Sorry, There Is No Algorithm For That: Orienting Computer Science Students To IDC Issues And Processes

Alyssa M. Alcorn
School of Informatics
University of Edinburgh
10 Crichton Street
Edinburgh EH8 9AB
Scotland
aalcorn@ed.ac.uk

Helen Pain
School of Informatics
University of Edinburgh
10 Crichton Street
Edinburgh EH8 9AB
Scotland
helen@inf.ed.ac.uk

Abstract
Technical members of IDC projects may have little background in design, but can benefit from a basic orientation to the roles, methods and outcomes of design work. Providing computer science (CS) students with opportunities to experience what interaction design for children means in practice helps them bridge the gap between their technical skills and the issues and processes in the IDC field, and may lead to more productive and cooperative future contributions. We discuss how such to provide such opportunities for orienting CS students to IDC.

Author Keywords
Teaching; design; computer science; children; autism

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction
Most projects in the field of *interaction design and children* (IDC) involve multi-member teams representing a range of disciplinary experiences across education, media, technology, and the social sciences. While junior IDC-focused researchers increasingly have trained in the specific *conjunction* of required skills, many IDC community members will continue to enter the field via more "traditional"

Teaching context

Institution: Large department in a large Scottish university, encompassing computer science, AI, robotics, cognitive science, computational linguistics, bioinformatics and other related disciplines.

Programme: Highly structured, little opportunity to take classes in other IDC-relevant disciplines (e.g. social sciences, the arts). The department offers one HCI course, focused on traditional methods for adults.

Students: UG4 project students generally have strong programming and software engineering skills, but limited experience of considering human users. The MSc students are more variable in their background and skills

Final projects: These are individual (rather than collaborative) and last one academic year for UG4 students (worth 1/3 of the year's credit), and over the summer for MSc students.

routes, with little knowledge of design processes and their contributions. This is particularly true of those hired with the expectation that they will focus on technical implementation. When considering the education of future IDC community members, it is important to consider those with specialised, relevant skills who are *not* in design programmes, but who are interested in the interdisciplinary nature of IDC and would like to contribute to this area.

We outline some of our experiences with teaching Computer Science (CS) students how to transition into the world of IDC, orienting them to major concerns and methods in this area. This has taken the form of supervising IDC-related final projects at the fourth-year undergraduate (UG4) and taught masters (MSc) level. Through trial-and-error over the past four years, we have learned that a central task of project supervision is orienting students to "the very idea" of design, and how it fits in with technical development. Our goal is not to transform our CS students into fully independent interaction designers. Instead, we aim to supplement their technical skills with hands-on design experience of the design process, and to foster an understanding of the types of knowledge that common design methods can—and cannot—produce.

We believe that our experiences may be useful to others researching and teaching IDC topics *outside* of specialised design programmes. While we refer to CS students, many of our suggestions may be applicable to wherever students need to "bridge" from their existing disciplinary skills and viewpoints into the more multidisciplinary, design-intensive world of IDC. Through this workshop, we hope to identify areas for improvement in terms of the concepts and ways of thinking to which we should be orienting our students.

IDC-related projects for computer science

Despite a current lack of emphasis on *Human Computer Interaction* (HCI) in our department, there is a strong student appetite for IDC project topics. The authors' proposed UG4/MSc student projects in this area are heavily oversubscribed. Most of our project proposals focus on technology for young children with autism (our current research area), and students' first task is to select a narrower, theoretically-motivated issue within this broad area (e.g. support for emotional regulation). Projects are at the proof-of-concept scale, usually building an app or program with one sophisticated activity, or several linked, simpler ones. Some student projects have explored extensions of current technologies for autism (e.g. [1]).

Prospective project students are enthusiastic about the idea of designing technology for children with autism but do not have a clear sense of what this entails—even when our proposals list explicit design tasks (e.g. design critique with experts) or emphasize that design will be THE major portion of project work [2]. An IDC-related project is *completely different* than what they have done before. To CS students, design may not appear to be "real work", or may seem like something they could "make up" off the top of the head. We must persuade them that design is more than a brief planning stage prior to writing code—that the main project focus (and credit-earning component) is design process, rather than designed product. This requires a shift from an implementation-oriented mentality to a research mentality of identifying problems and producing knowledge. It is a bigger shift to understand and apply design methodologies as a route to producing that knowledge.

Design: The very idea

While each student and project is different, our general process of orienting CS students to the idea of IDC involves addressing several "families" of concepts: 'design as a process'; 'design as particular, situated, and negotiable', and 'design as rooted in theory and practice'. These three design concept families help to illuminate a fourth family: that of technologies as *tools* to answer research questions. Table 1 summarises these families and key points we try to communicate about each. This last concept family has posed a surprisingly persistent teaching challenge. It has been very difficult to help students re-frame their thinking about project contributions in terms of what has been *learned* during the design and evaluation of the technology, rather than in terms of the technology as a self-justifying product. Teaching students to take this process-oriented mindset is another topic which we would be particularly interested to discuss during the workshop.

It makes little sense to teach the concept families individually, due to their interconnection. We use several of the same design activities each year to help students grasp these ideas by gradually using them. These activities and their intended lessons are summarised in Table 2.

Concept family	**Key points for students**
Design as a process	-Design is an *ongoing process*, not a one-off planning step that happens only to enable programming -The process is iterative, with multiple steps of designing and re-building. In this process, initial or favourite ideas may need to be "sacrificed" as the design and requirements evolve. -Overall, this process is non-linear and there is no algorithm to produce a good design or indicate when something is "finished".
Design as particular, situated, and negotiable	-It does not make sense to build or evaluate a design in a vacuum. We cannot ask "is it good?" without also asking "for whom, and for what, and in what context(s)?" -The same design may have different purposes, meanings, and effects in different contexts. -At every stage of the process, there is interpretation and negotiation about meanings and priorities. -There is no algorithm for design: there are no objectively "right" answers or agreed stopping rule
Design as rooted in theory and practice	-Consulting research literature and current practices can help us to determine where IDC can make novel and useful contributions. -Literature can illuminate the characteristics and needs of a participant group, and later provide evidence for justifying design features and decisions. -Theory and practice will not give us exact, ready-to-build specifications. -Designing for the autism community *demands* that work is clearly grounded in, and justified by, the literature. -Existing educational and behavioural strategies for children with autism cannot be directly implemented as technologies. Principles must be re-interpreted for a new medium, taking advantage of its affordances
Technolog-ies as tools to answer research questions	-Technology design is only part of research, it can be "successful" and novel without technical innovation. -The technology is primarily *a means to an end,* which is to answer research questions. -The project's main contribution is not the technology as product or artefact, but the knowledge produced by designing and/or using the technology. -In many cases, the contribution includes *proof-of-concept* about the viability of some approach.

Table 1. Concept families for orienting computer science students to IDC work

Summary and conclusions

Technical members of the IDC community who have a good overview of the design process can appreciate its contribution to creating technologies that are useful and meaningful. They will understand the types of knowledge that design activities can and *cannot* produce, most crucially that *they will not produce detailed, ready-to-build specifications.* The outputs of design activities are input for another stage of interpretation, negotiation, and decision-making before building begins (or continues). The process of design is not, in general, arbitrary or "of the top of the head", but grounded in the literature and highly sensitive to context. Even limited design experience and training can be extremely beneficial for technical members of the IDC community, and thus the community as a whole. In our experience, orienting technical members of IDC projects to

Design activity	Project phase(s)	Lessons learned and skills practiced
"Low-pressure" design critique: Meeting with supervisor(s) and students on related projects for informal discussion of design-in-progress.	Formative evaluation approximately monthly throughout project, from initial ideas to evaluation results	-Introduction to critique as a method -Repeated sessions (gently) push students to engage with design *as a process,* to explore alternatives, and to practice justification -Design seems less monolithic and more manageable because sub-problems are naturally identified in the course of discussion, highlighting where additional work needed (e.g. consult literature)
Design critique with multiple experts: 20-40 minute session with at least two experts from different disciplines (e.g. autism, psychology, education, assistive technology)	At multiple project stages: paper prototypes, first full version of technology	-Students must concisely present their design as situated in a particular context, consider how a design functions in a specific context, and what its (potentially many) meanings and effects may be -Exposure to plurality of views and opinions about what to design, how best to do it, and what constitutes sufficient and appropriate justification -Post-critique, students must negotiate way forward between conflicting recommendations, in light of time and resource limitations
Semi-structured interviews with expert evaluators: 20-40 minute sessions focusing on how the technology may be used in practice, its possible contributions.	Summative evaluation; may be used as a substitute where evaluation with children not feasible	-Introduce concept of participation by proxy, what we may learn from it, and its pros/cons in comparison with "real" participant evaluation -Explore the types of expertise/experts relevant to IDC, which includes parents and teachers in addition to academic expertise -Students are forced to move beyond familiar function and usability questions to gather information on their *research* questions
Evaluation with typically developing children: One session of play with the technology, usually conducted in the lab. Usually supplemented by short discussion with child.	Summative evaluation; less frequently used due to difficulty in reconciling project timeline with child availability	-Concretises previously abstract information about children's attention spans, comprehension of instructions, motor skills -Children's perception, use of technology may differ from what designer planned or anticipated -The data is not statistically analysable; analysis must be qualitative -Students must consider what claims can be made based on limited information, and how to explain project value as proof-of-concept.

Table 2. Learning and skills practice in project design activities

the concerns and methods of this area vastly increases the likelihood that their future work with designers will be cooperative and productive, not fraught with tension due to misunderstandings about the role and outcomes of design work. We strongly encourage that those experienced teachers and researchers helping computer scientists (or others in non-design disciplines) "transition" to the IDC area to spend time orienting them to "the very idea" of design, and its contributions to research and development.

References

[1] Fletcher-Watson, S. Hammond, S., Humphrey, A., Pain, H & McConachie, H. (submitted). Designing an iPad app to teach social attention skills to very young children with autism spectrum disorder. *J. of Autism and Developmental Disorders, under review*

[2] Scaife, M. and Rogers, Y. (2001). Informing the design of a virtual environment to support learning in children. *Int. J. of Human-Computer Studies*,55(2):115–143