

MPhil in Advanced Computer Science

Usability of Programming Languages

Leader: Alan Blackwell

Term: Easter

Prerequisites: Introductory HCI course presumed, but not essential

Structure:

- 4 x 1-hour introductory lectures
- 4 x 2-hour interactive seminars
- Variable number of lab-based experimental sessions, including students' own experiments (individually designed and scheduled in consultation with the course director), as well as participation in experiments conducted by other students.

Aims:

Compiler construction is one of the basic skills of all computer scientists, and thousands of new programming, scripting and customisation languages are created every year. Yet very few of these succeed in the market, or are well regarded by their users. This course addresses the research questions underlying the success of new programmable tools. A programming language is essentially a means of communicating between humans and computers. Traditional computer science research has studied the machine end of the communications link at great length, but there is a shortage of knowledge and research methods for understanding the human end of the link. This course provides practical research skills necessary to make advances in this essential field. The skills acquired will also be valuable for students intending to pursue research in advanced HCI, or designing evaluation studies as a part of their MPhil research project.

Syllabus:

Lecture 1: Principles of human factors in programming. Cognitive models of the programming process, introduction to the leading world research centres and current areas of theoretical attention. Integration of programming within the software development process. Sources of individual variation. Relationship between syntax and tools, including integrated development environments, visual languages, spreadsheets, scripting.

Lecture 2: Research methods in the study of programming. Controlled experimental methods, typical tasks, experimental manipulations of programming tools. Measurement, observation and protocol analysis techniques. Experiment design. Field study methods.

Lecture 3: Special classes of programming language use, including end-user programming, mashup and composition tools, educational languages. Overview of classes of technical approach in novel language development, requirements capture and evaluation for these classes.

Lecture 4: Planning practical empirical studies. Students will identify specific programming languages for investigation, and representative tasks and measures suitable for usability evaluation. Candidate languages can be drawn from current research in the Computer Laboratory or at Microsoft Research Cambridge, recent product launches and prototypes, or languages implemented in project work by MPhil students themselves.

Experiments designed and conducted by students will be presented in seminar sessions, for discussion and feedback among the class. Seminars will also include invited guests, postdoctoral researchers and PhD students carrying out active research in programming language design and evaluation.

Seminar 1 and 2: presentation and feedback on study proposals

Seminar 3 and 4: presentation of research study findings

Objectives:

This is a practical course, in which students will work in stages through the development of research hypotheses and proposals, to the design and implementation of their own empirical study. Necessary theory base and research skills will be developed through practical experimental work, with students reaching a level of competence sufficient for conducting and publishing their own original research in the field.

Coursework

Students will prepare three pieces of written work during the course. Each assignment will be submitted in two iterations - an initial proposal, then a revised version based on feedback in seminars and from the course director.

Assignment A: background to a proposed study, including description of the target language, paradigm, tool or environment, a review of the relevant theoretical literature and previous empirical studies

Assignment B: structure of the experimental design, detailed protocol of the proposed study, and outline of analytic methods to be used.

Assignment C: full experimental report, building on final versions of assignments A and B, and presenting data analysis and findings in a format suitable for publication at a specialist research venue such as the psychology of programming interest group.

Practical work

Practical sessions will make use of the dedicated experimental observation facility in the Gates Building, which provides a range of recording and measurement equipment. Each student will

design and conduct their own experiment, spread over approximately four practical sessions. Students will also be required to participate in at least three experiments conducted by other members of the class.

Assessment

- Assignment A, initial and revised versions, 10% each
- Assignment B, initial and revised versions, 10% each
- Assignment C, initial and revised versions, 20% each
- Contributions made in four seminars, 5% each

Reading List

Online proceedings of the Psychology of Programming Interest Group (<http://www.ppig.org>)

Cambridge guidance for human participants in technology research (<http://bit.ly/https-guide>)

Cairns, P. and Cox, A.L. (2008) Research Methods for Human-Computer Interaction. Cambridge University Press.

Hoc, J.M, Green , T.R.G, Samurcay, R and Gilmore, D.J (Eds.) (1990) Psychology of Programming. Academic Press.

Carroll, J.M. (Ed) (2003). HCI Models, Theories and Frameworks: Toward a multidisciplinary science. Morgan Kaufmann.