

## **Design by Bot: Power and Resistance in the Development of Automated Software Agents**

**R. Stuart Geiger**

UC-Berkeley School of Information  
United States  
stuart@stuartgeiger.com

### **Abstract**

In this paper, I discuss how Application Programming Interfaces (APIs) have enabled new modes of software development that complicates traditional distinctions between developers and users. Coders can build bots, scripts, scrapers, extensions, aggregators, and other tools that change how software applications, platforms, and protocols operate – all without requiring privileged access to software codebases. In Wikipedia, user-authored bots and tools perform a staggering amount of the work required to keep the collaborative encyclopedia project operating in the manner that it does. Bots remove vandalism and spam, alert administrators to conflicts, harmonize linguistic standards, and enforce discursive norms. In reddit, bots have recently emerged to provide new functionalities to the news aggregation and discussion site. I report from an ethnographic study of bot development and bot developers in Wikipedia and reddit, demonstrating the various ways in which the rise of automated software agents has enabled new forms of both power and resistance.

### **Keywords**

Bots, software agents, algorithms, ethnography, values in design

### **Body**

Lawrence Lessig famously declared that “code is law” (1999), arguing that developers have immense power when constructing software platforms and protocols. Those in the “values in design” movement (c.f. Knobel & Bowker 2011, Friedman and Nissenbaum 2004) have demonstrated the many different ways in which power relations can be produced through code. From YouTube’s anti-copyright infringement algorithms (Gillespie, 2010) to Facebook’s formalized ontology of romantic relationship statuses (Brubaker & Hayes, 2011), there is much attention on how code operates as infrastructure. However, unless users are explicitly brought into the design of software, code is seen as written from above and embedded into the lives of users. For the people who use these systems, the only choices seem to be to accept the platform, contest it through discourse, or, as has emerged in the area of participatory design, program an alternative platform that operates according to a different codebase. In this paper, I document an emergent mode of power based on the development of autonomous software agents, or bots.

Most existing studies of bots have been based on a functionalist approach to code, showing how software agents are key social actors in enforcing behavioral and epistemological norms. Tracing the banning of a malicious vandal in Wikipedia, Geiger and Ribes (2010) showed how tools and bots structure administrative workflows such that ad-hoc teams can quickly identify and block contributors. Recent research on Wikipedia’s editor decline have also asserted that quality control bots are having negative effects on socialization in Wikipedia (Halfaker et al, 2013). However, bot development itself is a relatively unstudied sociotechnical phenomenon, although studies of bots in IRC (Latzko-Toth, 2000) have shown how the notoriously thin and featureless chat protocol was extended through the development of a diverse set of automated bots. Most notably, these software agents were delegated key administrative privileges and responsibilities in IRC channels when their operators were away, producing new relations of power and authority without changing the protocol.

The primary empirical goal of this paper is to investigate the ways in which bot development comes to constitute social and technical relations in collaborative communities. To this end, I am ethnographically investigating how bot development is situated within various online communities, specifically Wikipedia and reddit. This work is based on participant-observation in both bot development itself and dispute resolution, semi-structured interviews with bot developers, and archival analysis of important

controversies. Theoretically, I relate these social sub-worlds (Strauss, 1982) of bot development, relating the emic perspective (Morey & Luthans, 1984) of what it means to be a bot developer. I use Bowker and Star's notion of "infrastructural inversion" (2000) to examine how user-generated infrastructures enable new modes of power and new modes of resistance.

This work extends the longstanding body of research into socio-cultural aspects of software development and software developers (e.g. Kelty, 2008; Crowston & Howison, 2005) as well as ethnographies of robotics as a socio-technical field (Vertesi, 2012). My guiding questions on this topic include issues such as: Who are bot developers and why do they build these automated software agents? How do bot developers conceptualize themselves and their work in relation to that of others in and around their community? What kinds of social relationships do bot developers have with "official" developers and "ordinary" users? I situate answers to these questions in the context of controversies surrounding bot developers, like Gillespie (forthcoming) has investigated with search algorithms. Given the way in which bot developers see themselves, how are the inevitable issues surrounding bot development articulated and resolved?

Furthermore, in line with the growing body of work on the socio-technical nature of interaction, this study also investigates the relations bot developers have with technical actors. This work particularly draws on the work of Ed Hutchins on distributed cognition (1996) as well as Suchman's reconfigurations of human-machine interactions (2007), exploring how bot development reshapes our notion of who – and what – participates in the online systems we inhabit. Questions on this topic include: What kinds of social and technical infrastructures do bot developers rely upon to do their work? How have changes in these infrastructures – such as altering an API's functionality, establishing rules for bots, or inviting a bot developer into the server-side codebase – alter the bot development process? Finally, I believe it is critical to investigate how bot developers relate to and experience their own bots. Is there a tension between bots as software maintained as features for the community and bots as extensions of the developer's own self.

Based on the first stages of my ethnographic fieldwork with bot developers, I have found that the ability to develop for a given software platform is not simply a factor of programming skill, but also is affected by a wide variety of factors, most notably whether a codebase is open source or not. Yet even in OSS platforms, some communities are less open to new members and new code than others, and to add a new feature to the live site, a developer may have to enroll a wide array of allies. Platforms are also typically written in a particular language and design approach, requiring that a developer gain familiarity with these potentially foreign modes of software development. Finally, some platforms are built to be modular and extensible, but many more are quite complicated assemblages that require substantial effort to extend and modify.

Bots seemingly do away with many of these barriers to software development. A developer may have to request access to an API or get authorization to run a bot, but this barrier is often far lower than getting a patch merged, much less full commit access. Bots can be written in most modern programming languages, and for specific platforms (like MediaWiki), frameworks have been developed in dozens of languages to make it easy for a novice developer to write their own bot in their favorite language. Bots do not require or allow the developer to interact with a platform's code, and as such, bot developers can treat the underlying software platform as a black box. However, bots raise new issues in the development of software, as bot developers cannot rely on the same regimes of sovereignty that make a feature embedded into a platform instantly implementable. Bot developers struggle with new issues of legitimation and negotiation, which I explore through controversy studies. Like Foucault argues with all forms of power relations (1976), I argue that bot development produces new forms of domination alongside new forms of resistance.

I conclude by discussing the implications that this new mode of software development has for how we understand values and design. I ultimately concur with both Lessig and the values in design movement in that the design of computational systems has profound implications for how social interaction and collaboration is constituted and made possible. However, I expand the location in which software design and development is typically understood take place. Beyond the specific case studies I investigate, I advocate that we should broaden our traditional understandings of where and how design takes place. To this end, I contextualize bot development as a case in which values are not embedded in design, but produced as a result of a decentered, distributed design process that situates users as designers.

## References

- Brubaker, J. R., & Hayes, G. R. (2011). SELECT \* FROM USER. *Proceedings of the ACM 2011 conference on Computer supported cooperative work (CSCW '11)*. doi:10.1145/1958824.1958881
- Bowker, J. & Star, S.L. (2000). *Sorting Things Out: Classification and Its Consequences*. Cambridge, Mass.: The MIT Press.
- Crowston, K., & Howison, J. (2005). The social structure of free and open source software development. *First Monday*, 10(2), 1–21. doi:10.1.1.89.3506
- Foucault, M. (1978) *The History of Sexuality Volume 1: An Introduction*. R. Hurley (trans.). London: Allen Lane.
- Friedman, B. & Nissenbaum, H. "Bias in Computer Systems." in B. Friedman (ed.). *Human Values and the Design of Computer Systems*. Cambridge: Cambridge University Press.
- Halfaker, A., Geiger, R.S., Morgan, J.T., & Riedl, J. (2013) The Rise and Decline of an Open Collaboration System: How Wikipedia's Reaction to Popularity Is Causing Its Decline. *American Behavioral Scientist*. doi:10.1177/0002764212469365
- Geiger, R. S., & Ribes, D. (2010). The work of sustaining order in wikipedia: the banning of a vandal. *Proc CSCW 2010*. New York: ACM Press. doi:10.1145/1718918.1718941
- Gillespie, T. (2010). The Politics of "Platforms". *New Media & Society*, 12(3). doi:10.1177/1461444809342738
- Gillespie, T. (forthcoming). The Relevance of Algorithms. in T. Gillespie, P. Boczkowski, & K. Foot (eds.). *Media Technologies*. Cambridge, MA: MIT Press. Retrieved from <http://www.tarletongillespie.org/essays/Gillespie%20-%20The%20Relevance%20of%20Algorithms.pdf>
- Hutchins, E. (1996). *Cognition in the Wild*. Cambridge, Mass.: The MIT Press.
- Kelty, C. (2008). *Two Bits: The Cultural Significance of Free Software*. Durham: Duke University Press.
- Knobel, C., & Bowker, G. C. (2011). Values in design. *Communications of the ACM*, 54(7), 26. doi:10.1145/1965724.1965735
- Latzko-Toth, G. (2000). L'Internet Relay Chat : un cas exemplaire de dispositif sociotechnique. *COMMposite, v2000.1*. Retrieved from <http://www.composite.org/index.php/revue/article/viewArticle/91>
- Lessig, L. (1999). *Code and Other Laws of Cyberspace*. New York: Basic Books.
- Morey, N. C., & Luthans, F. (1984). An Emic Perspective and Ethnoscience Methods for Organizational Research. *Academy of Management Review*, 9(1), 27–36. doi:10.2307/258229
- Strauss, A. (1982). Social worlds and legitimation processes. *Studies in Symbolic Interaction*, 4, 171–190.
- Suchman, L. (2007). *Human-Machine Reconfigurations: Plans and Situated Actions* (p. 314). Cambridge: Cambridge University Press.

Selected Papers of Internet Research 14.0, 2013: Denver, USA

Vertesi, J. (2012). Seeing like a Rover: Visualization, embodiment, and interaction on the Mars Exploration Rover Mission. *Social Studies of Science*, 42(3), 393–414.  
doi:10.1177/0306312712444645

### **License**

This article is ©2013 Authors, and licensed under CC-BY-SA 3.0.