibility for the project and its consequent success or failure.

Group work itself can cause some difficulties such as personality clashes and mis-communication. The Lecturer will need to play a facilitating/managing role to ensure that the group operates effectively. Group work also raises the question of how individuals within the group will be assessed. It may not be the case that all members of the group contributed equally and so it need not be the case that all members receive the same mark for the project. There are various strategies for handling group mark allocation (see [3] and its reference list for general comments, and [8] for an example in a computer science course) and this need not be seen as a problem.

4. Conclusion

Empirical studies play an important role in helping to make choices between user-interface design options. Many courses in human-computer interaction note the importance of empirical methods and in particular of experimentation. This paper has suggested that the topic of empirical research methodology should not just be presented to students in the abstract, but that their grasp of the topic will be greatly enhanced by giving them a real experience of designing and running an experiment, and of analysing and reporting on the resulting data.

Such experiences in empirical methods can be orchestrated to suit the abilities of novice researchers and the constraints of a one-semester course. The idea has been successfully implemented for several years in at the University of Natal, Pietermaritzburg.

The benefits of such an experience may be summarised as follows —

- The project provides an experiential basis for the subsequent understanding of theoretical issues in empirical research.
- The explicit link to prior knowledge aids knowledge transfer from the general domain of statistics to the specific domain of human-computer interaction.
- The novelty of the experience increases student interest and motivation.
- The process provides an exposure to broader life-skills such as verbal and written communication, and inter-personal interaction in a team.

References

1. ACM SIGCHI Curricula for Human Computer Interaction, ACM Press, 1992
2. Campbell, S. K. *Flaws and Fallacies in Statistical Thinking*, Prentice-Hall, 1974
3. Conway, R., Kember, D., Sivan, A., and Wu, M. Peer Assessment of an Individual's Contribution to a Group Project, *Assessment and Evaluation in Higher Education* 18, 1 (1983), pp. 45–56
4. Finlay, J., Abowd, G., Beale, R., and Dix, A. J. *Human-Computer Interaction*, Prentice-Hall, 1997
5. Eberts, R. E., and Eberts C. G. Four Approaches to Human-Computer Interaction. In Hancock, P. A., and Chignell, M. H. (eds) *Intelligent Interfaces: Theory, Research and Design*, North-Holland, 1989
6. Fenton, N. E., and Pfleeger, S. L. *Software Metrics: A rigorous and practical approach*, International Thomson, 1997
7. Kolb, D. A. *Experiential Learning: Experience as the source of learning and development*, Prentice-Hall, 1984
8. McNeill, J. B. Peer Appraisal of Group Projects, *Proceedings of the 27th Annual Conference of the Southern African Computer Lecturers' Association*, (1997)
9. Pfleeger, S. L. Experimental Design and Analysis in Software Engineering (Parts 1, 2 and 3), *Software Engineering Notes* (Oct 1994, Jan 1995, Apr 1995)
10. Shneiderman, B. *Designing the User Interface — Strategies for effective human-computer interaction*, Addison-Wesley, 1992
11. Singley, M. K. *The Transfer of Cognitive Skills*, Harvard University, 1989
12. Solso, R. L. *An Introduction to Experimental Design in Psychology: A case approach*, Harper and Row, 1984
13. Weil, M. M., and Rosen, L. D. The Psychological Impact of Technology from a Global Perspective, *Computers in Behaviour* 11, 1 (1995)

Endnotes

¹ One particular problem which has arisen several times springs from inadequate software testing. Students will write a program for the experiment and test it with one or two users, but fail to test it with a whole room of networked machines operated by the intended user population. The students are used to writing programs for which they are the only users, and fail to make the experimental software scalable or foolproof. There may be embarrassment when the software fails during the first session of the experiment

and some data may have to be discarded, but this itself provides a learning experience.