

Ethnographies of Code workshop

Position Paper: Values in Design and politics and programming made ordinary
Daniel Ashton, Institute for Cultural Research, Lancaster University

Abstract

This position paper seeks to outline concepts of values in design and how they may be usefully employed in everyday contexts of programming. The paper will first introduce significant contributions and works by Nissenbaum on values in design and Taylor on reverse engineering. Drawing on the peekabooby project, this paper will outline how particular values in design may be introduced and embodied within a given system. In translating this possibility of values in design, this paper moves to consider how everyday values in design can give rise to and open up questions of politics in everyday programming. Drawing on two difficulties highlighted by Nissenbaum for taking values into consideration during design, the aim is to articulate how questions of everyday politics may be brought to everyday programming.

Paper

Coming from a broadly Cultural Studies tradition with a specific interest in activist models of politics¹, I am interested in the relationship between programming and code in terms of ‘values in design’ (Nissenbaum and Introna, 2000; Nissenbaum, 2001, 2005) and ‘reverse engineering’ (Taylor, 2005). Nissenbaum suggests

Values affect the shape of technologies. Briefly, the values that systems and devices embody are not simply a function of their objective shapes. We must also study the complex interplay between the system or device, those who built it, what they had in mind, its conditions of use, and the natural, cultural, social and political context in which it is embedded
(2001: 117)

In specifically seeking to trace teleological specificities of building and designing a system and the realisation of that system, my main interest is in the possibilities for a politicisation of systems and code through values in design. Both Nissenbaum and Taylor discuss the radical possibilities bound up with computer engineering. For Nissenbaum, perceiving not only the usual set of properties that engineers build or design into a system but also moral properties means, “the challenge of building computer systems is transformed into a forum for activism – engineering activism” (2001: 119). Taylor considers engineering activism in terms of distinction between hackers and hacktivism. Taylor uses “‘hacktivism’ – the combination of hacking techniques with political activism – to explore this perennial tension between human agency and technological structures” and suggests with hacktivism “a normative element has been put back into objectified computer code” (2005: 626). Taylor’s key point follows

¹ For example, movements networked and linked through the Global Justice Movement (Alternative Globalisation Movement) in response to current neo-liberal policies.

Despite its affinity with the manipulation of technical systems, hacking is shown to have a fundamental blind spot in its engagement with the overarching system within which those systems are contained ... Hacktivism engages with this metasystem directly and, with its close ties to the politics of globalisation, marks the beginning of a significant new chapter in radical political technologies
(2005: 627)

Furthermore, for Taylor “the tendency to reverse engineer technology to do the opposite of its intended design” (2005: 628) is a core element of the early hacking ethic (outlined by Levy, 1984) which offers the basis for “the strategies to use technology to promote rather than hinder human agency” (2005: 628).

In taking the lead from Taylor and Nissenbaum and speaking to the interests and goals of the workshop, my concern is with possibilities for dialogues across activism and computing boundaries. This is not to argue for a ‘radicalising’ of computing in terms of a particular politics, for example the Alter Globalisation Movement, or to deny the mutability of such boundaries or, indeed, overlook current projects and contexts which question the validity of such boundaries. My concern specifically is a consideration of how values in design are embodied within particular contexts and what may be learnt from these contexts. For example the goal of the Peekabooty Project, as described on their website², is to “create a product that can bypass the nation-wide censorship of the World Wide Web practiced by many countries”. Peekabooty is a distributed or peer-to-peer application and taking “minimum network knowledge”, a requirement for Internet censorship circumvention systems, as a central consideration a list of resulting design requirements were drawn up. In this case, a particular goal of circumventing web censorship led to very specific design requirements and the values which the Peekabooty application may be seen to embody are those of privacy and anonymity. In this specific context the relationship between values in design and embodiment may be relatively easily identified and traced and to an extent provides a model, which may be considered alongside numerous others, for thinking through values in design.

In outlining this position and exploring a specific case study, a significant route of enquiry which follows is considering the practice of embodying values in design and the extent to which this recognising and activist engineering may become a routine or everyday consideration. Nissenbaum states two sets of factors which contribute to the difficulty of taking values into consideration (2005). The first, identified as epistemological in origin, concerns incorporating “diverse and frequently far-flung areas of knowledge and know-how into the design process that are not normally conceived as elements of the design toolkit”; and the second, broadly schematised as practical, stems from “the scarcity of explicit guidelines, or methodologies, for reliably embodying values in software systems (and in technology, generally)” (2005: 2). In considering the first set of factors in relation to everyday practices, a vital issue to address is how everyday politics, concerns and settings relate to practices of designs. For example, how specific and situated areas of knowledge and know-how may be incorporated as values within the design of programmes. Thus in contrast to particular values and corresponding designs, such as censorship and peekabooty, how

² Peekabooty project, <http://www.peek-a-booty.org/pbhtml/index.php> [accessed 16.02.06]

do wider lived and everyday issues translate into values in design. Further questions which then arise relate to how these values then correspond with 'normal' considerations such as functionality. In relation to the second set of factors and scarcity of guidelines, Nissenbaum states the hope that "over time, a library of analyses would be developed in service of the technical design context" and "ideally, this would relieve the burden on designers, allowing them to draw on concrete definitions of past work rather than having to grapple directly with abstract conceptions of key values" (2005: 8). A key point to work with here would be, again in considering the mundane and everyday, how can apparently trivial and assumed considerations usefully contribute to such a library of analyses? In outlining a specific case study in which an identifiable agenda may be seen as translated into design practice, the aim is to generate conversation and discussion around the everyday as a site for identifying agendas and bringing these to the design stage. In working through questions around everyday politics and everyday values in design and programming, a move to consider 'reverse engineering' as outlined by Taylor would be fruitful for imagining the political possibilities of everyday politics reversed engineered into technologies and systems.

In drawing on Certeau (1984) and other efforts within Cultural Studies to attend to the everyday, an important task emerges as to how to question and investigate the relationship between everyday politics and everyday programming. In drawing on values in designs and considering activist engineering as a model, my aim is to highlight the possibilities for everyday programming as a political activity and, in asserting and emphasising the potential for politics in everyday design, suggest how programmers in everyday contexts can relate to 'metasystems' and broad political concerns. In addressing these questions the emphasis could usefully be weighed across questions of everyday politics and what this may mean and questions of the practical nature of embodying values and reverse engineering.

Bibliography

Certeau, M. (1988[1974]) The Practice of Everyday Life. Berkeley: University of California Press

Introna, L. and Nissenbaum, H. (2000) 'The Public Good Vision of the Internet and the Politics of Search Engines' in Rogers, R. (ed) Preferred Placement: Knowledge Politics on the Web Maastricht : Jan van Eyck Akademie. pp. 25-47

Levy, S. (1984) Hackers: Heroes of the Computer Revolution. New York: Bantam Doubleday Bell

Nissenbaum, H. (2001) 'How Computer Systems Embody Values' in The Computer

Nissenbaum, H., Flanagan, M. and Howe, D. (2005) Embodying Values in Technology: Theory and Practice

Available at <http://epl.scu.edu/~stsvvalues/readings/Nissenbaum-VID.4-25.pdf>
[accessed 16.02.06]

Taylor, P.A. (2005) 'From hackers to hacktivists: speed bumps on the global highway' in New Media and Society 7(5) London: SAGE Publications pp. 625-646