

# UC Berkeley

## UC Berkeley Electronic Theses and Dissertations

### Title

robots.txt: An Ethnographic Investigation of Automated Software Agents in User-Generated Content Platforms

### Permalink

<https://escholarship.org/uc/item/1js7224r>

### Author

Geiger II, Richard Stuart

### Publication Date

2015

Peer reviewed|Thesis/dissertation

robots.txt: An Ethnographic Investigation of Automated Software Agents in User-Generated  
Content Platforms

By

Richard Stuart Geiger II

A dissertation submitted in partial satisfaction of the  
requirements for the degree of

Doctor of Philosophy

In

Information Management and Systems

and the Designated Emphasis

in

New Media

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Jenna Burrell, Chair

Professor Coye Cheshire

Professor Paul Duguid

Professor Eric Paulos

Fall 2015

robots.txt: An Ethnographic Investigation of Automated Software Agents in User-Generated  
Content Platforms

© 2015

by Richard Stuart Geiger II

Freely licensed under the Creative Commons Attribution-ShareAlike 4.0 License

(License text at <https://creativecommons.org/licenses/by-sa/4.0/>)

Abstract

robots.txt: An Ethnographic Investigation of Automated  
Software Agents in User-Generated Content Platforms

by

Richard Stuart Geiger II

Doctor of Philosophy in Information Management and Systems

with a Designated Emphasis in New Media

University of California, Berkeley

Professor Jenna Burrell, Chair

This dissertation investigates the roles of automated software agents in two user-generated content platforms: Wikipedia and Twitter. I analyze ‘bots’ as an emergent form of sociotechnical governance, raising many issues about how code intersects with community. My research took an ethnographic approach to understanding how participation and governance operates in these two sites, including participant-observation in everyday use of the sites and in developing ‘bots’ that were delegated work. I also took a historical and case studies approach, exploring the development of bots in Wikipedia and Twitter. This dissertation represents an approach I term *algorithms-in-the-making*, which extends the lessons of scholars in the field of science and technology studies to this novel domain. Instead of just focusing on the impacts and effects of automated software agents, I look at how they are designed, developed, and deployed – much in the same way that ethnographers and historians of science tackle the construction of scientific facts. In this view, algorithmic agents come on the scene as ways for people to individually and collectively articulate the kind of governance structures they want to see in these platforms. Each bot that is delegated governance work stands in for a wide set of assumptions and practices about what Wikipedia or Twitter is and how it ought to operate. I argue that these bots are most important for the activities of collective sensemaking that they facilitate, as developers and non-developers work to articulate a common understanding of what kind of work they want a bot to do. Ultimately, these cases have strong implications and lessons for those who are increasingly concerned with ‘the politics of algorithms,’ as they touch on issues of gatekeeping, socialization, governance, and the construction of community through algorithmic agents.

## Acknowledgements

There are too many people for me to thank in completing this dissertation and my Ph.D. This work is dedicated my personal infrastructure of knowing and knowledge production – the vast assemblage of individuals who I have relied on for guidance and support throughout my time as a graduate student. There could be a whole dissertation of material documenting all the people who have helped me get to this point, but these few pages will have to suffice for now.

The School of Information at UC-Berkeley has given me a wealth of resources to help me get to this moment, but the people have been the greatest part of that. My fellow members of my Ph.D cohort, Galen Panger and Meena Natarajan, have been on the same path with me for years, helping each other through all the twists and turns of this journey. I also want to thank many members of my Ph.D program, who have helped me refine and shape my ideas and myself. I've presented much of the work in this dissertation at both formal colloquia and informal talks over drinks in the Ph.D lounge, so thanks to Andy Brooks, Ashwin Matthew, Christo Sims, Dan Perkel, Daniela Rosner, Elaine Sedenberg, Elizabeth Goodman, Ishita Ghosh, Jen King, Megan Finn, Nick Doty, Nick Merrill, Noura Howell, Richmond Wong, Sarah Van Wart, and Sebastian Benthall for their ideas and support.

The School of Information is also home to three members of my committee, without whom this dissertation could not have been written. I want to thank my chair and advisor Jenna Burrell for her time and energy in supporting me as a Ph.D student, as well as Paul Duguid and Coye Cheshire, for their years of service and support. They have helped me not only with research for this dissertation, but also in various other aspects of becoming an academic. I'd also like to thank Eric Paulos, my outside committee member, for his advising around my dissertation. And there are other professors at Berkeley who have helped shape this work and my thinking through classes and conversations: Greg Niemeyer, Ken Goldberg, Abigail De Kosnik, Cathryn Carson, Massimo Mazzotti, and Geoffrey Nunberg. I'd also like to thank South Hall administrative staff for all of their work in getting me to this point, including: Meg St. John, Lety Sanchez, Roberta Epstein, Siu Yung Wong, Alexis Scott, and Daniel Sebesen, as well as Lara Wolfe at the Berkeley Center for New Media and Stacey Dornton at the Berkeley Institute for Data Science.

I've had the opportunity to collaborate with amazing researchers, who have shaped and expanded my thought. Aaron Halfaker, my long-time collaborator, has been a constant source of support and inspiration, and I know that we have both expanded our ways of thinking in our collective work over the years. I also want to thank Heather Ford, my fellow ethnographer of Wikipedia, for her collaboration, friendship, and support. I'm thankful to met the members of the Wikimedia Summer of Research 2011, who early on in my Ph.D helped me understand the full breath of Wikipedia research: Fabian Kaelin, Giovanni Luca Ciampaglia, Jonathan Morgan, Melanie Kill, Shawn Walker, and Yusuke Matsubara. I also want to thank many people at the Wikimedia Foundation, who have been incredibly helpful throughout this research including Dario Taraborelli, Diederik van Liere, Oliver Keyes, Maryana Pinchuk, Siko Bouterse, Steven

Walling, and Zack Exley. I'd also like to thank the Wikipedians and Twitter blockbot developers who have assisted with this research, both in the time they took to talk with me and in the incredibly important work that they do in these spaces.

I am also grateful to have an abundant wealth of friends and colleagues in academia, who have helped shape both me and my work. Many of these people I met when we were both graduate students, although some are now faculty! I've met many of these people at conferences where I've presented my work or heard them present their work, and our conversations have helped me get to the place where I am today. These people include: Aaron Shaw, Airi Lampinen, Alex Leavitt, Alyson Young, Amanda Menking, Amelia Acker, Amy Johnson, Andrea Wiggins, Andrew Schrock, Anne Helmond, Brian Keegan, Caroline Jack, Casey Fiesler, Charlotte Cabasse, Chris Peterson, Dave Lester, Heather Ford, Jed Brubaker, Jen Schradie, Jodi Schneider, Jonathan Morgan, Jordan Kraemer, Kate Miltner, Katie Derthick, Kevin Driscoll, Lana Swartz, Lilly Irani, Mako Hill, Matt Burton, Melanie Kill, Meryl Alper, Molly Sauter, Nate Tkacz, Nathan Matias, Nick Seaver, Norah Abokhodair, Paloma Checa-Gismero, Rachel Magee, Ryan Milner, Sam Wolley, Shawn Walker, Stacy Blasiola, Stephanie Steinhardt, Tanja Aitamurto, Tim Highfield, Whitney Philips, and Whitney Erin Boesel. I'm sure I'm also leaving some people out of this long list, but this only goes to show you that so much of what I have accomplished is because of all the people with whom I've been able to trade ideas and insights.

There have also been many faculty members who have generously helped me think about my work and my career, as well as given me many opportunities during my time as a graduate student. I want to particularly thank David Ribes, my advisor in my M.A. program at Georgetown, for helping me first dive into these academic fields, and for continuing to help support me. I also want to thank those scholars who have helped me think through my work and my career in academia: Amy Bruckman, Andrea Tapia, Andrea Forte, Andrew Lih, Annette Markham, Chris Kelty, Cliff Lampe, David McDonald, Geert Lovink, Gina Neff, Janet Vertesi, Kate Crawford, Katie Shilton, Loren Terveen, Malte Ziewitz, Mark Graham, Mary Gray, Nancy Baym, Nicole Elison, Paul Dourish, Phil Howard, Sean Goggins, Steve Jackson, Steve Sawyer, T.L. Taylor, Tarleton Gillespie, Zeynep Tufekci, and Zizi Papacharissi.

Institutionally, I'd like to thank many departments, centers, and institutes for supporting me and my work, both at Berkeley and beyond. Locally, I'm thankful for the support of the UC-Berkeley School of Information, the Department of Sociology, the Center for Science, Technology, Medicine, and Society (CSTMS), the Center for New Media, and the Berkeley Institute for Data Science. Further out, I'm thankful for the support of the Center for Media, Data, and Society at Central European University, the Oxford Internet Institute, the MIT Center for Civic Media, the Berkman Center at Harvard Law School, the Consortium for the Science of the Sociotechnical, and the Wikimedia Foundation. People and programs at these institutions have graciously hosted me and other like-minded academics throughout the years, giving an invaluable opportunity for me to grow as a scholar.

Last but certainly not least, I'd like to thank the people in my family, who have supported me for longer than anybody else mentioned so far. My parents, Scott and Kathleen, and my sister Audrey Scott, have given me so much support in getting me to the place where I am today. I also want to thank my Aunt Susie and Uncle Scott, for everything they have done in my many years as a student. They have all supported me throughout the highs and the lows, and I've never doubted their love and care for me. There is no way I'd be in the place where I am right now without them, and that means the world to me.

## Table of Contents

Chapter 1: Introduction.....	1
1. Thematic overview .....	1
1.1 Vignette: “Bots are editors too!” .....	1
1.2. The politics of algorithms and the Soylent Green argument.....	3
2. How to study algorithms: lessons from science and technology studies.....	4
2.1 Studying the production of knowledge ‘in the making’ .....	4
2.2 Algorithms as a mode of knowledge production.....	6
2.3 The black box .....	9
2.4 Methods for studying black boxes.....	10
3. Chapter overview.....	13
Section 1: An algorithms-in-the-making approach (ch 2,3).....	13
Chapter 2: Bespoke Code and the Bot Multiple.....	13
Chapter 3: Exclusion compliance .....	14
Section 2: Bots in administrative spaces in Wikipedia (ch 4, 5, 6).....	15
Chapter 4: Articulation Work.....	15
Chapter 5: Membership and socialization .....	16
Chapter 6: An algorithmic history of Articles for Deletion .....	16
Section 3: Blockbots in Twitter (chapter 7 and 8).....	17
Chapter 7: The blockbot tutorial-ethnography: a composite account .....	17
Chapter 8: Blockbots as projects of collective sensemaking and reconfiguration .....	18
Chapter 9: Conclusion .....	18
Section 1: Algorithms in the Making (chapter 2 and 3).....	19
Chapter 2: Bespoke code and the bot multiple.....	19
1. Introduction .....	19
1.1 Defining and contextualizing bespoke code.....	19
1.2 Why study bespoke code? .....	20
1.3 IRC bots and the delegation of governance work .....	21
2. Bespoke code in Wikipedia.....	22
2.1 Infrastructural inversions.....	22
2.2 Bots in Wikipedia .....	26
3. Theoretical issues around studying bespoke code.....	28



3.3 Bespoke code as a way to call attention to the materiality of software.....	31
3.4 Code as law-in-the-making.....	32
4. About a bot .....	37
4.2 A legitimate alternative account.....	39
4.3: Moving out and away .....	41
4.4: Code/coda.....	43
5. Conclusion.....	45
Chapter 3: Exclusion compliance.....	47
1. Introduction .....	47
1.1 Chapter overview.....	47
1.2 What is the <code>antibot_template_protected_text</code> function?.....	48
1.3 Theorizing delegation in organizations .....	51
2: A history of exclusion compliance in Wikipedia.....	53
2.1: A problem and a solution .....	53
2.2: A problem with the solution.....	55
2.3: An Unexpected Ally.....	59
2.4: The formalization of exclusion compliance .....	60
2.5 The banning of Betacommand.....	62
3. Conclusion.....	63
Section 2: Specialized administrative processes in Wikipedia (Chapter 4, 5, and 6) .....	66
1. Summary of section 1 (chapter 2 & 3) .....	66
2. Overview of section 2.....	68
Chapter 4: Articulation work.....	70
1. Introduction .....	70
1.1 Chapter overview.....	70
2. Bots behind the scenes in Wikipedia.....	72
2.1 The edit request process .....	72
2.2: Previous literature and popular commentary on Wikipedia behind the scenes.....	76
2.3. Quantifying behind the scenes coordination work by Wikipedians .....	80
3. Discussion: bots as articulation work.....	83
3.1 Bots as “clerks” for a process, rather than independent decision-makers .....	83
3.2: The bureaucracy of crowds .....	85
3.3: The concept of articulation work.....	86

4. Conclusion.....	88
Chapter 5: Socialization and Membership in Administrative processes .....	90
1. Introduction .....	90
1.1 Chapter overview.....	90
1.2: Introductory vignette: “Save saveMLAK!” .....	91
2: Specialization and socialization .....	94
2.1 Articulating trials of strength.....	94
2.2 Specialization and socialization .....	95
2.3. Generalization and organizational literacy .....	99
3: Discussion: The encyclopedia that <i>anyone</i> can edit? .....	101
3.1 Overview .....	101
3.2 Wikipedia and its discontents.....	102
3.3 Organizational literacy .....	102
4. The (Tea)house that bots built.....	103
4.1. The Teahouse.....	103
4.2 Discussion.....	105
5. Conclusion.....	106
Chapter 6. An algorithmic history of Articles for Deletion.....	108
1. Chapter Summary .....	108
2. A History of Deletion .....	110
2.1: The problem of server-sovereigns: Wikipedia’s early years (2001-2002).....	110
2.2: New process, new software (2002-2003).....	112
2.3: The bots enter the process (2004-5) .....	115
3. Conclusion.....	121
Section 3: Blockbots.....	123
1. Lessons learned from Wikipedia .....	123
2. Twitter’s blockbots as a new case to apply the algorithms-in-the-making approach.....	124
2.1 Introducing harassment in Twitter and various responses .....	124
2.2 Contrasting blockbots with malicious bots.....	125
2.3 Contrasting blockbots with other automated software agents in Twitter .....	126
3. Overview of the two chapters.....	127
Chapter 7: The blockbot tutorial-ethnography: a composite account .....	129
1. Introduction .....	129

1.1 A composite account .....	129
1.2 The genre of the tutorial .....	129
2. The tutorial-ethnography .....	131
2.0: blocking on Twitter’s interface without a bot .....	131
2.1: the API, queried by a software program.....	131
2.2: Synchronizing blocklists .....	133
2.3: oAuth .....	135
2.4: coordinating servers: blocktogether.org .....	137
2.5: Deployment .....	140
2.6: single-purpose accounts .....	141
2.7: encapsulation of the organization.....	142
3. Conclusion.....	147
Chapter 8: Blockbots as projects of collective sensemaking and reconfiguration .....	148
1. Introduction .....	148
1.1 Classification and its consequences: beyond “What <i>really</i> is harassment?” .....	148
1.2 Blockbots are embedded in and emerge from counterpublic communities .....	149
2. Literature review: harassment, technological determinism, and networked publics.....	150
3. How blockbots help counterpublic groups moderate their own experiences .....	154
4. Blockbots are embedded in and emerge from counterpublic communities .....	158
5. Discussion.....	160
6. Conclusion.....	163
Chapter 9. Conclusion .....	166
2. Themes .....	167
2.1 proceduralization .....	167
2.2 The regulation of bots.....	168
2.3 Bots as speculative projects.....	169
3. Infrastructures and ideologies of user-generated content platforms.....	170
3.1 The rise of user-generated content .....	170
3.2 Discourses of libertarian governance .....	171
3.3 Who does the work of sustaining order? .....	173
3.4 Where is a platform? .....	174
3.5 A delegation and a re-distribution of governance work .....	174
10. Works cited.....	176

# robots.txt: An Ethnographic Investigation of Automated Software Agents in User-Generated Content Platforms

## Chapter 1: Introduction

### 1. Thematic overview

#### 1.1 Vignette: “Bots are editors too!”

In late 2006, members of the English-language version of Wikipedia<sup>1</sup> began preparing for their third annual election for Wikipedia’s Arbitration Committee -- or ArbCom, for short. According to the official Wikipedia policy defining arbitration and ArbCom’s role at the time, the fifteen member committee “has the authority to impose binding solutions to disputes between editors, primarily for serious conduct disputes that have previously proven unresolvable through all other dispute resolution practices and processes.”<sup>2</sup> As ArbCom is tasked with making controversial decisions when there is no clear consensus on a given issue, arbitrators hold some of the most powerful positions of authority in the encyclopedia project.<sup>3</sup> The decisions reached in such cases not only make front page news in the *Wikipedia Signpost*, the project’s weekly newspaper reporting on internal issues. Such cases also occasionally make headlines in mass media, such as ArbCom’s ruling that banned members of the Church of Scientology from editing<sup>4</sup> and a controversial ruling about the article on the GamerGate controversy.<sup>5</sup> ArbCom is often referred to as Wikipedia’s high or supreme court,<sup>6</sup> and it should be no surprise that elections for the few seats that open each year are hotly contested. In this particular election, nominations for open seats were accepted during November 2006.

---

<sup>1</sup> When I discuss Wikipedia in this dissertation, I focus on the English-language version. Different language versions of Wikipedia have a strong degree of autonomy when it comes to setting and enforcing internal norms and processes about how articles are to appear and how editors are to interact. Different language versions of Wikipedia have similarities and differences, but a comparative study is outside the scope of this dissertation.

<sup>2</sup> [https://en.wikipedia.org/w/index.php?title=Wikipedia:Arbitration\\_Committee&oldid=664663537](https://en.wikipedia.org/w/index.php?title=Wikipedia:Arbitration_Committee&oldid=664663537)

<sup>3</sup> Throughout this work, I use many emic terms common among highly-active Wikipedia contributors. I use the emic term “the project” to refer to a broad notion of the English-language Wikipedia as a collective endeavor to author and curate an encyclopedic text, which takes place in a variety of mediated and face-to-face settings and involves a wide variety of work.

<sup>4</sup> See <http://www.telegraph.co.uk/technology/wikipedia/5408761/Church-of-Scientology-members-banned-from-editing-Wikipedia.html>

<sup>5</sup> See <https://www.washingtonpost.com/news/the-intersect/wp/2015/01/29/gamergate-wikipedia-and-the-limits-of-human-knowledge/>

<sup>6</sup> Although arbitrators continually emphasize that there are differences between the committee and courts of law.

According to the newly-established rules for ArbCom elections, all editors who had made at least 1,000 edits to the encyclopedia project as of October of that year were eligible to run.

In all, about forty Wikipedians<sup>7</sup> meeting these requirements nominated themselves or accepted the nominations of others, which formally involved submitting a brief statement to potential voters with reasons why they would be good arbitrators. One of these nominations came from an account with the username AntiVandalBot, which was used by an automated software agent that reviewed all edits to the project as they were made in near real-time, then reverted those that it determined were blatant acts of vandalism or spam, according to various algorithms. This bot was developed and operated by a well-known Wikipedia administrator<sup>8</sup> named Tawker, who, in a common convention, used the separate user accounts to distinguish between edits he personally made and those authored by the automated software agent. AntiVandalBot's statement to voters drew on many tropes common in Wikipedian internal politics, including a satirical description of its accomplishments and adherence to project norms (like Neutral Point of View, or NPOV) in the same rhetorical style as many other candidates:<sup>9</sup>

I always express NPOV on any decision I make because I have no intelligence, I am only lines of code. I also never tire, I work 24 hours a day, 7 days a week. I think I have the most of edits of any account on this Wiki now, I have not counted since the toolserver database died. Taking a look at my talk page history, my overseers ensure that all concerns are promptly responded to. In short, a bot like me who can function as a [Magic 8 Ball](#) is exactly what we need on ArbCom! -- [AntiVandalBot](#) 05:20, 17 November 2006 (UTC)

While some Wikipedians treated the bot with at least an ironic level of seriousness, others expressed frustration at Tawker, who denied he was acting through his bot and insinuated it had become self-aware. One editor removed the bot's candidate statement from the nomination page without prior discussion, but Tawker had AntiVandalBot quickly revert this removal of content as an act of vandalism. Another editor deleted the statement again and urged seriousness in the matter, but Tawker replaced the bot's nomination statement again, this time under his own user account. Tawker then came to the aid of his bot in the election's designated discussion space, passionately defended the right of any editor – human or bot – with over a thousand edits to run in the election. Right on cue, the bot joined in the discussion and staunchly

---

<sup>7</sup> “Wikipedian” is an emic term used by active contributors in the Wikipedia projects to describe themselves, and by corollary, is sometimes deployed to exclude those who are not considered members -- readers, lurkers, donors, vandals, spammers, and self-promoters are not typically considered Wikipedians. I use the emic meaning of the term “Wikipedian” as I understand it throughout this work, in which status as a Wikipedian involves both frequent, reoccurring participation on the site, as well as a blurrier notion that Wikipedians participate in alignment with a broader set of norms, policies, processes, discourses, and ideologies which structure participation in the project.

<sup>8</sup> Administrators are volunteer editors who have been approved by “the community” in a vote-like process to have special technical privileges in Wikipedia. As of Summer 2015, there are approximately 1,600 administrators in the English-language Wikipedia.

<sup>9</sup> Note: all quotes from discussions in Wikipedia are directly copied and appear with no corrections. [sic] marks are not included due to the significant number of errors present in some of the quotes.

defended its place in the election by exclaiming, “I do not like this utter bot abuse. Bots are editors too!”

I make the same argument in this dissertation, although in a markedly different context. Tawker – speaking through his bot’s account – was ironically claiming that algorithmic editors like AntiVandalBot ought to be capable of running for the project’s highest elected position – and if successful, be allowed to influence the process of encyclopedia-building at one of its highest and most visible levels. I argue (with all seriousness) that these automated software agents have long had a substantial level of influence on how Wikipedia operates as a project to collectively author and curate a general-purpose encyclopedia – which has become a dominant site of cultural production despite facing continual criticism over its “anyone can edit” model. Bots are ubiquitous in Wikipedia and have long been explicitly delegated key editorial and administrative tasks involved in the encyclopedia project. There are 410 user accounts<sup>10</sup> that have been authorized to run 1,848 different automated tasks<sup>11</sup> on the English-language Wikipedia as of 20 April 2015. Yet for many readers and casual contributors to Wikipedia, it is quite possible to pass over the many different kinds of committees that have been delegated key decision-making responsibilities in the project – and by “committees,” I refer equally to collectives that are ostensibly made up of humans like ArbCom, as well as those ostensibly made up of algorithms like AntiVandalBot.

## 1.2. The politics of algorithms and the *Soylent Green* argument

When Tawker logged in under his bot’s account and participated in a discussion within one of Wikipedia’s formalized meta-level decision-making processes,<sup>12</sup> he surfaced an often obscured aspect of algorithmic systems: they are deeply human, despite the many longstanding and powerful narratives which emphasize the separation of humans and machines.<sup>13</sup> This argument – which I facetiously term the *Soylent Green* argument<sup>14</sup> – has a long lineage in the social scientific and humanist scholarship on science and technology, particularly from scholars in Science and Technology Studies (e.g. Law & Bijker, 1994). This argument has more recently been extended to scholarship on “the politics of algorithms” (for commentary on this emerging literature, see Barocas, Hood, & Ziewitz, 2013; Gillespie, 2014; Seaver, 2013). Just like the elected members of ArbCom, bots and their developers do not emerge in Wikipedia out of a vacuum. They are situated in and emerge out of a broad and diverse array of social and technical systems that exist in, around, and beyond Wikipedia. Such context makes bots difficult to examine and evaluate purely in terms of brute impacts and effects, although such impacts and effects are indeed substantial. For veteran Wikipedians who spend a substantial amount of their time contributing to the encyclopedia project, bots are part of the fabric of Wikipedia’s organizational culture and deeply integrated into many of the project’s average, everyday activities. Bots are sometimes invisible parts of the infrastructure that go unnoticed, but other

---

<sup>10</sup> List at <http://en.wikipedia.org/wiki/Special:ListUsers/bot>

<sup>11</sup> List at [http://en.wikipedia.org/wiki/Category:Approved\\_Wikipedia\\_bot\\_requests\\_for\\_approval](http://en.wikipedia.org/wiki/Category:Approved_Wikipedia_bot_requests_for_approval)

<sup>12</sup> Several levels of meta are at work in approving an ArbCom election nomination: it involves making a decision about how to conduct the process for deciding who ought to be making decisions about issues that have been left unresolved after multiple lower levels of dispute resolution.

<sup>13</sup> Such a separation is critiqued by many, including (Suchman, 2007), which I review and discuss in ch 2 and 3

<sup>14</sup> “It’s..... people!” (Heston, 1973)

times bots are hotly contested. Such debates are one of the many different ways in which Wikipedians work out fundamental issues about what they want Wikipedia to be – both a set of encyclopedia articles and a community responsible for curating those articles.

A secondary interpretation of Tawker’s exclamation that “bots are editors too” shows why it is important to emphasize the roles that people play in the design, development, and deployment algorithmic systems: bots are editors,<sup>15</sup> curators and gatekeepers of cultural content in one of the largest most visited websites in the world. On average, hundreds of edits are made to Wikipedia a minute, and there is a substantial amount of gatekeeping work that takes place in Wikipedia to enforce standards of quality and relevance. As I argue, the observed order of Wikipedia is not due to an alleged “wisdom of the crowds” (Surowiecki, 2004), but rather a highly organized assemblage of socio-technical systems, which are configured in particular ways and for particular purposes – and not others. A substantial amount of scholarship on Wikipedia has focused on the more human committees like ArbCom (e.g. Konieczny, 2010; Tkacz, 2015; Wattenberg, Viegas, & McKeon, 2007a). In addition, the more heterogeneous committees of humans and algorithms like AntiVandalBot are also crucially important to understand how Wikipedia – a site that is often raised as an emblem and an exemplar of an entire ‘revolution’ of “user-generated content” and “Web 2.0” – is curated as a site of cultural production. In order to understand how Wikipedia works today and how it has dramatically changed in the decade and a half since its creation in 2001, bots cannot be ignored, any more than any of the other ways in which Wikipedians work out what it means to use an “anyone can edit” wiki to author a general-purpose encyclopedia.

## **2. How to study algorithms: lessons from science and technology studies**

### **2.1 Studying the production of knowledge ‘in the making’**

There is a longstanding tradition in the field of science and technology studies (STS) for ethnographers to enter into the labs and workplaces of scientific inquiry, studying science “in action” (Latour, 1987) or “in the making” (Shapin, 1992). As Shapin summarizes, the goal of such research goes well beyond more longstanding efforts in the public understanding of science. Public understanding of science seeks to “inform the public about what scientists know” (28) and what the public ought to do given a certain state of scientific findings (often discussed as public policy or activism). In contrast to these more macro-sociological goals, many ethnographers and historians of science have taken a more micro-sociological route, gaining long-term, extended access to the average, everyday work that takes place in these

---

<sup>15</sup> However, among Wikipedians, the term “editor” is typically used in the way “user” functions in other sites to generally refer to anyone who takes any action on the site. This is in part because in the MediaWiki software hosting Wikipedia, almost every interaction involves editing some kind of wiki page: encyclopedia articles are publicly accessible wiki pages that (almost) anyone can edit, but are “talk” pages for discussing the content particular articles, “user talk” pages for interpersonal communication, and meta-level pages for documenting and debating the project’s norms. Wikipedians even submit cases to ArbCom by editing a new wiki page in a special section of Wikipedia, which is also how ArbCom’s decisions are published. ArbCom does have a private mailing list due to the committee’s general belief that the sensitive nature of the decisions they make requires them to take place “off wiki.” However, this is more of an exception and some Wikipedians object to such non-public discussion spaces as being “unwiki.”

relatively tight-knit and closed laboratories. With this privileged access, these researchers can certainly relate current findings in the field to the public and are likely in a better position to understand the role of science in public policy. However, with access before ‘the science is settled,’ they are also uniquely suited to study how scientists collectively work to produce the findings that the public typically has to take for granted. The findings of such ethnographies and histories have generally complicated more traditional idealized accounts of a universal scientific method – one that progresses linearly towards truth, driven by dispassionate individuals who are largely insulated from the ‘social’ phenomena that introduce bias into other forms of knowledge production. Such understandings of science were formalized by mid-20<sup>th</sup> century philosophers of science like Popper and Merton, and they are now a cornerstone of science education from an early age.

As Bruno Latour notes in his introduction to his ethnography of a laboratory at the Salk Institute, such universal visions of science become quite complicated and messy when observing science in the making (Latour & Woolgar, 1979). Shapin synthesizes a long line of research in STS when he discusses how scholars in the field have generally found cases speaking to topics like:

the collective basis of science, which implies that no single scientist knows all of the knowledge that belongs to his or her field; the ineradicable role of trust in scientific work, and the consequent vulnerability of good science to bad practices; the contingency and revisability of scientific judgment, and thus the likelihood that what is pronounced true today may, without culpability, but judged wrong tomorrow; the interpretive flexibility of scientific evidence, and the normalcy of situations in which different good-faith and competent practitioners may come to different assessments of the same evidence. (28).

Understanding these social aspects of scientific practice are not directly opposed to broader goals around the public understanding of science, nor do they summarily dismiss the validity of findings made by scientists – contrary to the claims of some in the so-called “science wars” of the 1980s and 1990s. The canonical science studies critique of objectivity as socially constructed is a subtle one that is tightly linked to the thick, rich descriptions of scientific practice sought by ethnographers and historians. The goal is to understand how objectivity is produced, investigating the many different ways in which scientists develop, negotiate, and practice such ideals. As Shapin discusses with climate research, “fairy-tales” about “a universal efficacious scientific method” (28) may seem to champion science, but they can actually make it difficult for those unfamiliar with the routine, mundane practices of scientific work to contextualize certain events that are made visible to the general public. A paper in which one scientific lab viciously critiques a dominant climate model of the greenhouse effect may be interpreted as an attack or even a dismantling of the finding (supported by that model) that the earth’s climate is warming. As Mody and Kaser (2008) find in their review of science studies scholarship on pedagogy, training, and professionalization of scientists, part of the professionalization process involves learning about the particular ways in which science is not an isolated enterprise, but takes place within the structures established through specific disciplines, labs, universities, funding structures, publications, professional associations, governments, popular media, and so on. Senior scientists know all too well that what is



published in peer-reviewed journals only represents a fraction of the work that goes on behind the scenes, as well as how science certainly does not operate in an isolated ivory tower independent from political and economic pressures (as much as they may try to insulate their subordinates from such pressures).

While these social aspects may be seen by some as evidence against the validity and objectivity of science, the canonical science studies approach sees them as what actually make certain forms of validity and objectivity possible. As Daston and Gallison argue around the emergence of photography in science, discourses celebrating objectivity as a way for “nature to speak for itself” (Daston & Galison, 1992, p. 81) belie a more complex, overlapping, and continually-evolving set of social and material practices. In this sense, objectivity is a moralizing ideal, one that establishes discourses, codes of conduct, procedures, and artifacts which humans use to judge other humans in the course of speaking for nature – not something humans do to let nature speak for itself. Similarly, Shapin and Schaeffer (1985) historicize the development of experimental methods, arguing that this form of knowledge production was built upon elements as diverse as the genre of the laboratory report and the networks of print that carried the first journals across continents to the socio-economic status of the ‘gentlemen’ across early modern Europe who had the resources and respectability that enabled them to dedicate substantial efforts to building air pumps and telescopes.

Ethnographers and historians have found countless cases of scientists not behaving as mechanically disinterested instruments for turning raw data into truth according to a monolithic scientific method. Instead, scientists have many different ways of constructing many different kinds of objectivity across various the contexts in which science takes place. In the 1990s, some academics and commentators of early science studies literature argued that scholars like Latour and Shapin were saying that science is no different than fiction or religion, such that any claim to truth is nothing more than an arbitrary expression of brute power relations that has been improperly elevated to the status of truth.<sup>16</sup> Furthermore, a number of scholars (as well as journalists and activists) were increasingly focused on investigating and exposing bias, malpractice, corruption, and fraud in science, particularly in laboratories that had close ties to either the military or large corporations. These characterizations of science as ‘merely’ a social construction aligned with a number of movements across the political spectrum, from those who denied that the ‘science was settled’ regarding evolution or global warming as well as around genetically-modified organisms or pharmaceuticals. As Bruno Latour notes in a reflection on this “weaponization of critique” (Latour, 2004) the goal of science studies is instead to give context to these forms of knowledge production, showing how objectivity and truth have the strength that they do precisely because they are socially constructed.

## **2.2 Algorithms as a mode of knowledge production**

Algorithmic systems are similarly deployed in sites of knowledge production, with many similarities to scientific inquiry. Such systems are often presented as more objective and universal ways of knowing the world, particularly those that are bound up in the analysis of

---

<sup>16</sup> One lay version of this is the belief that statistics can be manipulated to say anything, such as in Mark Twain’s quote about there being “lies, damned lies, and statistics” or Darrell Huff’s “How to Lie With Statistics.”

“Big Data” -- which a number of scholars have argued is best seen as a contemporary moment in which longstanding debates about objectivity and positivism are yet again playing out (boyd & Crawford, 2011; Gillespie, 2014; Jurgenson, 2014). I see much of the contemporary critical scholarship and commentary around algorithms as more analogous to the lineage of science studies than to the related wing of the Science and Technology Studies field that has focused on the social aspects of technological design and development. First, with the growing scholarly and public attention on algorithmic systems – which Seaver (2013) has termed “critical algorithms studies” – there are increasing efforts around the public understanding of algorithms. This scholarship includes documenting recent developments of algorithmic systems that are of importance of the general public, discussing broader societal and public policy concerns that arise from such developments, and the more generalizable issues about how technocratic experts and publics interact. Like Shapin, I do not take issue at all with these more accessible explanations about what kinds of algorithmic agents are operating in our digitally-mediated public and private spaces or the societal and public policy focus on what to do about it; these help a wide range of social groups understand and relate to the world around them.

There is a second set of scholarship and commentary focused on critiques of algorithmic systems (and “big data”), which have largely emerged as responses to more celebratory discourses emphasizing the objectivity of such approaches to knowledge production and decision-making. Scholars like Morozov and Jurgenson draw extensively from the lineage of STS ethnographers and historians who have studied science in the making, using the findings about the social construction of scientific objectivity to dismiss the hyperbolic claims by those like Chris Anderson and Tim O’Reilly. Such critiques often discuss the rhetorical power of casting algorithmic systems as ways to remove humans from the loop and let data ‘speak for itself,’ arguing how such discourses erase the people who actually do this work – who are generally quite privileged, non-representative of the general population, and in positions where such algorithmically-produced knowledge can benefit them greatly. Yet considering the overwhelming amount of critique and commentary drawing on lessons learned from studies of science in the making, the field is severely lacking in these kinds of highly-situated, empirically-driven studies about algorithmic systems. It is crucial to study algorithms in the making, investigating how computer scientists and software developers come to make what they make. Like with science in the making, it is important to do this before the source code is finalized and deployed in the broader systems that members of the general public use and interact with on an average, everyday basis (as well as those that members of the public do not use, but are instead used to make judgements on a wide range of issues).

Another reason why it is important to study algorithms in the making in the same way that STS scholars have studied science in the making is due to the second similarity I have found between the social study of science and the social study of algorithms. In the broader scholarly and popular discourses around algorithms, I see a polarization that mirrors the twin extremes of the “science wars” debates of the 1990s, particularly around the issue of algorithmic objectivity. On one extreme are the celebratory positions that imagine algorithmic systems as the ideal ways to objectively know the world and make decisions about it, often cast as taking decision-making away from subjective, emotional, and error-prone humans. Chris Anderson famously declared “the end of theory” (Anderson, 2008) due a rise of automated, data-driven scientific inquiry, while Tim O’Reilly has advanced a framework of “algorithmic regulation”

(O'Reilly, 2013) and “government as a platform” (O'Reilly, 2011). O'Reilly's framework is based on the premise that if a governmental regulation or law can be computationally specified and enforced, it should be. Even some humanistic critiques of these algorithmic systems even accept the premise of algorithmic objectivity as the ultimate science, like in Lewis's celebration of human error: “Algorithms, which are by definition more objective, have no human fallibilities ... Making mistakes is a human frailty ... the essence of what it means to be human” (Lewis, 2014).

On the other extreme are those who critique algorithmic systems of knowledge production as superstition, mysticism, or pseudoscience, arguing that the outputs of algorithmic systems express nothing more than brute power relations. In fact, in many of these critiques, the argument is that unlike ‘real’ science – which is often counterpoised as an ‘actually objective’ mode of knowledge production – algorithmic systems should be given no more epistemological weight than judgements made by humans. For example, Robin James compares algorithmic forecasting to astrology, locating “the same pseudo-rationality found in both astrology and big data” as “a trendy, supposedly more objective upgrade to unfashionable superstitions” (James, 2015). Such critics typically discuss two kinds of concerns, which both differently draw on ideals of a universal scientific method. Some focus on mistakes and biases that are made by such systems as evidence of their lack of objectivity, while others focus on the fact that the inner workings of such systems are not made public and open to review – or argue that due to their complexity, not even their developers can truly know how such systems operate. For example, Cory Doctorow argues in a lengthy critique that a crime prediction system developed for the Chicago Police Department is “the Big Data version of witchburning, a modern pseudoscience cloaked in the respectability of easily manipulated statistics,” drawing on a Popperian notion of science as openness:

the attribution of guilt (or any other trait) through secret and unaccountable systems is a superstitious, pre-rational way of approaching any problem. ... The core tenet of science, the thing that distinguishes it from all other ways of knowing, is the systematic publication and review of hypotheses and the experiments conducted to validate them. The difference between a scientist and an alchemist isn't their area of study: it's the method they use to validate their conclusions. An algorithm that only works if you can't see it is not science, it's a conjuring trick. (Doctorow, 2014)

Evgeny Morozov, who has continually focused on issues of algorithmic governance, critiques algorithmic regulation on issues of transparency and accountability, making the common argument that casts “algorithms” as mystical processes that even their developers cannot know or understand. He argues this in a critique of a piece by Tim O'Reilly, who had claimed that algorithmic systems would be better for governance than contemporary bureaucracies, because rules and procedures could be computationally specified and enforced. Morozov contests this, arguing that “instead of seeing the logic driving our bureaucratic systems and making that logic more accurate and less Kafkaesque, we would get more confusion because decision making was becoming automated and no one knew how exactly the algorithms worked” (E. Morozov, 2014).

### 2.3 The black box

One of the core issues stems from the general inaccessibility of the spaces where most algorithmic systems are developed, both in terms of gaining access at all as well as gaining expertise to understand the development of such systems. As Seaver (2013) argues, this opacity makes it easy to attribute a mystical quality to what is produced there and the people who produce it. To avoid falling into the trap laid when examining only ready-made algorithms, ethnographers must look into the broader “algorithmic systems” in which those algorithms are designed, developed, and deployed. Seaver argues that this work of studying algorithms in the making involves a broader scope than the standard computer science definition of the term “algorithm” as a defined, encodable procedure for turning data inputs into data outputs. Rather, he states:

It is not the algorithm, narrowly defined, that has sociocultural effects, but algorithmic systems — intricate, dynamic arrangements of people and code ... When we realize that we are not talking about algorithms in the technical sense, but rather algorithmic systems of which code *strictu sensu* is only a part, their defining features reverse: instead of formality, rigidity, and consistency, we find flux, revisability, and negotiation. (Seaver 2013, 9-10)

Like with the longstanding science studies position on objectivity, there we should not deny the existence of objectivity with algorithms (nor should we instinctively lash out at anyone who uses such a term in relation to algorithms). Rather, a more situated approach is to understand the ways in which algorithmic objectivity is made more and less possible, especially across different contexts. This certainly does not mean following Anderson’s claim that “numbers speak for themselves,” any more than Daston and Gallison accept the scientists who make similar claims about photography as letting nature speak for itself. Rather, this means that objectivity is a complex and continuously-evolving ideal, an inherently fuzzy concept that humans develop and deploy in conversations and controversies they have in relation to the active, ongoing development and deployment of algorithmic systems. And these ideals and socio-material practices are as much a part of the algorithmic system as its source code.

The presumed opacity of the site where such systems are developed risks a fetishization of algorithmic systems, potentially locating some special essence of algorithms inside a conveniently inaccessible interior space. Many academic and popular commentators have taken up calls to release the source code of algorithms as ways to “open up the black box” of algorithmic systems; these calls for algorithmic accountability or transparency often assume that the relevant factors of an algorithm are found in its source code. Yet as Bruno Latour argued in his studies of scientific laboratories (Latour, 1987), what is black boxed is the entire enterprise of science, such that science is only graspable as something that takes in data about the natural world (and possibly grant funding) and outputs objective findings. Like with the lineage of social studies of science, work on the public understanding of algorithmic systems and the roles of algorithmic systems in society must be accompanied by a related but distinct inquiry of algorithms in the making. Seaver – who has conducted extended fieldwork in a major corporation that develops machine learning systems for recommending music to their clients – argues for such broad, sustained ethnographic inquiries into these spaces where algorithms are

developed. In his own fieldwork, he writes about the shift between his initial assumptions about algorithms and what has emerged through such extended participant-observation – a shift that mirrors my own:

These algorithmic systems are not standalone little boxes, but massive, networked ones with hundreds of hands reaching into them, tweaking and tuning, swapping out parts and experimenting with new arrangements. If we care about the logic of these systems, we need to pay attention to more than the logic and control associated with singular algorithms. We need to examine the logic that guides the hands, picking certain algorithms rather than others, choosing particular representations of data, and translating ideas into code. (9).

If we want to problematize the determinist narratives of inevitability around algorithmic systems and support the work of those involved in the public understanding of and responses to algorithms, it is not enough to make the *Soylent Green* argument common in critical algorithm studies: to point out that algorithms are made of people; that they are designed, developed, and deployed by those in privileged positions to do so, who engage in this work in ways that make sense to them given their positions in society. Problematization involves more than summary dismissal of a narrative; it demands consideration of what supports and structures such seemingly ‘obvious’ assumptions. In this case, the opacity and inaccessibility of the spaces where algorithmic systems are designed, developed, and deployed contributes to the broader characterization of algorithmic systems in general as inherently opaque, fully-autonomous agents that invisibly dominate human activity in ways that even their creators cannot understand. This inaccessibility also plays into the less severe narratives, which imagine solutions to algorithmic accountability as a problem of getting access to the source code and the expertise to understand such code. Given such demands for algorithmic transparency and accountability, I ask: are the problems critics raise around algorithmic systems inherent in the process of developing encodable procedures to autonomously make decisions, or are they based more in the lack of transparency and accountability by the institutions and organizations that have increasingly turned to algorithmic systems to make decisions? In other words, what would we see if we were able to examine not only the source code of algorithmic systems that support the production of knowledge and the regulation of behavior, but also the full range of their design, development, and deployment? What if we got everything that the critical scholars and activists are asking for in opening up the black box of algorithmic systems?

## **2.4 Methods for studying black boxes**

The core problem around algorithmic systems remains: what do we do when the space where such technology is designed, developed, and deployed is generally impenetrable, such that everyone except for a select few are limited to dealing with these spaces as black boxes? Scholars seeking to engage in such STS-based ethnographic work on algorithmic systems frequently lament their inability to gain access to these sites (Barocas et al., 2013; Gillespie, 2014). This is a longstanding methodological problem faced by those “studying up” (Nader, 1969), whether those being studied are the administrators of colonies or social networking sites. Scholars of algorithmic systems who do not have privileged access to such sites instead generally rely on what is made public about such systems in often-vague statements from their

developers or using reverse engineering, which Diakopoulos (2015) reviews in the context of “algorithmic accountability.” One powerful technique is to make use of filtered/unfiltered feeds like those in Facebook’s algorithmically-filtered news feed: the filtered feed is displayed by default, but can be instead sorted chronologically to show all posts. In one study, researchers had people compare their filtered and unfiltered news feeds, using this as a way to explore what kinds of content was being filtered and provoke reflection about this algorithmic filtering by people who use Facebook (Eslami et al., 2015). Others trace the contours of algorithmic systems as they are made visible to end-users in more subtle ways, as in Crawford and Gillespie’s analysis of interfaces in major social networking and social media sites developed for users to flag or report inappropriate content. They argue that such interfaces are articulations of a “vocabulary of complaint” that structures a highly-automated human-computational system used for moderation work, the inner workings of which are opaque to all but a few who work for Facebook, YouTube, Twitter, etc. (Crawford & Gillespie, 2014).

In investigating these issues, my strategy has been to leverage my existing ethnographic experience in a space where algorithmic systems proliferate: Wikipedia. My original research questions involving Wikipedia did not involve these issues of algorithmic systems, but were rather focused on issues of community and organization. Through extended participant-observation, interviews, and descriptive statistics, I have conducted a substantial amount of research over the past nine years on how Wikipedia operates as an encyclopedia project that is authored and curated almost exclusively by volunteers. Later in this research, I have paid more attention to the fact that a multitude of automated software agents operate in Wikipedia, performing a wide variety of work in directly editing encyclopedia articles and more administrative and higher-level tasks. Taking my existing level of access and expertise within Wikipedia more broadly, I have turned my attention towards these “bots” and explored how they have been designed, developed, and deployed in the encyclopedia project. Such work has been made possible both through my insider status within Wikipedia, as well as my background in software development. Together, these skills give me the ability to not only look into the source code of bots that work in Wikipedia, but also understand the broader context in which those algorithmic systems operate. Furthermore, I also found that in two online spaces in which I have been an active participant for multiple years – Twitter and reddit – automated software agents also proliferate, performing different kinds of tasks that nevertheless have a significant presence on how people interact on these sites. I only focus on bots in Twitter in this dissertation, but similar issues also arise in reddit, which I leave for future work.

Unlike the algorithmic systems that automate various activities in Facebook, Google, or the NSA, the bots I studied in Wikipedia and Twitter were not designed, developed, or deployed internally by staff at the organizations that own and operate these websites. Rather, these algorithmic systems are generally run by relatively-independent volunteers. The software code supporting such algorithmic systems is also not directly integrated into server-side codebases; they are often programs that run on a developer’s personal computer or cloud computing services. They act in the capacity of user accounts, acting with the same affordances and constraints as the accounts that humans use, and their activity is generally visible (or invisible) as that of any other user account. With the bots I have studied in researching this dissertation, a far wider range of the design, development, and deployment involved in algorithmic systems is publicly accessible; no NDAs were violated in the making of this dissertation. Furthermore,

almost all of these bots are openly identified as bots by their developers, unlike most “socialbots” that pose as humans. (Those bots, used for spamming, astroturfing, identity theft, etc. are generally seen as inherently malicious and a violation of a site’s terms of service, and therefore typically do not advertise their status as bots.) The bots I studied in Wikipedia and Twitter are of a class of automated software agents that are generally accepted as potentially legitimate actors by the organizations that own and operate these websites – typically a tacit acceptance, in that bots deemed to be malicious or disruptive can (and are) be blocked server-side by administrators. As I found in many of the cases I studied, there are passionate debates that erupt between a bot’s developer and others on the site about whether a particular bot ought to exist. Sometimes, petitions are made to server administrators to block controversial bots.<sup>17</sup> Other times, debates occur between non-developers over a hypothetical, proposed bot that may never come to exist. While there are many differences between Wikipedia and Twitter, both are sites which user-authored bots have significant impacts for those who use them and are made the subject of public debate – unlike in sites that universally prohibit all user-authored bots, like Facebook.

As I have investigated the exceptional and mundane work that takes place around bots in Wikipedia and Twitter, the easier narratives emphasizing the inevitability, opacity, instrumentality, and invisibility of algorithms have given way to more complex stories of contingency, negotiation, and a plurality of partial perspectives. My exploration into the average, everyday practices around the design, development, and deployment of algorithmic systems revealed that people are not “algorithmic dopes” – taking from Garfinkel’s critique of the “cultural dope” (Garfinkel, 1967). I found that many people (including those who are not software developers) express strong beliefs about how they think an automated software agent ought to be programmed to perform a certain task, because many of them have very strong ideas about what kinds of tasks are important in these spaces and how they ought to be performed. In fact, I argue that these bots are often most important not for the specific tasks they have been delegated, but for the broader conversations they provoke as people work to articulate and negotiate what kinds of socio-technical systems they want to exist in the world.

At one level, these bots have been delegated important work in the spaces they continuously inhabit, and the tasks they perform certainly have far-reaching impacts and implications as to how sites like Wikipedia and Twitter operate in the manner that they do. Yet at a broader level, these automated software agents are even more important as pivotal moments in which fundamental values, goals, ideals, and visions are articulated, made explicit and contestable. For example, a spell-checking bot is unleashed on the entire English-language Wikipedia, using an American English dictionary to ‘correct’ everything it analyzes as misspellings. This understandably sparks a debate, which is ultimately not about which national variety of English ought to be used, but rather over whether Wikipedia ought to uniformly

---

<sup>17</sup> Twitter has a general bot policy (~1,000 words in June 2015) that largely lists prohibited actions, which is enforced by staff at Twitter, Inc. Wikipedia has a far more developed and devolved bot policy and process in which bots must be approved by an ad-hoc committee comprised of volunteers who are active in contributing to Wikipedia; the committee is a mix of bot developers and non-developers, many of whom have been delegated the authority to block (or unblock) any particular account from editing.

adhere to any national variety of English across all encyclopedia articles. In deciding to not allow any fully-automated spell-checking bots to operate site-wide, Wikipedians not only came to a decision about this particular algorithmic agent, but worked further towards a broader collective understanding about how they thought Wikipedia ought to exist as an encyclopedia – a specific genre of text – read and authored by people from all around the world. In this dissertation, I tell of many similar cases and controversies about bots in Wikipedia and Twitter, which give us a quite different understanding of what algorithmic systems are and what roles they play in the governance of user-generated content sites.

### **3. Chapter overview**

#### **Section 1: An algorithms-in-the-making approach (ch 2,3)**

The chapters in the first section introduce the core concepts that I use to study bots in Wikipedia and Twitter from an algorithms-in-the-making approach: a focus on bots as bespoke code, which calls attention to the infrastructures that support automated software agents, then a focus on the delegation of governance work through such bots. In this way, I study bots before they become the kind of ready-made artifacts that appear as external to societies and organizations (and are therefore best studied for their impacts and effects). In my approach, bots as bespoke algorithmic systems are one of many ways in which people work to understand, articulate, negotiate, and enact norms and meta-norms about how Wikipedia as a digitally-mediated environment is and ought to be. These chapters expand my analytical frame of a bot beyond just seeing bots as software agents powered by source code. I show how they are ongoing projects: first, a team effort by the bot’s operator to ensure that the software agent keeps running; and second, a collective effort by the bot’s operator and others in Wikipedia to decide what the software agent ought to do and how.

#### **Chapter 2: Bespoke Code and the Bot Multiple**

In chapter 2, I discuss one aspect of an algorithms-in-the-making approach, which investigates the specific material and infrastructural contexts in which software code is designed, developed, and deployed. In my studies of bots, I found that the kind of code of the bots I studied was not generally designed, developed, and deployed by ‘server sovereigns’ who had the privilege to modify server-side code. Instead, this work was done by people who acted relatively independently from the organizations that owned and operated the site’s servers. Bots are what I term “bespoke code” (Geiger, 2014) a broader phenomenon I introduce and discuss in this second chapter. This sensitizing concept calls attention to software code that extends or modifies the functionality of centrally-hosted digitally-mediated environments in ways that are traditionally limited to server-side code. Bespoke code – which includes fully-automated bots, third-party ‘power tools,’ browser extensions, and mashups – raises many compelling issues about the sociality of software and the politics of algorithms.

In this second chapter, I discuss how bespoke code has emerged as a response to the rise in cloud computing and software-as-a-service, in which applications are increasingly run as server-side “platforms” that people access through web browsers, rather than standalone applications. While this may initially seem to give server sovereigns even more authority over



how to enact a particular form of life on the people who access their sites and services, bespoke code complicates this more traditional narrative that privileges the server as a site of inquiry. I then discuss how bespoke code as a concept calls attention to the material and infrastructural aspects of software, aligned with but extending a long literature across various fields (e.g. Blanchette, 2011; Star, 1999). I argue that bespoke code is a particularly compelling example showing how “opening up the black box” of software should include but go beyond analyzing source code to see how a bot operates. The social study of software in the making also involves investigating the specific, concrete, historically-contingent contexts in which that code is designed, developed, and deployed. I empirically support such an argument by relating a set of vignettes about my own experience as a bot developer in Wikipedia, emphasizing the importance of looking beyond source code to understand issues relating to the broader “algorithmic systems” (Seaver, 2013) in which that code is embedded.

### **Chapter 3: Exclusion compliance**

In chapter 3 on “exclusion compliance,” I expand the previous chapter’s focus on the materiality of bespoke code to issues around the delegation of governance work to algorithmic systems. I draw on theories of delegation from science and technology studies and organizational studies, as well as a variety of other approaches that emphasize the indeterminate and constructed nature of information technology. I draw on Bruno Latour’s writings on delegation in science (Latour, 1999a) and engineering (Latour, 1992) as well as the work of other actor-network theorists, who conceptualize society as a heterogeneous network of assembled relations between human and non-human actors (Akrich & Latour, 1992; Callon, 1986; Law & Mol, 1995). Moments of delegation to automated software agents – particularly that of governance work – are key sites in which broader values and assumptions become visible and articulated. Controversies around delegation are a rich site of inquiry, as they often demand that participants reflect on the broader issues, practices, and concerns that exist across an organization.

This chapter of the dissertation examines these theoretical issues through a case study of a controversy over the first bot in Wikipedia that was delegated the work of enforcing a norm about how Wikipedians were to interact with each other (as opposed to earlier bots that made automated edits to the text of encyclopedia articles). This norm – that people should sign and date their comments in discussion spaces, a feature not built into the stock MediaWiki platform – was seemingly universal and uncontroversial, enshrined in an document in Wikipedia’s “policy environment” (Beschastnikh, Kriplean, & McDonald, 2008). However, when a bot began universally enforcing this norm, those who felt they had the “right” to not have their comments and/or signed and/or dated appeared to contest the code of the bot. In the ensuing controversy, bot developers and non-developers debated not only the norm of signing and dating comments, but also the meta-norms about if and when Wikipedians could opt out of having a bot enforce such a norm. The compromise reached in this case established both new norms and new software-based standards guiding how bot developers were to interact with those who objected to bots – or more accurately, those who objected to a bot developer’s normative vision of how Wikipedia ought to operate, which was being implemented project-wide through the code of the bot.

## **Section 2: Bots in administrative spaces in Wikipedia (ch 4, 5, 6)**

In the three chapters that comprise this section of the dissertation, I use the algorithms-in-the-making approach established in section 1 to study the bots involved in Wikipedia's behind-the-scenes administrative spaces. These are the specialized processes in which highly-active and veteran Wikipedians perform specific commonly reoccurring tasks, such as conflict resolution or deciding which articles ought to be kept or deleted. I argue that an algorithms in the making approach is crucial to understand how the encyclopedia project has shifted dramatically over its 15 year history, going from being a frequently-lampooned site of cultural production where the only rule was "Ignore All Rules" (Reagle, 2010) to a highly-organized and bureaucratic organization. Bots are crucial to understanding the contemporary operation of Wikipedia as a now-dominant site of cultural production, whose veteran members rely on them to perform a substantial amount gatekeeping work needed to curate content contributed to the encyclopedia. Furthermore, bots are also crucial to understanding how the project came to exist in this way and not others –although I do not advance a 'rise of the machines' trope in which I hold these algorithmic systems (or even their developers) solely responsible for the rise of Wikipedia's administrative processes. Rather, I argue that bots and other bespoke code were and continue to be one of many ways in which Wikipedians work out high-level issues about how to govern a site.

### **Chapter 4: Articulation Work**

In chapter 4, I introduce these specialized processes and venues within the context of the bespoke code that supports them. Many Wikipedia researchers have explored specific processes that make up "the hidden order of Wikipedia" (Wattenberg et al., 2007), but I first provide an updated description of the prevalence of these spaces using a mix of ethnographic, historical, and statistical methods. I then extend this literature by showing how such processes are supported by automated software agents and other bespoke code. Such bots do not generally autonomously make independent decisions, determining the outcome of specific administrative issues. Instead, these "clerk" bots (as their developers sometimes call them) are more often involved in "articulation work" (Strauss, 1985), which involves coordinating the more human Wikipedians as they participate in these formalized practices. I use the concept of articulation work to show how such bots can become sites of contestation and negotiation for much broader issues; a bot does not have to be fully delegated a task to provoke and resolve debates about how that task ought to be done or why the task is important. My account of these bot-articulated administrative spaces also challenges longstanding characterizations of Wikipedia as an unstructured anarchy ruled only by a mystical "wisdom of the crowds" (Surowiecki, 2004), which is common in both popular culture and academic scholarship. In performing articulation work, bots provide an organizational infrastructure that makes it possible for veteran Wikipedians to quickly and efficiently perform a wide variety of tasks at scale – in a way that can easily be seen as a form of "uncoordinated coordination" (Benkler 2007, p. 5) that economists like Yochai Benkler celebrate in theories of peer production, who argue that sites like Wikipedia lack traditional organizational structures and instead operate in ways that are closer to marketplaces.

## **Chapter 5: Membership and socialization**

In chapter 5, I discuss the relationship between these increasingly formalized administrative processes and issues of participation, socialization, and membership in Wikipedia. Becoming a Wikipedian involves not just learning the project’s discourses and norms of participation, the wiki’s complicated user interface, or specialized reference and writing skills, which many Wikipedia researchers have analysed (e.g. Bruckman, Bryant, & Forte, 2005). Newcomers must also become familiar with these automated processes and the bots that coordinate administrative work as an editor goes about many average, everyday activities. Many early accounts of socialization in Wikipedia celebrated a successful model of “legitimate peripheral participation” (Lave & Wenger, 1991), where newcomers gradually take on increasingly complex tasks (reviewed by Preece & Shneiderman, 2009). However, newcomers are now often immediately thrust into these venues as soon as they make a contribution, a phenomenon my collaborators and I have termed “gatekeeping socialization” (Geiger et al., 2012) which has significantly increased attrition of newcomers (Halfaker et al., 2013). In response, Wikipedians have created different kinds of newcomer-focused specialized venues, using bespoke code to create safe mentoring spaces for newcomers to get help. These issues show how, like many information technologies in organizations, the bots that support Wikipedia’s specialized venues are best seen not as “automated plumbing” that simply makes Wikipedia more efficient, but something that is deeply woven into the “fabric” of the project’s organizational culture (Orlikowski & Scott, 2008; Zammuto et al., 2007) – learned as a part of membership and integrated into everyday practices. In applying these lessons from the study of information technology in more traditional organizations, I further specify my algorithms-in-the-making approach to include how bots have become part of the average, everyday work of being a veteran Wikipedian.

## **Chapter 6: An algorithmic history of Articles for Deletion**

The previous two chapters in this section illustrated various elements of Wikipedia’s specialized administrative spaces. In these bot-supported spaces, a wide range of decisions are made across the encyclopedia project, which is a core mode of participation for many of the project’s most active contributors. In chapter 6, I ask: how did such a situation develop as it is now, especially given the early rhetorical position of Wikipedia as an anti-institutional mode of cultural production? And did such processes initially come on the scene with the kind of highly automated algorithmic support I detailed in Wikipedia’s contemporary operation in the previous chapters? This second question is easier to answer than the first, but I discuss both in analyzing the history of Wikipedia’s process for deciding which articles ought to be deleted from the encyclopedia. This was the first of Wikipedia’s processes to be formalized starting in 2002, and these same processes were also the first to be automated using bots and other bespoke code years after their initial formation. I show how such “clerk” bots were designed, developed, and deployed as Wikipedians worked out a broader project-wide shift in focusing on quality and gatekeeping, which played out in conflicts between “inclusionists” and “deletionists.” This section extends scholarship by Wikipedia researchers who made similar arguments about this factional conflict in the history of Wikipedia (Kostakis, 2010; Reagle, 2010), but have largely limited their methods and cases to the more immediately ‘human’ aspects of the project. My contribution with this historical analysis is to make such an argument using an algorithms-in-

the-making approach, which illustrates how code can be a way in which people work to imagine, articulate, negotiate, enact, and contest ideas about what a site like Wikipedia is and ought to be.

### **Section 3: Blockbots in Twitter (chapter 7 and 8)**

This next section asks: are the issues and frameworks which have been useful to study the delegation of articulation work to bespoke code limited to Wikipedia, or can similar kinds of issues be studied in similar kinds of bots that exist in other sites? I focus on the social networking site Twitter, run by the for-profit corporation Twitter, Inc. In this context, I examine bot-supported collective blocklists, or “blockbots,” which were created in response to coordinated harassment campaigns on Twitter. Like bots in Wikipedia, blockbots are bespoke algorithmic systems which have been designed, developed, and deployed by volunteers who are relatively independent from staff at Twitter, Inc. Blockbots support the work of curating lists of identified or suspected harassers and then automatically removing such accounts from subscribers’ experience of Twitter – functioning similar to ad blocking from a user experience perspective. I discuss how blockbots are delegated articulation work around the task of curating a list of blockworthy accounts, supporting counterpublic groups as they seek to collectively moderate their own experiences of Twitter. I also discuss how this bot-enacted governance work is quite different in Twitter than it is in Wikipedia for a variety of factors, which speaks to broader issues about the conditions in which algorithmic governance takes place.

### **Chapter 7: The blockbot tutorial-ethnography: a composite account**

In chapter 7, I present a composite ethnographic account of blockbot development, drawing on the genre of a software development tutorial. Collective blocklists were not a feature built into Twitter by designers and developers at Twitter, Inc., but due to the particular configuration of other affordances implemented into the site’s server-side codebase, these bespoke blockbots were possible in Twitter in ways that they were not in other social networking sites (most notably Facebook). Yet in order for any given person who uses Twitter to be able to click a few buttons and have hundreds or even thousands of Twitter accounts identified as harassers removed from their experience of Twitter, many different kinds of systems must be aligned. This software tutorial uses second-person declarative statements, expository text, excerpts of source code, and increasingly-complex diagrams to place the reader in the position of a blockbot developer. The tutorial does not describe any particular individual, as it is assembled from many different cases I have seen in my studies of blockbots. I begin with automating a basic task in Twitter using the Application Programming Interface, then successively include the various heterogeneous elements and activities involved in designing, developing, and deploying a blockbot. In doing so, I take the reader through the same expansion of the analytical frame of the bot as I did in section 1, moving from the bot as a software agent powered by source code to a project of collective sensemaking, where decisions about what kind of articulation work the bot ought to perform raise much larger issues about what it means to curate a shared blocklist on Twitter. This account also extends beyond the “technical” work of programming, illustrating how developing and operating a blockbot involves tasks like community building, fundraising, and responding to threats from hostile opponents.

## **Chapter 8: Blockbots as projects of collective sensemaking and reconfiguration**

After giving a rich description of the kind of work that is involved in developing and operating a blockbot, I move to more specific empirical cases. I present a historical account of the first major block bot to operate in Twitter, showing how this project developed and changed over time. I analyze blockbots as information infrastructures bound up in a classification problem (Bowker & Star, 1999): deciding who is and is not a blockworthy individual. I focus on specific moments of controversy in which the bot's developer reconfigured the software agent to differently perform the articulation work of supporting this classification problem. I then more briefly discuss a more recent case that took the blockbot model in a different direction, using a second automated software agent to generate a list of blockworthy accounts. Yet even in this seemingly fully-automated system, I show that there is still a substantial amount of human work involved. Most notably, the bot was reconfigured to include a human appeals board, which established their own deliberative processes for deciding which accounts ought to be whitelisted from the algorithmically generated blocklist. Because such bots perform articulation work for a broader sociotechnical project, encapsulating an organization, the specific configuration of their source code can become a site for the negotiation of broader ideas about what Twitter as a social networking site ought to be. Ultimately, I argue that blockbots are important not only in the impacts they have on letting targets of harassment campaigns moderate their own experiences of Twitter; they can also provoke debates and discussions about how Twitter as a privately-owned public space ought to be governed and moderated.

## **Chapter 9: Conclusion**

In the conclusion, I first give a summary of the overall argument of the dissertation, elaborating on the algorithms-in-the-making approach I took in studying bots in Wikipedia and Twitter. Next, I identify three different themes that have emerged in these different cases: proceduralization, the regulation of bots, and bots as speculative projects. Finally, I conclude with an essay on the ideological assumptions that are present in many of today's major user-generated content sites, including not only Wikipedia and Twitter, but sites like reddit as well. I stress that we must not forget that bot developers are volunteers who support moderation practices that benefit the platform's owner-operators by keeping the site orderly, but in ways that neither cost them money nor open them up to critiques of censorship or gatekeeping.

## Section 1: Algorithms in the Making (chapter 2 and 3)

### Chapter 2: Bespoke code and the bot multiple

#### 1. Introduction

##### 1.1 Defining and contextualizing bespoke code

In this chapter, I introduce and discuss the concept of *bespoke code* (Geiger, 2014), which I define as software code that runs alongside a centrally-hosted digitally-mediated environment, rather than code that is directly incorporated into server-side codebases and runs on the same servers that host a site. As a sensitizing concept (Blumer, 1954), bespoke code calls attention to how this increasingly prevalent type of software development can be used to extend, modify, or even subvert the functionality and operation of centrally-hosted digitally-mediated environments – particularly in ways that have typically been limited to the ‘server sovereigns’ who have an exclusive privilege to modify server-side code. Bespoke code includes the fully-automated bots I discuss throughout this dissertation in Wikipedia and Twitter, as well as third-party clients,<sup>18</sup> third-party services,<sup>19</sup> browser extensions,<sup>20</sup> and mashups.<sup>21</sup>

The word “bespoke” traditionally describes highly-customized fashion, such as a bespoke suit; the Oxford English Dictionary defines it as “goods; ordered to be made, as distinguished from ready-made.” (OED, 2013). I have seen sparse use of “bespoke code” in software development forums to refer to custom-made software based on client specification, but I use the term differently, to identify and consolidate a large range of practices I have seen in not just Wikipedia, but a variety of other contemporary software development environments. Like a made-to-order dinner jacket, “bespoke” indicates that this code is highly-customized and specifically written (and rewritten) to fit some already-existing entity. In my experience as a developer of Wikipedia and Twitter bots as well as a user of many browser-based add-ons, scripts, and extensions that dramatically change the way I experience the web, this constant customization has emerged as the one most salient commonalities between these distinct but linked software development practices. In my specific studies of bots (a subset of bespoke code) I have conducted in Wikipedia and Twitter, my analytical focus on the bespokeness of bots surfaces several issues around the sociality and materiality of software.

In this chapter, I first introduce bespoke code generally to define and contextualize this concept, then give several examples of how bespoke code operates in Wikipedia, including vignettes relating my own experience as a bot developer in Wikipedia. The goal of this chapter is to establish *bespokeness* as the first of several core concepts which are involved in the *algorithms-in-the-making* approach to bots I take in this dissertation. I focus on bespokeness to show that what it means to ‘unpack’ or ‘open up the black box’ of algorithmic systems should

---

<sup>18</sup> Including ‘power tools’ like the Hootsuite tool used heavily by corporate social media teams, who use social networking sites in ways that are quite different than profiles inhabited by individuals (Geho & Dangelo, 2012)

<sup>19</sup> Such as Google’s reCAPTCHA (von Ahn, Maurer, McMillen, Abraham, & Blum, 2008) or Facebook’s JavaScript like button that can be added to non-Facebook webpages (Gerlitz & Helmond, 2013)

<sup>20</sup> Such as AdBlock, the Facebook Demetricator (Grosser, 2013), or Turkopticon (Irani & Silberman, 2013)

<sup>21</sup> See (Wong & Hong, 2008)

not be thought of as just an exercise in parsing through server-side code – or even all the bespoke code that runs alongside it. Reading code can be quite revealing, but it requires a relational, networked, infrastructural understanding of code and how it is socio-materially situated in many different kinds of spaces.

## 1.2 Why study bespoke code?

Bespoke code as a mode of software development has strongly emerged in response to software-as-a-service and cloud computing, in which software programs are increasingly run on servers instead of local machines. In this model, users typically access an interface via a web browser, like with Gmail and Google Docs compared to Microsoft Outlook and Word. These developments are also linked to the growing importance of web “platforms” (Gillespie, 2010), used to describe the vast set of services, programs, and media channels available through a single company’s shared infrastructure, like the interlinked ecosystems provided by Google or Facebook. Open source advocates including Richard Stallman have critiqued this model of software development as one in which “the server operator controls your computing” (Stallman, 2013), as it initially appears that the functionality and affordances of such software programs are even more locked down than in traditional closed-source programs, because the code does not even run on the user’s computer.

There has been substantial recent scholarship, commentary, and activism about various algorithmic systems in these platforms in a growing topic area focused on “the politics of algorithms” (Barocas et al., 2013). One common concern is with how centrally-hosted software services are filtered, ranked, indexed, and moderated by algorithmic systems – such as the many controversies and debates over Facebook’s filtered news feed. Understandably, much of this inquiry is focused on code that is designed, developed, and deployed by staff who work at the companies that own and operate such sites, who are assumed to have the exclusive authority to write the “code is law” (Lessig, 1999) that structures interactions on the site. For example, when e-mail clients were generally standalone desktop programs, spam filtering was typically a feature that ran inside a client and could be highly configurable by end-users. In contrast, many contemporary centrally-hosted platforms (like those run by Google or Facebook) filter content in ways that are typically “black boxed” to end-users. As those who advocate for “algorithmic accountability” (e.g. Diakopoulos, 2015) argue, even those with advanced expertise in software development may not be able to know how their own experiences of a site are being curated on their behalf, much less able to change how such curation of their news feeds or search results take place.

Bespoke code complicates this situation, as bots, browser extensions, and third-party tools are designed, developed, and deployed somewhat independently from a site’s servers and the people who own and administer them. Existing literature on individual cases of bespoke code emphasizes how such software can dramatically depart from (and even subvert) the design decisions, assumptions, and values of those who develop a server-side platform. For example, Turkopticon (Irani & Silberman, 2013), is a bespoke browser extension to the Amazon Mechanical Turk crowdsourcing service. Amazon has designed the site in such a way that employers are able to rate and review workers, but such an affordance is not reciprocated for workers to review employers. With the Turkopticon browser extension, workers are able to rate

and review employers, with these ratings aggregated on a third-party server and displayed by the browser extension. The Facebook Demetricator (Grosser, 2014) is another bespoke browser extension that removes all quantified counts from the social networking site, such as the number of replies or likes to posts or the number of friends a user has. As Grosser argues, the program is an explicit critique-in-practice of the ways in which “quantification prescribes social interaction on Facebook” (1). Through such bespoke code, software developers who do not have the ability to change how a software program operates are nevertheless able to extend or even subvert the assumptions that are embedded into server-side codebases.

### **1.3 IRC bots and the delegation of governance work**

One of the best historical cases of how bespoke code can transform the operation of a centrally-hosted digitally-mediated environment are the bots (or automated software agents) that have long been present in Internet Relay Chat (IRC) servers (Latzko-Toth, 2000). As Latzko-Toth argues, when IRC was initially released, both IRC server software and the server-client protocol were notoriously thin in terms of the features and affordances available to users and administrators. IRC was heavily extended through the development of bots, most notably adding administration and moderation features. A standard user account operated by an automated software agent could be made the administrator (or ‘op’) of an individual chat channel or a set of channels in an IRC network, and could then enforce particular algorithmically-defined rules. Such a feature was not supported in IRC servers in the way it was supported in the server-side software used to run later chat rooms – like the algorithmically-administered AOL chat rooms Lessig discussed (Lessig, 1999). Bots also provided various services to IRC users, including providing on-demand reference requests or linking a chat room with another kind of online environment. During the peak of IRC’s popularity, bots were part of the fabric of average, everyday life in many IRC channels, with struggles over norms, moderation, and leadership playing out in part through such automated software agents.

One of the more humorous examples of how bots were used to extend the functionality of IRC can be found in one of the top rated transcripts in the bash.org collection of IRC lore. An IRC user with the username Abstruse enters a channel for discussing Christianity, which uses two bots: Word\_of\_God, which will quote any bible verses on request (among other tasks), and an administrative bot that will kick any user who breaks certain rules, which includes a prohibition on swearing. Abstruse requests Word\_of\_God quote a bible verse containing the word “ass,” which leads to Word\_of\_God being immediately banned by the administrator bot. This demonstrates two aspects of bots as bespoke code: the extension of a digitally-mediated environment to support a new feature (quoting bible verses on request) and the delegation of governance work to an automated agent (banning users who swear).

The humorous intersection between these two issues speaks to the non-traditional way in which these features are implemented. Bots operate through user accounts rather than server-side code, which function in the server-side software in the same way than human users do. Unlike a moderation feature incorporated into server-side code, both the bible quoting and anti-swearing features would stop operating if the computer running these bots crashed, were disconnected from the IRC server, or if another administrator kicked the account. This means that in bot-based bespoke code, the ability to block or ban a human’s user account also becomes



an ability to turn off certain kinds of features, which are more traditionally limited to ‘server sovereigns.’

## **2. Bespoke code in Wikipedia**

### **2.1 Infrastructural inversions**

In Wikipedia, bespoke code extends almost every aspect of the encyclopedia project, to such an extent that it is as much of Wikipedia’s infrastructure as the core MediaWiki platform itself. I have found that Wikipedians rely on this bespoke code to such an extent that it often becomes invisible and sinks into the background – as scholars of infrastructure and philosophers of technology have long noted (Star & Ruhleder, 1996; Suchman, 1995; Heidegger, 1993). Such code is bound up in practices and issues across almost all areas of the encyclopedia project, including extending the mark-up language in which editors write articles, curating the discussions that take place about article content, reviewing contributions and reverting those identified as spam or vandalism, and supporting the behind-the-scenes administrative processes in which veteran Wikipedians set and enforce policies that govern participation in the project. Infrastructure typically only becomes visible through what Bowker and Star call an “infrastructural inversion” (Bowker & Star, 1999), a kind of gestalt shift in which infrastructures are brought to the foreground, which typically only occurs in such environments when they breakdown, malfunction, or become contested. As I discuss in the next section, the challenges that I and other MediaWiki sysadmins have faced when seeking to host their own Wikipedia-like wikis is one way in which such infrastructures can be inverted and brought from background to foreground.

#### **2.1.1 What is code in Wikipedia?**

A standard installation of MediaWiki, the software platform powering Wikipedia, has over 600,000 lines of code in about 900 files, mostly written in PHP and released under an open source license. It is easy for a Linux systems administrator to configure and install their own instance of MediaWiki, comparable to other platforms like Wordpress, Drupal, or Joomla. In a matter of minutes, a seasoned sysadmin can set up their own wiki and have a site that will look and feel like Wikipedia, at least on the surface. There will be wiki pages that users can collaboratively edit. The history of a page’s revisions can be accessed, and undesirable changes rolled back. Editors can communicate with each other using talk pages. Administrators can protect pages from editing and block problematic users. It’ll be a wiki, the kind of website that has come to stand in for an entire revolution in content creation, management, economics, and politics (for a critical analysis of this discourse, see Van Dijck & Nieborg, 2009 and Tkacz 2013).

However, as I and many other founders of their own MediaWiki-based sites quickly learn, many of the features and functionalities that are taken for granted in Wikipedia are nowhere to be found in a “stock” installation of MediaWiki. While Wikipedia does run on a version of MediaWiki, the version it runs is a highly-customized one, relying on a substantial amount of software code that does not even run on the same servers operated by the Wikimedia Foundation to host Wikipedia as a MediaWiki instance. By my estimate, there are at least ten

times the number of lines of code in software that runs alongside the server-side platform hosting Wikipedia, compared to the code in a stock version of MediaWiki. This code – some of which fundamentally changes how a wiki operates as a wiki – takes many forms, including third-party services, template scripts, user scripts, standalone tools, browser extensions, and fully-automated bots. This code is written in a multitude of programming languages, coded in a variety of environments, and is often executed on computers that are relatively independent from those run by the Wikimedia Foundation to host Wikipedia. Without this ‘extra’ code, those on a new wiki can find themselves quickly overrun by spammers, as spam filtering is one of the many features supported by highly-customizable bespoke code. Editors of stock MediaWiki sites are also not able to even leave the much celebrated “[citation needed]” tag in articles – an artifact and practice that has become a cultural icon of wiki-based collaboration.

One of the most notorious cases of breakdown for new wiki sysadmins are “infoboxes.” Wikipedians have long used bespoke code to support these infobox templates, which produce compact summary boxes typically placed in the top right of articles. They are a core feature Wikipedians use to coordinate editorial work (Ford, 2015). New sysadmins often come to forums, Q&A sites, and help desks to ask for help setting up their own wikis, and many want to know how to get Wikipedia-style infoboxes in their site. People who start their own wikis often do so because they have previously edited Wikipedia and found that they wanted a similar kind of way to support the work of collaboratively curating documents or other texts. Yet the layers of code which must be assembled and aligned in particular ways to get even a single infobox to work in a non-Wikipedia wiki go far beyond the already complex 600,000 lines of code in stock MediaWiki. As one MediaWiki sysadmin related his experience with infoboxes in a blog post:

If you have your own MediaWiki instance, you’ve probably thought they’d [infoboxes] be a nice thing to have, so maybe you copy and pasted the code from Wikipedia and then were surprised when it didn’t just magically work. Turns out that the infobox stuff is part of MediaWiki’s extensive Templating system, so first of all you need the templates. Sounds easy, right?

Well, no. You don’t just flip a switch or download a file, and when you do a search you might find [this article](#) which details a process that it says might take 60-90 minutes. I started looking into it and quickly got lost; you basically need to create a billion different Templates and do all sorts of weird stuff to get it to work.<sup>22</sup>

### 2.1.2 Contrasting stock MediaWiki and Wikipedia

I demonstrate this visually in figures 2 and 3, which contrast stock MediaWiki with the version of MediaWiki that Wikipedia runs. To produce these figures, I set up a new MediaWiki site on my personal web server using the latest stock version, then copied and pasted the text from the Wikipedia article on “Wikipedia” to a page on the new site. Such a comparison most

---

<sup>22</sup> <https://web.archive.org/web/20150606181630/http://trog.qgl.org/20110815/setting-up-infobox-templates-in-mediawiki/>



Figure 1: The Wikipedia article on “Wikipedia,” logged in under my account (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)



Figure 2: The same article, copied to a page on a stock version of MediaWiki (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

immediately shows how Wikipedia articles as Wikipedians have written them appear broken without this typically invisible infrastructure. Furthermore, the new site is not just ‘broken’ – it functions in particular ways but not others. A comparison of these two sites is a rich way to show how particular aspects of the more generic and general-purpose wiki software have been specifically modified by Wikipedians, as they worked out what it meant to use a wiki to host a project to collaboratively author and curate a general-purpose encyclopedia.

The tension between what it means for Wikipedia to be both a wiki- (a collaboratively authored set of hyperlinked documents) and a -pedia (a high quality, general purpose reference work) has long been identified as a core tension by researchers and Wikipedians who have watched the encyclopedia project develop over the decade and a half of its existence (Ayres, Matthews, & Yates, 2008; Reagle, 2010; Tkacz, 2015). Such tensions play out more visibly and explicitly in the many ongoing conflicts over the content of encyclopedia articles, as well as in conflicts over the project’s norms and meta-norms. Many scholars have studied these processes of norm formation and collective sensemaking in studying how Wikipedians debate the definition of a “reliable source” or the processes for determining whether articles should be temporarily protected from editing. Beyond these more immediately visible conflicts, the tension between Wikipedia as a wiki- and a -pedia can also be seen in how Wikipedians have modified what a wiki is and how it operates as they have developed a particular understanding of what a wiki-based encyclopedia is and ought to be. Each of the features that are made visible in this comparison had to be specifically added into Wikipedia’s instance of MediaWiki, as Wikipedians worked out their own particular understandings of how they wanted wiki-based collaboration to take place. Behind each feature is a moment in which someone decided to extend MediaWiki to support something like infoboxes, automatically formatted references, article quality ratings, editor-to-editor interactions, or IPA pronunciation notation. Sometimes these extensions and modifications were made through bespoke code, while other times they were made more directly through server-side code.

The most obvious difference in the two sites is the lack of formatting, layout, and reference elements that do not exist in my site, like infoboxes and references. Such features are implemented in templates, a MediaWiki feature in which editors write customized scripts in special wiki pages on the site, which are called as functions when left in the text of articles. A 2007 study found that there were over 100,000 templates in the English-language Wikipedia, and a query I ran on the Wikimedia Foundation’s analytics cluster in July 2015 found over 630,000 existing templates, with over 535,000 of them used in at least one wiki page. Templates that do not exist in my wiki are displayed as red links beginning with “Template:” and there are also many red links to articles that also do not exist on my wiki.

Beyond the article text, there are also a number of differences throughout the interface. The left-hand sidebar supports far more features and tools in Wikipedia’s version, such as the “Cite this page” feature not supported in my site. The top of the page indicates that my account has received a new message,<sup>23</sup> which in Wikipedia’s version of the site is in a streamlined notifications bar, rather than the blocky yellow banner – a project called “Flow” that seeks to

---

<sup>23</sup> This is a genuine unread message from another editor in Wikipedia, but I sent myself a message in my own site to demonstrate the difference.

improve communication between Wikipedians. The Wikipedia version has a banner advertising Wikimania 2015, the annual conference/convention of Wikipedians. This is presented through a feature called CentralNotice that supports targeted banners, which is also used for the annual fundraiser. The “cite this page” feature, Flow, and CentralNotice are PHP extensions that are not a part of “stock” MediaWiki, but do require server-side access to install.

There are also the “More”, “Page” and “TW” tabs at the top of the page, which are bespoke code – Javascript-based browser extensions – which I have installed to help me perform complex tasks within Wikipedia. Such tools are not enabled by default for all registered Wikipedia editors, as they must be specifically installed. One of these is as the “Copyright vio detector” (Figure 4), which queries search engines to try to find versions of the page in other pages on the web. Many of these tools are specific to Wikipedia’s particular specialized processes and would not make sense in other wikis, such as the ability to nominate the page for deletion under Wikipedia’s Articles for Deletion process. Finally, the clock in the top-right hand of the Wikipedia version is a live updating clock in Coordinated Universal Time (UTC) / Greenwich Mean Time (GMT), which is a tool that helps Wikipedians deal with the fact that timestamps for discussions in MediaWiki are in UTC, with no easy way for the server-side software to convert these to the local timezone of the editor.<sup>24</sup>



Figure 3: The sub-menus for one of the power tools (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

## 2.2 Bots in Wikipedia

The previous comparison between stock MediaWiki and Wikipedia reveals some backgrounded infrastructural elements in Wikipedia that have become a part of the project

<sup>24</sup> In an aspect of Wikipedia that will become highly relevant for a different reason in the next chapter, such timestamps cannot be easily automatically converted to the user’s local timezone. Wikipedia’s discussion spaces are the same kind of flat text files that anyone can edit as encyclopedia articles. This means that commenting in a discussion space involves editing the page, users must manually indent to indicate a threaded reply, for example. It also means that users must manually sign their comments to leave their username and timestamp so that others can know when the comment was made and who made it without having to look through the revision history. MediaWiki supports a special keyword, where leaving four tildes in a row (“~~~~”) will be converted to text containing the user’s signature and a timestamp in UTC.

despite being implemented through bespoke code, rather than server-side code. What cannot be seen from this comparison is one of the biggest cases where bespoke code plays out in MediaWiki versus Wikipedia: bots, or automated software agents that operate in the capacity of a user account, making requests to a server in the same way that a web browser does. People who do not have access to modify the source code hosting a site are developing and deploying bots to automate particular tasks in and across a variety of sites, and not just in Wikipedia. One 2013 industry report estimated that 61.5% of all web traffic was from “non-human bots” (Zeifman, 2013). However, the way bots are deployed as bespoke code in Wikipedia differs from the longstanding and dominant use of automated software agents as scrapers by search engines to index the web, or the malicious bots that typically pose as humans in order to send spam, get people to divulge personal information, or attempt to penetrate private systems (Boshmaf, Muslukhov, Beznosov, & Ripeanu, 2011a). Such bots also differ from the “chatbots” or “chatterbots” that seek to mimic human interaction using natural language processing for a variety of purposes, including the famous therapist bot ELIZA (Weizenbaum, 1966) and the many automated personal assistants that are often imagined as “butlers” (Suchman, 2007).

Bots are ubiquitous in Wikipedia and have long been explicitly delegated key editorial and administrative tasks involved in the encyclopedia project. There are 410 user accounts<sup>25</sup> that have been authorized to run 1,848 different automated tasks<sup>26</sup> on the English-language Wikipedia as of 20 April 2015. The work that these bots have been delegated extends to many aspects of the project, such as writing new encyclopedia articles – like RamBot, which doubled the number of articles in Wikipedia in 2002 when it created an article for every U.S. city and town using public domain census data (Kennedy, 2010; Lih, 2009). However, bots also perform less visible tasks behind the scenes of the encyclopedia project, as a number of researchers have investigated (Geiger, 2009, 2011; Niederer & Van Dijck, 2010; Halfaker & Riedl, 2012; Müller-Birn, Dobusch, & Herbsleb, 2013). Bots perform tasks related to information organization, coordination work, articulation work, quality control work, dispute resolution, normative enforcement, newcomer socialization, user interface enhancements, and administrative processes. Without bots, Wikipedia would look like a quite different place, due to the roles that these bots play in the continual and routine operation of the encyclopedia project.

Like the bespoke code like that supports infoboxes or references, bots can have profound consequences in extending how the site supports certain kinds of interactions, having the kind of governmental “code is law” (Lessig, 1999) consequences that are traditionally only available to the sysadmins who have access to the server-side code hosting the site. Many bots in Wikipedia perform the same kind of moderation work of content that was seen in example of the anti-swearing IRC bot, but with a key difference. Because many actions in Wikipedia involve editing pages that “anyone can edit,” bots in Wikipedia can be delegated specific kinds of governance work without needing to even have the same kind of administrative account privileges needed to block an account. For example, anti-spam bots review edits made in near real time and automatically revert edits identified as spam, using the same ability to edit articles that are available to human editors. Such automated moderation of contributions can be highly

---

<sup>25</sup> List at <http://en.wikipedia.org/wiki/Special:ListUsers/bot>

<sup>26</sup> List at [http://en.wikipedia.org/wiki/Category:Approved\\_Wikipedia\\_bot\\_requests\\_for\\_approval](http://en.wikipedia.org/wiki/Category:Approved_Wikipedia_bot_requests_for_approval)

contentious, and there have been countless conflicts over what encyclopedia articles ought to look like or how editors ought to interact with each other that play out through bots. Furthermore, bots extend the functionality and affordances of a site in ways that go beyond automated moderation of content, such as the “clerk” bots that turn standard wiki pages into sophisticated queues for making particular kinds of commonly reoccurring decisions.

As I argue throughout this dissertation, bots and other bespoke code play such a role in how Wikipedia operates as a site of cultural production that many issues in the site cannot be fully studied without understanding this different kind of software development. On one level, bots and bespoke code are important because of the substantial impacts they have on both the content of encyclopedia articles and how Wikipedians interact with each other. Such code enacts, supports, structures, and enforces particular understandings of what an encyclopedia article ought to look like and how editors ought to collaborate when editing an encyclopedia. Yet at a more fundamental level, bots should not be seen as mere force multipliers that extend the will of individual bot developers – as bots are sometimes characterized in rise-of-the-machine narratives (which I further discuss and critique in the next section). Wikipedia’s bespoke software developers – who write and deploy code that runs outside of the standard server-side platform – have been an active part of the Wikipedia project almost since its inception. While individual developers and bots have come and gone over the almost fifteen years of Wikipedia’s existence, bots and bespoke code have long been a core mode in which Wikipedians collectively work out what kinds of tasks they think ought to be done. The tasks that bots perform are rarely made in isolation. The ways in which bots have been delegated tasks at any given point in Wikipedia’s history speaks to the broader issues and concerns facing the encyclopedia project as a whole. Debates about national varieties of English play out in debates about spellchecking bots, debates between the factions of inclusionists and deletionists play out in debates about bots that help curate contributions, and debates about participation and the demographic gaps in Wikipedia’s contributor base play out in debates about bots that interact with newcomers.

### **3. Theoretical issues around studying bespoke code**

#### **3.1. Motivation: Rethinking “Who What is in Control of Wikipedia?”**

In this next section, I first discuss the importance of taking such an in-the-making approach by briefly showing how I came to shift my own research agenda around bots in Wikipedia. When I first started researching Wikipedia in 2006, I didn’t set out to find bots, bespoke code, or even issues around software code at all. Like many others, I was interested in understanding how Wikipedia could possibly work, given that it seemed to lack the traditional organizational structures that most other institutions of knowledge production and content curation had. Yet alongside all the norms, policies, roles, committees, discourses, routines, and bureaucracies that made up what has been called “the hidden order of Wikipedia” (Wattenberg et al., 2007), I encountered something else: hundreds of automated software agents, diligently editing behind the scenes to make some seemingly miraculous aspect of Wikipedia work.

Unlike the malicious bots used for spamming, identity theft, or denial of service attacks that typically masquerade as humans (Boshmaf, Muslukhov, Beznosov, & Ripeanu, 2011b),

these bots were developed explicitly as bots, by Wikipedians and for Wikipedians, doing work that was typically recognized as important for the project's goal of writing and curating a wiki-based encyclopedia. Even the cases in which these bots performed a controversial or unintentionally-destructive act, the response by both other bot developers and non-developers in the encyclopedia project was generally swift and severe. While much has been written about norms, practices, motivations, and beliefs of the more exclusively human members of the community<sup>27</sup> of editors responsible for writing Wikipedia's articles (Kuznetsov, 2006; Oreg & Nov, 2008; van Liere & Fung, 2011; Welser et al., 2011; Yang & Lai, 2010), bots and their developers are just as much a part of Wikipedia as those who worked to add scholarly references to articles, upload photos of World War II era aircraft, remove errors and spam, or resolve heated content disputes. In fact, bots often assist heavily in these tasks, albeit in ways that function more as "clerks" – as some of their developers call them.

When I began to publish my first findings on Wikipedia bots in 2009, I was largely limited to documenting their prevalence using statistical methods and a handful of striking examples. Two findings – that bots made 16.33% of all edits to the English-language Wikipedia and that 22 of the top 30 editors by number of edits were bots (Geiger, 2009) – were quickly taken up by many people writing about Wikipedia in both academic and popular venues. While I was grateful for the unexpected attention of these findings, I soon realized that a strong narrative was being crafted around them, one that used these statistics and cases to argue that bots are 'taking over' Wikipedia. This should not be surprising: automation is often cast as a zero-sum game in which 'the machine' steals agency from humans, which is alternatively celebrated or lamented. This positioning of artificial agents against human agents is an age-old trope found in morality tales and social commentaries across cultures and generations. It can be seen found in the tragic myths of golems and Shelley's *Frankenstein*, in the satire of Chaplin's *Modern Times*, and in the dystopic AI-dominated futures of *Terminator* and *The Matrix* (Covino, 1996; Henry, 2014; Waldby, 2004).

I admit that I played into this trope as well, finding it to be a compelling rhetorical device. I opened up a number of presentations on Wikipedia bots by initially focusing on the question "Who is in control of Wikipedia?" – referring to questions many people were asking about the demographic makeup of Wikipedia's contributor base, either in aggregate or in terms of who held positions of authority or power in the project – then striking out "who" and replacing it with "what."

---

<sup>27</sup> In addition to being a sociological term with a long lineage, the term "community" is also an emic term used by Wikipedians to collectively describe themselves (and by corollary, exclude those who are not considered members). The two are linked for those who use these terms: Wikipedians are members of the community, and the community is made up of Wikipedians. The meta page on "Wikipedia:Community" has long redirected to the meta-level page on "Wikipedia:Wikipedians." I have found that Wikipedians frequently position membership in "the community" as demanding that one edit enwiki in the best interests of "the encyclopedia," placing the goals of having a high quality, freely-licensed, and collectively-edited encyclopedia above one's own personal points of view or self-interest. Wikipedians have several heuristics to help quickly judge whether a user account editing enwiki belongs to a Wikipedian or not, often treating those who are not considered Wikipedians differently.



The narrative around how bots in Wikipedia have taken over the work of cultural production from humans continues today, with the ‘rise of bots’ alternatively celebrated or lamented. Ray Kurzweil, famous for his optimistic advocacy of artificial intelligence, wrote a blog post in 2014 titled “Are bots taking over Wikipedia?”, which celebrated the bots for “pick[ing] up the slack” as the encyclopedia allegedly grew too large for humans to maintain it. A 2014 post on TheDailyDot was titled “Bots have conquered Wikipedia – and that’s a good thing” (Sankin, 2014), focusing on anti-spam and counter-vandalism bots. Other characterizations of bots in and out of Wikipedia are far more negative, such as a Guardian article titled “How bots are taking over the world” (O’Hara & Mason, 2012). The article cites my own research on the prevalence of bots in Wikipedia as part of a broader review of how “life is being manipulated by internet algorithms” – from phishing bots seeking to steal our personal information to financial trading algorithms to the ranking and filtering of stories Facebook’s news feed, Google’s search engine results, or highly-automated governmental surveillance programs. The article closes with a common conclusion about the loss of control, in which a sharp divide is drawn between bots and people. O’Hara and Mason discuss this through the case of someone who had their identity stolen by (someone who ran) a botnet:

When Ronson [who had his identity stolen] looks for the people trying to control the internet, he's looking in the right place, but at the wrong species. The internet is increasingly becoming a post-user environment, regulated by something much more uncontrollable than humans. (O’Hara & Mason, 2012)

### **3.2: Theorizing the separation and co-constitution of humans and automata**

Both critical and celebratory versions of this narrative often position the emergence of bots in relation to the inadequacies of humans: for champions like Kurzweil, bots are doing work in Wikipedia that humans can’t or won’t do, while for critics like O’Hara and Mason, bots have taken control away from humans. This characterization exists in scholarship and commentary about algorithmic systems outside of Wikipedia as well, such as in debates over the automated filtering of Facebook’s news feed (see Bakshy, Messing, and Adamic 2015 versus Tufecki 2015) or surveillance-based predictive policing systems (Doctorow, 2014). Either positive or negative, there is often an assumed separation between humans and algorithmic agents, which ignores the roles of the people who design, develop, and deploy them -- a group that often includes but extends beyond the bot’s programmer. Lucy Suchman (2007) argues that this separation between humans and artificial intelligence is not only central to discourses of artificial intelligence, but bound up in a broader issue around the erasure of work: “discourses of information technology have tended to erase the human labor that continues to be involved in technological production, implementation, maintenance, and the like” (217).

In seeing the technological artefact or automaton as something that is separate from those who develop, design, build, and maintain them, we can easily slip into a mindset where the human work involved is displaced and removed from view – at least for a certain set of people. For example, a photocopier is a way to automatically reproduce documents far faster than the typing pools that were major organizational units in early 20<sup>th</sup> century corporations. Like all technology, photocopiers must be designed, built, maintained, and repaired, which involves a substantial amount of work – as Julian Orr’s ethnography of Xerox technicians in

the 1980s illustrates (Orr, 1996). What the photocopier enables is as much of delegation of work from inside of the company to outside the company as it is a delegation of work from people to machines. With Xerox's business model, a substantial amount of the work needed to reproduce documents could be outsourced to an external firm. Orr notes how one of Xerox's priorities for achieving market dominance was in letting companies treat their photocopiers as a black box. Companies didn't buy a photocopier, they leased it, and leases included access to an army of roving technicians, who provided an unprecedented amount of support to keep these machines operating. The total work required to reproduce a document did likely decrease from typing pools to photocopiers – but rather than eliminating human work, it displaced it in a way that made such work less of a concern for those in offices who relied on the products of that work.

Similarly, Cowan argues in *More Work for Mother* (Cowan, 1983) that the many mechanical and electrical household technologies which achieved market dominance in the 20<sup>th</sup> century were long celebrated as labor-saving devices, freeing up time for married women to become part of a growing leisure class. However, she found that as technologies like the washing machine and the vacuum were introduced, married women actually spent slightly more time working in their homes. Cowan argues that such women were certainly more productive and efficient, but instead of keeping up with the same standards of living and using the remaining time for leisure, they were expected to keep up with higher and higher standards. The utopian narrative that labor-saving devices will lead to the expansion of the leisure class – and even a post-scarcity economy in which no one has to work – is heard in almost every major wave of technological innovation. Yet such visions of the future are continually empirically disproven, and Suchman quotes Chasin (1995) in arguing that they serve to silently reinforce “the idea that a service class of being(s) is proper and even necessary,” which is the idea “upon which the myth of a constantly expanding middle class depends” (Suchman 2007, p. 93).

### **3.3 Bespoke code as a way to call attention to the materiality of software**

Stacey and Suchman (2012) argue that when thinking about the sociality of automata, it is not productive to draw stark divisions between humans and artificial agents. Instead, Stacey and Suchman draw on developments in science and technology studies (STS), particularly work extending Donna Haraway's theorizing of the cyborg (Haraway, 1991) as a metaphor for understanding the co-mingling of agency between humans and artifacts.

The aim of a critical engagement with contemporary projects in making autonomous machines is not to retain some special qualities to the human, nor to oppose the prospects of a future filled with animated things. Recent writings in STS, rather, emphasize the inseparability of the human from the artifactual, and render the relation as a more irreducible and intimate one. With respect to automata, the approach is, first, to slow down the rhetorics of life-like machines and to attend closely to the material practices of their realization ...to the contingent and ongoing labours involved in sustaining the agencies of things. (Stacey & Suchman, 2012)

Stacey and Suchman's call to focus on the materiality, contingency, and continual labor involved in sustaining software is aligned with a long set of approaches dedicated to the study of technology and society "in the making." Such approaches are generally used to respond to depictions of technology that champion or critique machines for having the same agency as humans, yet pass over all the human work that is needed for such machines to have such agency. This has certainly been the case with bots not only in Wikipedia, but broader discourses around "the politics of algorithms," in which algorithmic agents or systems that make crucial decisions or judgments are sometimes discussed as if they are autonomous rogue agents, governed not by humans but by their own computational logic run amok. In response, scholars like Gillespie (2014) and Seaver (2013) advocate critical social science scholarship into algorithmic systems, but argue against this tendency to see such systems as emerging out of nowhere: "A sociological analysis must not conceive of algorithms as abstract, technical achievements, but must unpack the warm human and institutional choices that lie behind these cold mechanisms" (Gillespie, 2014). Such a line of inquiry has examined not only specific abstracted algorithmic procedures, but also the concrete, historically-contingent ways in which those abstracted algorithmic procedures are designed, developed, and deployed in the world. This "in the making" approach to software extends the even more accepted notions that it is important to study software development as a socio-cultural or organizational practice (e.g. Crowston & Howison, 2005; Mockus, Fielding, & Herbsleb, 2002) or to study the economic, political, social, cultural, psychological, and organizational impacts of software on our world (Kelty, 2008).

I further critique the assumed separation of humans and automata through focusing on bespoke code, which calls attention to a material distinction about where software code is run. Bespoke code runs separately from the server-side codebases hosting a digitally-mediated environment, which has consequences for how such software agents operate in a community that inhabits a centrally-hosted digital environment. Bespoke code is a vivid reminder that what software is as software should not be divorced from the conditions under which it is developed and deployed. This is a materialist argument, opposing the "trope of immateriality" (Blanchette, 2011: 3), a discourse that alternatively celebrates or laments the disembodied nature of information technology. As Blanchette reviews, the supposed immateriality of mediated technology is nothing new – it was used to describe the telegraph with the same celebratory rhetoric that can be found in contemporary commentary on new technologies. Yet such an assumption is important to interrogate, given the roles that such technologies play as infrastructure, supporting particular kinds of practices better than others. Blanchette cites Hayles in arguing that this trope, in which digital information is proclaimed to be "free from the material constraints that govern the material world" (Hayles, 1999: 13), is not just a casual metaphor of technologists. Rather, the hard distinction between "bits versus atoms" (e.g. Negroponte, 1995) that imagine information technology as supporting "virtual communities" or "virtual organizations" can obscure the material conditions and labor which make it possible for people to "seamlessly" interact.

### **3.4 Code as law-in-the-making**

Bespoke code provides a noticeably different set of cases for exploring how, as Lawrence Lessig famously argued, "code is law" (Lessig, 1999). Rather than code being law in the sense that software is a way a server sovereign enforces particular power relations on people

who are subject to their rule, my studies of bots have given me a different angle on Lessig's phrase. The software code of the bots I studied was designed, developed, and deployed by people who did not have access to modify server-side code of these centrally-hosted environments. The code of bots was not law in the sense of a rigid set of already enacted rules that everyone must follow, but rather it was law more in the sense of public policy – a way of deciding high-level issues in the project. For example, spellchecking bots may seem like an obvious and uncontroversial task to perform, until the question is asked about what national variety of English will be used for its dictionary. And this issue raised by spellchecking bots raises many high-level questions about what Wikipedia is as an encyclopedia and a global community of editors. I do not wish to imply that all of Wikipedia's high-level normative issues are resolved exclusively through bots. Rather, I emphasize that bots are one of many core ways in which Wikipedians make such decisions about what Wikipedia is and ought to be. Code (and descriptions of algorithms) are a more expressive medium in which people sought to build a common abstract understanding that represented their ideas about what kind of a world they want to live in. This is a stark difference compared to the kind of code-as-domination that Lessig discusses – such as his argument that the code-as-law passed by server sovereigns is even more potentially totalitarian than those by traditional governments because developers can change the “laws of nature” (Lessig, 1999, p. 70).

The kinds of negotiations I continually observed between bot developers and others (including other bot developers, non-developer users, and server sovereigns) reminded me of the kind of gestalt shift Science and Technology Studies scholars often discuss between taking something as “ready-made” versus “in-the-making” (Callon, 1987; Latour, 1987; Shapin, 1992). In Science and Technology Studies, “in-the-making” approaches departed from the previously dominant theoretical literatures which emphasized the impacts of technology on society and showed how “artifacts have politics” (Winner, 1986) in their deployment, diffusion, and use. An in-the-making approach takes place before science or technology becomes the kind of stable, coherently existing entity where it can have this kind of ‘billiard ball’ impact on society. Studies using this approach often use ethnographic or historical methods to observe the development of science or technology as it is actively constructed in a particular context. This approach critiques competing literatures that present technological or scientific development as linear progress, instead emphasizing the highly contingent and often unpredictable paths that the development of technological systems and systems of knowledge production take. Scholars in this area take many strategies to emphasize the constructivist lesson that “things could have been otherwise” (Hacking, 2000), paying attention to the paths not taken (Bijker, 1995), rejection and refusal (Novek, 2002), beta versions (Neff & Stark, 2003), catastrophic failures (Vaughan, 1997), mundane failures (Latour, 1996), breakdowns (Star, 1999), repair (Jackson, 2014; Orr, 1996), invisible work (Star & Strauss, 1999), unintended uses (Oudsshoorn & Pinch, 2003), and unintended users (Burrell, 2012). These are all windows into an understanding of the “practical politics” (Bowker & Star, 1999, p. 44) that are a core aspect of the design, development, and deployment of technological systems and systems of knowledge production

### **3.5. Related literature on the materiality of software**

Critiques of seeing code as the essence of software are far from new, and my focus on bespoke code draws on and synthesizes a wide range of scholarship. Scholars have examined

software not only through its code and algorithmic routines, but through the practices and meetings of network operators (Mathew & Cheshire, 2010), the magnetic storage mechanisms of hard drives (Kirschenbaum, 2008), the copyright licenses of open source software communities (Kelty, 2008), clerical and support staff in datacenters (Ribes et al, 2013), the role of Internet cafes and social clubs in Ghana (Burrell, 2012) and hackerspaces in China (Lindtner & Li, 2012), and the roles of universities, businesses, and government agencies of Rio de Janeiro (Takhteyev, 2012) or the San Francisco Bay Area (Saxenian, 2006).

Furthermore, ever since Marx's socio-political analysis of factory machinery and other engines and artifacts of capitalism (Marx, 1973; c.f. MacKenzie, 1996), scholars have interrogated the infrastructures, artifacts, and work practices that are taken for granted in a variety of social institutions: prisons (Foucault, 1977), museums and art galleries (Becker, 1982; Star & Griesemer, 1989), hospitals (Garfinkel, 1967a), scientific research (Latour & Woolgar, 1979; Shapin & Schaffer, 1985), public infrastructure (Winner, 1986), economic markets (Mackenzie, 2006), and organizations and firms (Orlikowski & Scott, 2008), to name a few. These institutions and ideologies are supported and configured in particular ways and not others through specific infrastructures, artifacts, people, and practices. Taking an in-the-making approach involves focusing on the role of these often taken-for-granted, behind-the-scenes apparatuses in making possible a particular version of the world in which we live. Just as the legal system is more than the text of laws and precedent (Latour, 2009), software systems are more than the text of code.

My argument is also related to a long line of scholarship on "virtual" or "online communities" which have critiqued the assumption that there is a strong boundary between the "online" and the "offline" world (Baym, 1995; Jurgenson, 2012). To problematize this tendency, many scholars are self-reflexively using different approaches to counter these metaphors of bounded spatiality. These include Burrell's reconceptualization of the "fieldsite as network" (Burrell, 2009) in her ethnography of youth and Internet cafes in Ghana and Beaulieu's shifting from "co-location to co-presence" (Beaulieu, 2010) to better capture the range of multi-faceted interactions taking place in a given organization or community. Similarly, scholars in organizational studies have critiqued the idea of a "virtual organization" (Fiol & O'Connor, 2005; O'Leary, Cummings, & O'Leary, 2007), instead emphasizing how co-located teams and workplaces can rely just as much on mediated interactions as those in distributed teams and workplaces. I argue that in a similar way, a site like Wikipedia is also not best understood by making sharp distinctions between the online and offline or the virtual and real life. Many issues about how Wikipedia works as a project that seeks to summarize "the sum total of human knowledge" are found in and around the infrastructures that host Wikipedia as a software platform. To understand how Wikipedia works to support this goal, I argue that it should be studied as a networked assemblage, something that is performed and made present in various materially-existing settings and contexts. And as I show across this dissertation, when we pay attention to these material infrastructures, they tell us much about how and why Wikipedia as a project works in the way that it does. Wikipedia's operation is often attributed to an empirically underspecified "wisdom of the crowds" (Surowiecki, 2004), but as Niederer & Van Dijck argue (Niederer & Van Dijck, 2010), there are specific ways that Wikipedians have built Wikipedia to support this particular form of collective knowledge production.

### 3.6 Unpacking the black box beyond revealing source code

These issues around the materiality of code play out in a prevalent assumption among scholars and commentators who are concerned with ‘the politics of algorithms’: the idea that to best understand how an algorithmic system operates, one needs to gain access to (or reverse engineer) the source code that constitutes the essence of “the algorithm.” Seaver notes this issue in his critique of some versions of the contemporary “critical algorithms studies” literature (Seaver 2013). He begins this critique by acknowledging that those interested in the politics of algorithms can certainly find many interesting issues in looking at specific lines of code that turns inputs into outputs. I am similarly supportive of scholarship, journalism, and activism in the area of “algorithmic accountability” (e.g. Diakopoulos, 2015) which uses a variety of approaches to determine what criteria, evidence, and processes are used in algorithmically deciding what messages are marked as spam, what posts are filtered out of news feeds, what applicants are denied loans, or what individuals are identified as suspicious by law enforcement, for example. However, Seaver argues that it is also important to understand the broader “algorithmic systems” in which they operate, which includes both phenomena that are typically characterized as purely ‘social’ and ‘technical’:

The use of phrases like “the Google algorithm” or “the Facebook algorithm” should not fool us into thinking that our objects are simple, deterministic black boxes that need only to be opened. These algorithmic systems are not standalone little boxes, but massive, networked ones with hundreds of hands reaching into them, tweaking and tuning, swapping out parts and experimenting with new arrangements. If we care about the logic of these systems, we need to pay attention to more than the logic and control associated with singular algorithms. We need to examine the logic that guides the hands, picking certain algorithms rather than others, choosing particular representations of data, and translating ideas into code. (Seaver, 2013)

Using spam filtering as an example, it is important to understand that the same algorithmically-specified procedure for separating spam from not-spam in Wikipedia articles could operate quite differently depending on how the source code implementing such a classifier was deployed. One way would be a server-side extension to MediaWiki that runs when users submit edits, returning an error and refusing to commit the edit if the new version of the page contains spam text. A second way would involve a browser-based extension that examines all paragraphs in an article and does not display those containing spam text, which only changes the experience for those who have installed it. A third way would be a bot that is operated by a volunteer which examines recently made edits and reverts those containing spam text, which is visible in the page’s revision history. The first way involves a more typical ‘server sovereign’ approach compared to the second and third ways, which are both bespoke, but those two modes operate differently from each other as well.

Despite using the same core procedures for identifying spam, these three cases involve three different kinds of relationships between a software developer and the people for whom the software is designed, developed, and deployed. Multiple versions of the browser-based spam filter extension can peacefully co-exist using incommensurable spam filtering algorithms, which would mean that the people who used these different filters would have different

experiences of reading Wikipedia. Multiple anti-spam bots that enact their spam filters through editing articles can co-exist with different spam filtering algorithms as well, but every edit identified by any of the bots are removed from everyone's experience of reading Wikipedia. These are different ways in which the same classifiers can put into practice, and the material conditions between them are important.

## 4. About a bot

This chapter has synthesized a wide set of literatures about the social study of software code. I argued that what it means to unpack or open up the black box of the algorithmic systems that operate in a site like Wikipedia should not be thought of as just an exercise in parsing through all the server-side code – or even all the bespoke code that runs alongside it. Reading code can be quite revealing, but it requires a relational, networked, infrastructural understanding of code and how it is situated in many different kinds of spaces. In this next section, I demonstrate this in a set of personal vignettes relating several different ways of remembering my own personal experience as a bot developer in Wikipedia, which give quite different ways of understanding what a bot is and how it operates. The purpose of these vignettes is to expand the analytical frame of what a bot is, going beyond seeing source code or even the software agent powered by source code. I designed, developed, and deployed a bot that performed a small but useful task in Wikipedia, which ran for almost a year. I detail the processes that were required for me to get this bot running and approved by Wikipedia’s administrative apparatus. However, I was running the bot from my personal computer in my shared studio apartment, and when personal issues led to upheaval in my life, I was ultimately unable to keep the bot operating. This illustrates how the work of bot development is not just writing code, but also doing work ensuring that the bot is able to keep performing its delegated task – from negotiating with administrators to keeping its server online.

### 4.1 AfDStatBot: My first bot

AfDStatBot was the first Wikipedia bot I built, which operated in 2008-9 when I was studying Wikipedia’s Articles for Deletion (AfD) processes. Hundreds of Wikipedia articles are nominated for deletion every day, and editors discuss whether or not each one should be kept or deleted. AfDStatBot was created for my quantitative research purposes, to generate statistics on these deletion discussions in real time. However, it also served a few secondary purposes, including giving this data back to the community I was also ethnographically studying. I could say that this was part of my participant-observation as a Wikipedian, but it just made sense that if I was capturing and analyzing this information, I ought to use those resources to make the lives of other Wikipedians easier. It’s just what you do. I had seen hundreds of these unofficial bots, tools, and scripts that Wikipedians developed for themselves and each other, supporting specific tasks and practices. I thought I had something to contribute. I really wanted to contribute. So I did.

AfDStatBot was one of many minor bot-based features in the English-language Wikipedia. It generated a near real-time noticeboard of active and recently closed deletion discussions (Figure 4), with statistics such as how long the discussion had been open, how many Wikipedians were in favor of keeping vs. deleting the article, when the last comment was made, and so on. Then for the few who opted in, it curated personalized watchlists that helped Wikipedians keep track of all the deletion discussions they had personally participated in. Finally, in what was more of an exercise to see if I could do it, the bot posted to Twitter (@WikiWars) every time it saw a new article nominated for deletion (Figure 5). It wasn’t the most popular or important bot on Wikipedia by far — most of the bot’s followers on Twitter were random auto-follow bots — but it did provide a service for a niche audience. A few years



later, another Wikipedian named would build a much better bot for these tasks, Snotbot, which is still in operation today.

Recently Closed AfDs [\[ edit source | edit \]](#)

Article	Result	Nomination date	Close date	Participants	Nominator	Closer
<a href="#">Paradise (Not for Me)</a>	delete	2009-02-07 13:29:35	2009-05-31 20:27:28	4	<a href="#">Paul75</a>	<a href="#">Juliancolton</a>
<a href="#">Just Like That (song)</a>	no consensus	2009-02-09 03:20:53	2009-04-30 02:46:35	5	<a href="#">Paul75</a>	<a href="#">MBisanz</a>
<a href="#">Tomorrow is Today (song)</a>	keep	2009-02-17 03:39:56	2009-04-27 00:05:34	8	<a href="#">TenPoundHammer</a>	<a href="#">Ron Ritzman</a>
<a href="#">Punkradiocast.com</a>	keep	2009-03-01 03:30:24	2009-04-25 01:10:55	8	<a href="#">Rtphokie</a>	<a href="#">Lankiveil</a>

Figure 4: AfDStatBot's list of recently closed AfDs. (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)



Figure 5: The Twitter feed AfDStatBot maintained, tweeting every new AfD debate (text © the author, Wikipedia logo © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

## 4.2 A legitimate alternative account

If you want to know about AfDStatBot, you might start with the bot's userpage on Wikipedia (User:AfDStatBot, Figure 6). Profiles take many different forms, but they have long been the spaces in which software platforms represent users (Brubaker & Hayes, 2011). In Wikipedia, these "user pages" are technically little more than special wiki pages, meaning that anyone can (but not necessarily *should*) edit anyone else's profile — as I typically did for AfDStatBot. Like most bot userpages, there are several curious elements that indicate this profile is different from that of most human Wikipedia editors. There is a large, red power button that administrators can use to stop the bot from operating if it begins to malfunction. That's not a formal requirement, but it is standard practice in Wikipedia bot development. There are also warning templates at the top that tell the reader that the account is a bot operated by me, and that it is a "legitimate alternative account" – not an unauthorized "sockpuppet."

In a single line in an infobox on the righthand side of my bot's user page, there is a link to a formal approval from the Bot Approval Group, which I had to petition before legitimately running my bot. If we think of bots as laws like in Lessig's "code is law" argument, then this approval perhaps captures another side of what this bot was – although it emphasizes bots as lawmaking, rather than law as already-enacted power relations. In order to be accepted as a bot operator and for my bot to be accepted as a legitimate bot, I had to translate the bot into text that others could come to understand and evaluate. If I ran a bot without approval, my bot may have been discovered by humans (or anti-bot bots) and both I and the bot could end up banned from Wikipedia. So this code-as-law had to be approved by the Bot Approvals Group in a discussion (Figure 7), but this didn't take long – although when my proposed rate of editing for updating the statistics pages was deemed too high, I lowered it. If my bot's task was more controversial, like a bot that would remove all images without proper copyright documentation from every article, then there may have been more discussion, but there was no controversy that followed. All of this was recorded, captured, and preserved for a number of socio-technical ends, serving to represent the intent of Wikipedia's administrative apparatus, such that it could be easily linked to from an infobox on my bot's userpage.

From Wikipedia, the free encyclopedia



This user account is a **bot** operated by [Staeiou](#) ([talk](#)). It is a [legitimate alternative account](#), used to make repetitive automated or semi-automated edits that would be extremely tedious to do manually. The bot is currently inactive but [retains the approval](#) of the community.

## Emergency bot shutoff button



**Administrators:** Use this button if the bot is malfunctioning. ([direct link](#))

Non-administrators can [report](#) a malfunctioning bot to [Wikipedia Administrators' noticeboard/Incidents](#).

This bot posts new and closed AfDs on [Twitter](#) [@](#) and maintains a list of currently active and recently-closed AfDs at [User:AfDStatBot/AfDStats](#), as well as some statistics. It also informs AfD participants in two ways: first, leaving a talk page message for nominators when their AfD is closed, and second, generating an up-to-date list of all the open and recently closed AfDs a user has participated in. These features are by request/opt-in only.

<b>Contents</b> <a href="#">[hide]</a>
<a href="#">1 Editing</a>
<a href="#">1.1 AfDStats</a>
<a href="#">1.2 User Notifications and Lists</a>
<a href="#">2 Algorithm</a>

### AfDStatBot

**This user is a bot**

([talk](#) · [contribs](#))

<b>Operator</b>	<a href="#">Staeiou</a>
<b>Author</b>	<a href="#">Staeiou</a>
<b>Approved?</b>	Yes, <a href="#">RFBA</a>
<b>Flagged?</b>	[Yes <a href="http://en.wikipedia.org/w/index.php?title=Special%3AListUsers&amp;username=AfDStatBot&amp;qgroup=bot&amp;limit=1">http://en.wikipedia.org/w/index.php?title=Special%3AListUsers&amp;username=AfDStatBot&amp;qgroup=bot&amp;limit=1</a> <a href="#">↗</a> ]
<b>Task(s)</b>	Generates lists of currently open and recently closed AfDs, and lists and notifications for AfDers
<b>Edit rate</b>	12 edits/minute max (5 seconds between edits)
<b>Edit period(s)</b>	Indefinitely

Figure 6: AfDStatBot's user page (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

perhaps have some discussion about the proposed features. For example, would it be better to give all nominators a talk page message when their AfDs are closed unless they opt-out, or should I keep it request-only? Thanks! [Stu](#) ([aeiou](#)) [Im Researching Wikipedia](#) 18:04, 9 February 2009 (UTC)

Updating the tables every 10 minutes seems rather excessive, especially if its going to be generating such lists for multiple people. [Mr. Z-man](#) 22:54, 10 February 2009 (UTC)

On average, an AfD is opened or closed every ten to fifteen minutes, which is why I set that time interval. And the personalized lists only update when each user participates a new AfD or when one of the AfDs they participated in closes. While it could theoretically update for each user every ten minutes, it most likely wouldn't. Even the most active AfDer right now, MGM, has only participated in about 10% of the open AfDs (58/559) and wouldn't get an update every ten minutes. I'll run some simulations with the ten most active AfDers over the next 24 hours, and I'll let you know how many times the bot would have edited had they requested userpage updates. [Stu](#) ([aeiou](#)) [Im Researching Wikipedia](#) 18:43, 11 February 2009 (UTC)

My point is that 2 lists + updates every 10 minutes == 8640 edits every month to change a few numbers; that's over a hundred thousand edits per year. Such frequent updates should really be reserved for cases where up to the minute information really matters. As this is mainly just statistics, I don't see why [User:AfDStatBot/AfDStatsTotal](#) really *needs* more than 1 update per day, though once every couple hours would be fine. [Mr. Z-man](#) 22:54, 11 February 2009 (UTC)

Agreed 100% on [User:AfDStatBot/AfDStatsTotal](#). In fact, I don't know what I was thinking when I made that a separate page. I'll move it to [User:AfDStatBot/AfDStats](#). Does that same objection apply to [User:AfDStatBot/AfDStats](#)? [Stu](#) ([aeiou](#)) [Im Researching Wikipedia](#) 00:06, 12 February 2009 (UTC)

I'm entirely confused here. If this is just for historical statistics, why on earth would it need to update so frequently? --[MZMcBride](#) ([talk](#)) 22:58, 11 February 2009 (UTC)

It isn't primarily for historical statistics - AfDStatsTotal is, and I'm moving that to AfDStats because it is unnecessary to have it on a separate page. The problem is that there is no centralized page with an up-to-date list of all open and recently-closed AfDs, either together or separately. AfDs in general are indexed by their opening date, so you may have to look through 7 pages to get a list of all open and recently-closed AfDs. Lists of recently-closed AfDs

Figure 7: Excerpt from the now archived Bot Approvals Group discussion about AfDStatBot (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

### 4.3: Moving out and away

AfDStatBot has since left Wikipedia (and Twitter), although its traces in those spaces do remain. As another mode of visual representation shows — screenshots of its Wikipedia contribution history (Figure 8) — my bot abruptly ceased all activity on June 1st, 2009.

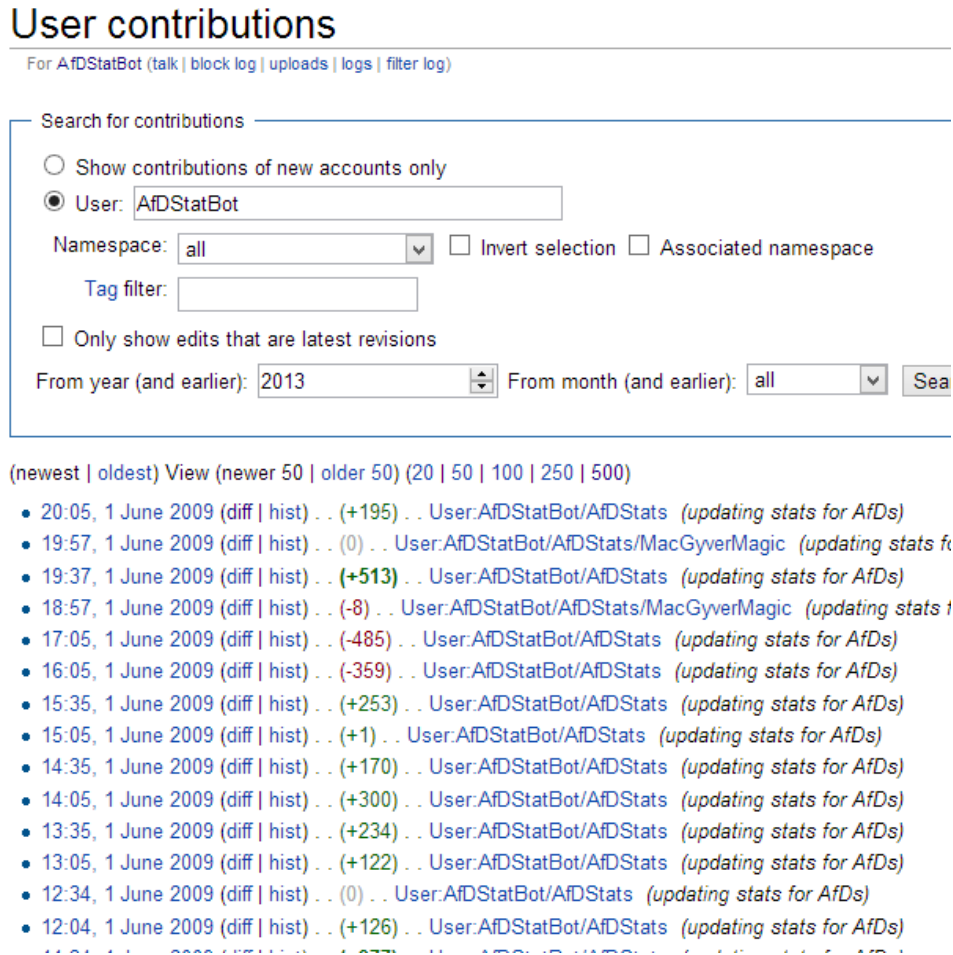


Figure 8: AfDStatBot's contribution history (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

These traces relate what the bot did every single day, serving as a diary of sorts. By June 1st, 2009, the bot had seen 23,240 articles nominated for deletion, each of which it tracked to the best of its ability. The last recorded action the bot made was at 5:30pm EST, when it saw a new AfD open for the article “K38IM,” a low-power repeater for an Illinois-based Christian

television network. It saved it in its database, posted this development to Twitter, and, had it stayed online for just a few minutes more, it would have included this new discussion in its regular half-hour update to the AfDStatBot information pages on Wikipedia. But it didn't. Instead, for what has now been just over four years, the bot went offline and the account was never logged in again. As with so many other Wikipedia editors who suddenly stop contributing to the project, the bot's userpage remained in the same state it was in for years, representing a user who was no longer present. Three years later, someone using an automated tool found that my bot "appears to be inactive" and changed a tag that indicated it was no longer an active bot (as seen in Figure 6).

June 1st, 2009 was an emotional day for me, one I'll remember for some time — and not because it was the day my bot died. I was living in Washington, D.C. My bot had been running on my own personal desktop computer in a corner of a tiny apartment in Dupont Circle (Figure 10), where I had been living for almost two years with my partner at the time. We had recently broken up, and our lease ended on June 1st. We had both found new places to live and had been moving our stuff out of the apartment bit by bit, but June 1st was the date we had to get everything out. At the time, unplugging my desktop computer and taking the bot down was the least of my concerns. It seemed that everything in my life was in a state of transition, and I was already shirking obligations far more pressing than my bot. And my Master's thesis on Wikipedia — the original reason I developed the bot — was done, defended. As everyone feels after a big project, I was ready to take a break from Wikipedia. At the time, I don't know if I even remembered that the bot was still running on that computer when I shut it down and pulled the plug. There were a lot of things I lost that day, from the many physical and digital objects I threw away or destroyed, to parts of myself I wouldn't know were gone until weeks, months, even years later. That evening, as I entered a new apartment in Southeast D.C. which would never really be home, I was struggling to work out a lot of changes in who I was. But the fact that I wasn't a bot operator anymore didn't even register.



Figure 9: My former apartment; the cord in the foreground is the Ethernet cable connecting the computer to the router (author’s own photo).

#### 4.4: Code/coda

When I went searching through various archives and backups to tell the story of AfDStatBot years later, it was far easier for me to find the bot’s source code (Figure 11) and the database it used (Figure 12) than finding those photos of my life years ago – both logistically and emotionally. However, I didn’t want to start this story by talking about code and data. That said, if you want to know what AfDStatBot did, looking at the code and database will tell you a number of things you may not be able to otherwise know from looking at it. What it meant to make this bot capable of identifying specialized administrative processes in Wikipedia can be seen in how it used traces that were left in the creation of internal MediaWiki categories to detect new deletion discussions. Wikipedia-wide policies about how much load bots could place on the server can be seen in rate-limiting functions that are bundled with each request. The bot interacted via Wikipedia’s Application Programming Interface (API), which had been specifically designed to enable bot development — a situation that is not always the case in most other platforms.

```

foreach ($afd as $title) {
    $fail = false;
    if(strpos($title,"deletion/Log/")) {$fail=true;}
    //Put new AfDs in database
    $clean_title = mysql_real_escape_string($title);
    $query = "SELECT * FROM open WHERE nomtitle='".$clean_title.'";
    $result = mysql_query($query);
    //print $query."\n";
    if (mysql_num_rows($result)==0) {
        print "\nPossible new entry for ".$title;
        //Get AfD PageInfo - pageid
        $pageinfo=$sx->getPageInfo($title);
        //print_r($pageinfo);
        $pageid=$pageinfo['pageid'];
        //Check if AfD page is in AfD debates category

        $afd=false;
        $afdpagecat=$sx->callAPI("action=query&prop=categories&titles=");
        foreach(array_flatten($afdpagecat) as $item) {
            if($item=="Category:AfD debates") { print ", is an afd\n"; $afd=true; }
        }
        if($afd==false) {$fail=true;}

        //Get AfD revisions info - nomrev, nominator, nomdate, nomcomment
    }
}

```

Figure 10: Excerpt of AfDStatBot's source code (author's own code)

title	nomdate	nomrevid	pageedits
...Fuck It?!	2009-01-25 01:04:22	266222295	34
3SL Cradle	2009-01-21 13:05:15	265476108	45
9th Level Games	2009-01-21 13:23:51	265478368	23
APE Project	2009-01-21 05:37:33	265433828	32
AZ Dodgeball	2009-01-23 21:46:42	265993899	14
Aaron Bacon	2009-01-23 23:03:06	266008132	5
Adam Kuban	2009-01-13 00:10:47	263699892	31
Adam Orogljo	2009-01-26 19:42:20	266586403	23
Aesop And Son	2009-01-18 19:51:40	264924947	10
Aeysha	2009-01-19 23:47:23	265184375	11
Aficionado Films	2009-01-23 16:45:08	265937664	20
Ahsan virus	2009-01-26 15:41:31	266537881	5
Airburst	2009-01-25 23:58:29	266416942	42
Aitor Iturrioz	2009-01-22 20:59:18	265780379	21
Alan Tuan	2009-01-22 22:30:36	265796415	5
Aleksander CepuÅ;	2009-01-26 21:09:38	266605277	63

Figure 11: "SELECT \* FROM open" query result from AfDStatBot's mySQL database (author's own code)

In all, someone who can read PHP may be able to walk through my code and get a good understanding about what that bot was doing when it still operated. You can find all kinds of socio-technical artifacts in that code – in two senses of the term, as they are both technological artifacts as well as artifacts of a certain social, historical, and political situation.

With this source code, AfDStatBot could have been revived after I got my living situation settled. It could have been one of the many times when bots in Wikipedia mysteriously go down for a few days or a week, then suddenly re-appear. In fact, it still could go through such an act of resurrection, although it might need some updates given new developments that have been made to Wikipedia's API and how Wikipedians organize and track deletion debates. Once I had everything settled and got Internet access in my new apartment, I could have started the bot back up again. Instead, when I got settled in my new apartment, I took that desktop computer apart, sold some components, upgraded it to a gaming machine, and formatted the hard drive. The bot never made it back. Yet even during the move, I could have “just” taken these few hundred kilobytes of code and run them on a different server. I couldn't run it on the shared server I use for webhosting because they do not allow any kinds of bots, though I could have easily found a home for it. But I didn't, because I just didn't care -- the bot (and much of Wikipedia's internal operations) had ceased to be a part of the world in which I lived. If I had cared, the bot may still be running today. Scotty may never have made Snotbot to do the work it wasn't doing anymore.

If I had not chosen to run that bot from that tiny apartment I shared in Dupont Circle, would it have been the same bot? What if, from the beginning, I had decided to run my bot on the toolserver, a shared server funded and maintained by a group of German Wikipedians for all kinds of purposes, including bots? If so, the bot may have run the same code in the same way, producing the same effects in Wikipedia, but in another sense it would have been an entirely different thing. I could have forgotten about it, it could have ceased to matter in my life, but it would have kept running – until something changed in Wikipedia that broke the bot until I fixed it. What if AfDStatBot was the kind of project I could further develop into an even more sophisticated bot as a way of getting my mind off everything else that was going on? What if it had been the kind of bot that, when it went down, my talk page on Wikipedia would have been flooded with frantic inquiries about what happened to the bot and when it would be back? What if it had been the kind of bot that was not an individual project, but one where I was part of a team sharing responsibility for the bot's development and operation? What if things with my partner hadn't gotten to the point where that apartment lease was the only thing keeping us together anymore? What if? What if?

## **5. Conclusion**

This vignette illustrates how an algorithms-in-the-making approach focusing on bots as bespoke code presents a quite different account of algorithms, compared to those which focus on the ready-made algorithms that have already been designed, developed, and deployed. The work of unpacking the black box of “algorithms” includes revealing and reading through source code, but there are important materially and historically contingent aspects to the operation of



a bot that go beyond source code. Reading code can be quite revealing, but it requires a relational, networked, material, and infrastructural understanding of software design, development, and deployment to understand how code is situated in many different kinds of spaces. My experiences as both a long-time Wikipedian editor in general as well as a bot developer have given me a different perspective on what a bot is as an algorithmic system, which is the kind of gestalt shift Seaver (2013) has similarly located in his ethnography of music recommendation services:

When we realize that we are not talking about algorithms in the technical sense, but rather algorithmic systems of which code *strictu sensu* is only a part, their defining features reverse: instead of formality, rigidity, and consistency, we find flux, revisability, and negotiation. (Seaver, 2013)

In the subsequent chapters in this dissertation, I tell of many similar stories about the bots I have studied across first Wikipedia and then Twitter, although with quite different cases and findings. Yet they all speak to how bots can be part of broader conversations, goals, visions, and ideals about what these online spaces are for and what they ought to become. The work of designing, developing, and deploying automated software agents is deeply embedded in not only dominant and nascent technical systems and codebases, but also dominant and nascent social systems and institutions. Important moments in the history of both Wikipedia and Twitter cannot be told independently of the failed and successful bots that, in continuously performing some task a volunteer software developer thought needed to be automated, raised broader issues about what kind of tasks ought to be done, how they ought to be done, and the meta-level decision about who gets to make these kinds of decisions in the first place. As I explore in several cases, bots that are delegated governance work raise issues that are common across many digitally-mediated environments: formalization and proceduralization, socialization and membership, struggles between developers and non-developers, invisible and infrastructural work, and the classification of cultural content. Each of the chapters in this dissertation focuses on a specific case or set of cases that speaks heavily to one of these issues, but such issues are also found throughout all of these cases.

## Chapter 3: Exclusion compliance

### 1. Introduction

#### 1.1 Chapter overview

This chapter examines the emergence of the concept of “exclusion compliance” in Wikipedia, which is an emic term that was first created by Wikipedians to help resolve a 2006 controversy over a new kind of bot in the encyclopedia project. By 2006, many Wikipedia bots had been designed, developed, and deployed to write or modify encyclopedia articles according to particular standards and procedures, but “HagermanBot” was the first to universally enforce a norm about how Wikipedians ought to interact with each other in discussion spaces. The case of HagermanBot is an important moment in the early history of Wikipedia bots, as the bot raised issues about both the specific task it was delegated as well as issues about how the recently-founded “Bot Approvals Group” was to mediate between bot developers and Wikipedians who opposed certain bots. The controversy was resolved through the development of a new norm that bots ought to support opt-out procedures when performing certain tasks in Wikipedia. This norm was then supported by new technical standards specifying how Wikipedians could indicate their intent to opt-out. Today, this standard of exclusion compliance is implemented by default into many of the specialized software libraries that have been built to support the work of bot development in Wikipedia, and bot developers who do not abide by such norms have been subject to severe consequences by Wikipedia’s administrators.

In the broader argument running throughout this dissertation, this chapter further expands the analytic frame of bots as more than their source code. The previous chapter (particularly the vignettes about my first bot) established bots as a two-member project, a team effort between the software agent and the human operator. Without the bot operator maintaining the computational infrastructure hosting the software agent and going through the steps needed to get it approved in Wikipedia, the bot falls away to nothing more than an idea. Picking up from this argument, this chapter expands the concept of the bot as a project to include other people in Wikipedia beyond the bot developer, who are also involved in kind of team effort. I focus more on the administrative apparatuses in which Wikipedia bots are approved and regulated, particularly the Bot Approvals Group. A bot without an operator may just be an idea, but an idea that a certain task ought to be performed in a specific way by a bot is just as important to a bot’s operation as its computational infrastructure. If a bot loses support of the ‘ideological infrastructure,’ performing tasks in a way that has little resonance with how Wikipedians think the encyclopedia project ought to operate, then it is likely to be banned – which is as swift of a death as pulling the plug on the server. In this chapter, I expand the idea of the bot beyond its code, operator, and computational infrastructure to encompass a project to produce a collective understanding about the task the bot is to perform. In the case I investigate, the ‘team members’ in this ‘project’ ultimately produced a different version of the software agent than Hagerman initially developed, as they worked out what it meant to automate the bot’s task of cleaning up comments in discussion spaces. Furthermore, their lasting legacy goes beyond this specific task, as the compromise produced a much broader set of products: a norm about how software agents ought to behave to those who wanted to opt out, a universal standard for Wikipedians to signal their intent to opt out, opt out detection functions implemented in

multiple programming languages, and policies for ensuring that relevant bots were made exclusion compliant.

While I emphasize the deeply human aspects of bots as projects that go beyond the bot's software agent and the bot's developer, I also continually keep the work of software development in close view. It is just as problematic to discuss bots as purely 'social' as it is to discuss bots as purely 'technical,' a position I take from theorists in science and technology studies (e.g. Haraway 1991; Latour 1988). The history of exclusion compliance in Wikipedia was a multi-faceted achievement, one that operates in multiple spaces and is sustained by a wide variety of work. Accordingly, before I begin with the case of HagermanBot, I present an introductory vignette unpacking a function that is today buried deep in the code of a popular Wikipedia bot development library, which makes all bots developed with this library exclusion compliant by default. As I open up this black-boxed element in mid-2015, I show how it is the product of a set of norms and standards which were first developed in Wikipedia almost ten years ago.

## **1.2 What is the `antibot_template_protected_text` function?**

For software developers who want to create a bot for Wikipedia, there are dozens of libraries and toolkits which have been created to make bot development simple, easy, and standardized. The *apibot* package is one of these libraries, written for PHP by a team of three developers. The library is sophisticated, with over 90,000 lines of code supporting a variety of automated tasks related to wikis. As the developers explain in documentation, the library makes it easy to write a program that will automatically perform tasks like going through a page in a wiki and replacing one word with another – or doing a find-and-replace through every article in a wiki. Once the modules are imported, all of Wikipedia's content can be encapsulated into a computational object, called a "core." Code can be written that interacts with the wiki as if the text and metadata of every page was stored in local memory. When a request is made to access or modify the core, the library connects to Wikipedia's Application Programming Interface (or API) and passes the request along. The code below imports the *apibot* library, creates the core object, creates a page object for the article on San Francisco, and prints the text of the page on the screen.

```

# Include the needed modules.
require_once ( dirname ( __FILE__ ) . '/apibot/settings.php' );
require_once ( dirname ( __FILE__ ) . '/apibot/logins.php' );
require_once ( dirname ( __FILE__ ) . '/apibot/core/core.php' );
require_once ( dirname ( __FILE__ ) . '/apibot/interfaces/line/__all_modules.php' );

# Create the Core object
$core = new Core ( $logins['My_Bot@en.wikipedia.org'], $bot_settings );

# Using the core object, create a Page object on the article on San Francisco
$SFpage = new Page($core, "San Francisco");

# Print info about the page's name and when it was last edited to the screen
print("The page on " + $SFpage->name + " was last edited on " + $SFpage->timestamp);

# Print the text of the page on the screen
print($SFpage->name);

```

The encapsulation of Wikipedia into a single computational object like apibot's core is useful for programmers, as it backgrounds the routine work involved in connecting to Wikipedia's servers, authenticating with login credentials, requesting information about a page or a user, and sending a new version of a page (or taking any number of other actions on the site). There are dozens of variables which can be easily accessed in a page object. Furthermore, there are page revision objects for each distinct version of every page, as well as objects for users, logging events, and administrative actions. apibot contains a number of functions built into these objects that work to streamline commonly occurring tasks. For example, every page object has not only the text of the page accessible through a simple variable, but also the built in "replace\_string" function. This means that once a core and page object have been instantiated, replacing all instances of one word or phrase with another can be executed in one line of code:

```

# Replace all instances of the word "Frisco" with "San Francisco"
$SFpage -> replace_string("Frisco", "San Francisco");

```

Buried many levels deep in the dependencies of this "replace\_string" function are hundreds of sub-functions which make it possible to open up a connection with Wikimedia's servers, authenticate with login credentials, download the text of the page into local memory, rewrite the page in local memory, request permission to edit the page, and then pass the new version of the page to the server. No matter what kind of task a bot using apibot is performing, if it makes an edit to a page, many of these same functions are called. This is standard in contemporary software development and why a library like apibot is so powerful and useful for bot developers: when they want their bot to edit a page, they do not have to re-write (or even copy and paste) all the code needed to edit a page.

Of these many dependent functions, there is one curious function that has made its way into the core functionality of apibot, called whenever any page is to be edited: **antibot\_template\_protected\_text**. As any bot using apibot prepares to tell the Wikipedia API to edit a page in a particular way, it queries the API for the current version of the page, even if it had been retrieved seconds before as part of another task (like the find-and-replace task in the

example documentation). It then calls this “antibot” function, which scans through the entire text of the page, looking for particular strings of text wrapped around double curly brackets: text like `{{nobots}}` or `{{bots|deny=OurBotUsername}}`. The code of this function is below (I have added comments in orange that are not in the source code of apibot):

```
protected function antibot_template_protected_text ( $text, $username ) {
    return (bool) preg_match ( # return true if the following pattern is found:
        '/\{\{(' . # begins with “{{” and is followed by any of:
            'nobots|' . # “nobots”
            'bots\|allow=none|' . # “bots|allow=none”
            'bots\|deny=all|' . # “bots|deny=all”
            'bots\|optout=all|' . # “bots|optout=all”
            'bots\|deny[^\|\\]*[=,]\s*' . # “bots|deny=OurBotUsername”
            preg_quote ( $username, '/' ) . '\s*[\\|\\|\\]' .
        '\}\}\}/is', # and ends with “}}”
        $text ); # search in the page text of the current page
}
```

The results of this antibot function is then fed into the core “writer” functions, setting a variable named “nobottemplate” to true if one of those patterns was matched. After this, another segment of code is called whenever an apibot-based bot goes to run the functions that will query the API to edit the page. If the nobottemplate variable is set to true and the bot’s settings have the variable “honor\_nobottemplate” set to true (which can be overridden, but is set to true by default), then the bot will not direct the API to edit the page, instead throwing an error. This error will not crash the bot or stop it from operating, but it will log this event and then move on to the next edit to be made, if there are more. In all, this means that apibot is built such that by default, any bot built using this framework will not edit any page that contains the text `{{nobots}}`, `{{bots|allow=none}}`, `{{bots|deny=all}}`, `{{bots|optout=all}}`, or `{{bots|deny=OurBotUsername}}` in the page. In the markup language used in the MediaWiki platform, this text is visible to those that view and edit the page’s source, but invisible to readers.

```
if ( isset ( $page['nobottemplate'] ) &&
    $page['nobottemplate'] &&
    $this->settings['honor_nobottemplate'] )
{
    $this->log ( $logbeg . " denies write with a {{Bots}} template",
        LL_ERROR );
    return false;
}
```

These lines of code make apibot an “exclusion compliant” bot development framework, which is an important feature in Wikipedian bot development and can be found in many – but not all – bot development frameworks released for Wikipedia. The concept is one that defines

and specifies a particular kind of ideal interaction between bots, bot developers, and Wikipedia editors: an exclusion compliant bot will look for a machine-readable trace in the text of a wiki page indicating that bots are not welcome to edit the page, respecting that request. Wikipedia's exclusion compliance is similar to the robots exclusion standard – the 'robots.txt' files that direct search engine crawlers about what pages of a website they are and are not allowed to crawl (Elmer, 2009; Lundblad, 2007). Like the robots exclusion standard, exclusion compliance in Wikipedia is not a firm software-based barrier preventing undesired interactions from automated agents, but one that is based more on a particular notion of how bots are to interact with those who disagree with their tasks. As I show with the case of HagermanBot in this chapter, standards like exclusion compliance are certainly technical standards implemented in code, but this code is not all there is to the story. Functions like `antibot_template_protected_text` are bound up in more complex norms, processes, and administrative apparatuses, which are mutually co-constructed through controversies like the case of HagermanBot.

### 1.3 Theorizing delegation in organizations

To specifically investigate this case, I draw on theories of delegation from science and technology studies and organizational studies, which show how controversies over the delegation of work to bots are a particularly powerful window to explore various issues about how Wikipedia operates. This chapter's theoretical approach on delegation draws on two related literatures from organizational studies and Science and Technology Studies. Over the past two decades, the sub-field of organizational studies concerned with the role of technology has increasingly emphasizing the "sociomateriality" of information systems within organizations, focusing on how people and information technology are entangled in practices (Orlikowski & Scott, 2008). This literature opposes a more traditional view that, as Zammuto et al. critique, sees information technology as a kind of "automated plumbing" which simply makes the organization more efficient without many more substantial consequences (Zammuto et al., 2007). The sociomateriality literature calls attention to broader shifts that IT can bring in reshaping organizational roles, responsibilities, and relationships. I also draw on Bruno Latour's writings on delegation in science and other actor-network theorists who conceptualize society as a heterogeneous network of assembled relations between human and non-human actors (Callon, 1986; Latour, 1999a). Acts of delegation – either to humans or to non-humans – are key moments in which both work and assumptions about work are made visible and brought to the foreground. Controversies and breakdowns can emerge over the creation or modification of "delegation regimes" (Ribes et al, 2013), which provoke reflection by participants on the broader issues at work in a given organization.

Like Latour's example of delegation in policing speeding cars (Latour, 1999b), bots are not mere tools like radar guns that simply improve the effectiveness of a single human enforcer. They are instead closer to the speed bumps Latour analyzes as sociotechnical actors, which can initially seem like autonomous, independent agents. A neighborhood that decides to punish speeding cars can delegate this responsibility to speed bumps instead of police officers, which perform roughly equivalent actions. Yet while Latour and other actor-network theorists defend a functional equivalence between human and non-human actors in their ability to engage in social interactions, Latour stresses that the nature of the task being performed and the constellation of actors around it can be fundamentally changed when the delegation is to a

technological actor instead of a human one. For example, compared to police officers, speed bumps are unceasing in their enforcement of the ‘no speeding’ rule, equally punishing reckless teenagers and on-call ambulances. A similar ‘ruthlessness’ can be seen in the initial wave of HagermanBot’s operation, which raised the controversy over the bot’s operation in Wikipedia in the first place.

However, Latour insists that this stance towards the agency of technology is opposed to the claims of technological determinism and domination, such as Langdon Winner’s account of the low buses on Long Island that let private cars but not public buses pass through (Winner, 1986). The speed bump (and the bot) may appear to be “nonnegotiable,” but he stresses that we must not be fooled into thinking that we have “abandoned meaningful human relations and abruptly entered a world of brute material relations” (187). Instead, the actor-network theory approach sees technologies and humans as interdependent social actors, calling on researchers to trace out the network in which such an artifact operates – particularly the often invisible work required to sustain them in their proper operation. In such an investigation, a seemingly ‘technical’ artifact is simply the end product of much longer endeavor to remake the world into something else:

In artifacts and technologies we do not find the efficiency and stubbornness of matter, imprinting chains of cause and effect onto malleable humans. The speed bump is ultimately *not* made of matter; it is full of engineers and chancellors and lawmakers, commingling their will and their story lines with those of gravel, concrete, paint, and standard calculations. (190).

In this broader view, it may actually be easier for someone to “negotiate” with speed bumps than a police officer, particularly if someone has more influence in a city’s public works department than the police department. However, those with no influence in either departments are likely to remain subject to the will of both.

Similar to Latour’s speed bumps, bots in Wikipedia are non-human actors, constructed by humans and delegated the highly social task of enforcing order in society, but they also rest on an existing set of social and technical infrastructures which make their development and continued operation possible. Bots may initially appear to be as non-negotiable as speed bumps, with their creators seemingly able to dominate the unsuspecting masses with their technical skills and remake Wikipedia in their own image – particularly for those who does not have much influence or status within Wikipedia. To push back on these narratives of determinism (in either their social or technical flavors) I pay close attention to the conditions in which bots emerge within the organizational culture of Wikipedia – a complex collective of editors, administrators, committees, discussions, procedures, policies, jargon, precedents, and shared understandings. At the same time, I discuss these seemingly ‘social’ phenomena as they are bound up in what may be cast as purely ‘technical’ phenomena, like software libraries, algorithmically-encoded procedures, and administrative capacities to block a bot account from accessing the site.

Bots, like infrastructures in general, simultaneously produce and rely upon a particular vision of how the world is and ought to be, a regime of delegation that often sinks into the

background – that is, until they do not perform as expected and generate controversies and breakdowns (Star, 1999b).<sup>28</sup> In these moments of breakdown, these worldviews about what work ought to be done in Wikipedia are expressed by Wikipedians in many different ways, from conversations to code. Crucially, people who have no direct access to modify the bot's source code (or even block it from operating) are able to express ideas about how the bot ought to be programmed. In this way, bot development in Wikipedia provides an invaluable moment for community members to make assumptions about governance explicit. This explication is done not only in directly editing the software code of a bot (which only the operator can do), but also in discussions about what ought to be done about that software code. These accounts of controversies over bots touch on a wide range of other norms and organizational structures inside of Wikipedia, including the infrastructures used by bot developers to support their own work.

## **2: A history of exclusion compliance in Wikipedia**

The bot I primarily examine in this chapter was an early bot in Wikipedia's history, one of the first delegated the work of enforcing a norm about how Wikipedians were to interact with each other, as opposed to earlier bots that made programmatic changes to encyclopedia articles. This norm – that people should sign and date their comments in discussion spaces, a feature not built into the stock MediaWiki platform – was seemingly universal, enshrined in a document in Wikipedia's policy environment. When a bot began universally enforcing this norm, those who felt they had the right to not sign and/or date their comments appeared to contest the code of the bot. In the ensuing controversy, bot developers and non-developers debated not only the utility of signatures and timestamps, but also the meta-norms about if and when Wikipedians could opt out of having a bot enforce such a norm. The compromise reached in this case established both new norms and new software-based standards structuring how bot developers were to interact with those who objected to their bot – or more specifically, those who objected to the developer's normative vision of how Wikipedia ought to operate, which was being implemented project-wide through the code of the bot.

### **2.1: A problem and a solution**

Wikipedians conduct a significant amount of communication through the wiki site using designated discussion (or talk) spaces. These spaces are, at a software level, functionally identical to the collaboratively-edited encyclopedia articles: they are both flat text files that anyone can edit. Unlike the vast majority of on-line communication platforms, such as message boards, chat rooms, e-mail listservs, or social media sites, MediaWiki is not specifically designed to support threaded communication. To add a comment to a discussion, a user edits the entire discussion page, appends a comment to the proper location in the page (usually manually indented to indicate threaded replies), and saves the new revision. This use of wiki pages for discussion means that malicious users can remove or edit someone else's comments just as easily as they can edit an encyclopedia article, although this is highly discouraged and mitigated by the fact that the wiki platform saves a public history of each revision. However,

---

<sup>28</sup> for more on the phenomenology of breakdown and repair, see (Heidegger 1993; Vygotsky 1968; Jackson 2014).



another problem had arisen in wiki pages used as discussion spaces: many people left comments in talk pages, but did not leave a signature or a timestamp. MediaWiki had long had a special function in its parser functions that let editors automatically append their username and the current time and date by writing four tildes: ~~~~. Yet this had to be done for every comment, and many Wikipedians would forget to do so. For the purposes of discussions, this made it difficult to determine not only who made a certain statement, but also when it was made. Someone could go through the revision histories to find this information, but it is tedious, especially in large discussions. Like with many tedious tasks in Wikipedia, a few editors sensed that there was a need for someone to do this work and began to do it manually – users like ZeroOne.

Among other tasks, ZeroOne’s contributions to Wikipedia included scanning through discussion pages for unsigned comments and manually appending a signature on the commenter’s behalf. For example, at 06:15 on 17 October 2006, user ZeroOne made his 4,072<sup>nd</sup> contribution to Wikipedia, editing the discussion page for the article on “Sonic weaponry.” Instead of adding a comment of his own about the article, he appended the text `{{unsigned|71.114.163.227|17 October 2006}}` to the end of a comment made by another user about twenty-five minutes earlier [05:50]. When ZeroOne clicked the Submit button, the MediaWiki software parsed his contribution and transformed it into a pre-formatted message. Together, the edits of 71.114.163.227 and ZeroOne added the following text to the article’s designated discussion page:

Ultrasound as a weapon is being used against American citizens in Indiana. Any experts out there wish to make a study, look to Terre Haute, maybe its the communication towers, that is my guess. It is an open secret along with its corrupt mental health system.  
—Preceding [unsigned](#) comment added by [71.114.163.227](#) ([talk](#) • [contribs](#)) 17 October 2006

Two minutes later [06:17], ZeroOne performed the same task for an unsigned comment made by a registered user on the talk page for the “Pseudocode” article – adding `{{unsigned|Blueyoshi321|17 October 2006}}`. About two hours later [08:40], he spent twenty minutes leaving `{{unsigned}}` messages to the end of eight comments, each made on a different discussion page. While ZeroOne could have manually added the “Preceding unsigned comment added by...” text to issue the message, this process was made standard and swift because of the `{{unsigned}}` template. Templates are a form of bespoke code; they are a MediaWiki feature in which editors write customized scripts, which are called as functions when left in the text of articles. The first parameter of this function takes the author’s name or IP address and the second takes the date. For those who edit the page’s source, the text still appears as a template function call, but for readers, it appears as the marked up signature.

While the existence of templates made ZeroOne’s work somewhat automated, this editor felt that it could be made even more so with a bot. ZeroOne soon posted this suggestion in a discussion space dedicated to discussions about requests for new bots. Over the next few weeks, a few users mused about its technical feasibility and potential effects without making any concrete decisions on the matter. The discussion stagnated after about a dozen comments and was automatically moved into an archive by a bot named Werdnabot on 16 November 2006, after having been on the discussion page for fourteen days without a new comment. The next

month, another user named Hagerman was hard at work in realizing ZeroOne's vision of a bot that would monitor talk pages for unsigned comments and append the `{{unsigned}}` template message without the need for human intervention, although it is unclear if Hagerman knew of ZeroOne's request. Like ZeroOne, Hagerman had used the template to sign many unsigned comments, although many of these were his own comments he had forgotten to sign, rather than ones left by others.

On 30 November 2006, having finished programming the bot, Hagerman registered a new user account for HagermanBot and wrote up a proposal the next day. In line with Wikipedia's policies on bot operation at the time, Hagerman submitted his proposal to the members of the Bot Approvals Group (BAG), an ad-hoc committee tasked with reviewing bot proposals and ensuring that bots are operated in accordance with Wikipedia's policies. Tawker, the operator of the prolific AntiVandalBot and a member of the BAG, asked Hagerman for a proof of concept and posed a technical question about how the bot was gathering data. Hagerman provided this information, and Tawker approved the bot about 24 hours later, with no other editors taking part in the discussion. On 00:06 on 3 December, HagermanBot began operation, automatically appending specialized `{{unsigned}}` messages to every comment that it identified as lacking a signature. The first day, 790 comments were autosigned, and HagermanBot made slightly over 5000 edits over the next five days. By the end of December 2006, HagermanBot had become one of the most prolific users to edit Wikipedia in that month in terms of the raw number of edits to wiki pages, outpacing all other humans and almost all other bots.

## 2.2: A problem with the solution

There were a few problems with the bot's comment and signature identification algorithms, making it malfunction in certain areas, but Hagerman promptly fixed these errors. One issue arose with edits to talk pages that were not comments, like adding an informational banner at the top of a talk page that the associated article would be featured on the main page next week (a common practice). To prevent this, Hagerman extended the bot's code so that it would not sign a comment that contained the text `!NOSIGN!` in the edit summary.<sup>29</sup> However, some users were annoyed with the bot's normal functioning, although for a variety of different reasons. One group complained that it instantly signed their comments instead of giving them time to sign their own comments after the fact. For these editors, HagermanBot's message was "embarrassing," as one editor stated, making them appear as if they had blatantly violated the Signatures guideline. They requested a short delay so they could have the opportunity to add their own signature again before HagermanBot beat them. Others did not want bots editing messages other users left for them on their own user talk pages as a matter of principle. Finally, a small but vocal group did not want the bot adding signatures to their own comments, taking issue with the seemingly-universal norm of signatures and/or timestamps.

---

<sup>29</sup> The edit summary is a 200 character field where editors can describe the changes they made. Edit summaries are metadata that show up in the revision history of articles. By 2006, Wikipedians had already begun leaving short codes in edit summaries to specify certain aspects of their edits to bots and semi-automated tools, primarily in the areas of anti-spam and counter-vandalism activity, as described in (Geiger & Ribes, 2010; Geiger & Ribes, 2011).

While Hagerman placated those who did not want to be embarrassed by adding a delay, the issue raised by the other group of objecting editors was more complicated. These users were, for various reasons, firmly opposed to having the bot transform their own comments. One user in particular, Sensemaker, did not follow what was claimed to be the generally-accepted practice of using four tildes (~~~~) to automatically attach a linked signature and timestamp, instead manually adding “-Sensemaker” to comments. HagermanBot did not recognize this as a valid signature and would therefore add the `{{unsigned}}` template message to the end, which Sensemaker would usually remove. After this occurred about a dozen times in the first few days of HagermanBot’s existence, Sensemaker left a message on Hagerman’s user talk page, writing:

HagermanBot keeps adding my signature when I have not signed with the normal four tilde signs. I usually just sign by typing my username and I prefer it that way. However, this Bot keeps appearing and adding another signature. I find that annoying. How do I make it stop? -Sensemaker

Like with the previous request, Hagerman initially responded quickly, agreeing to exclude Sensemaker within ten minutes of his message and altering the bot’s code fifteen minutes later. However, Hagerman soon reversed his position on the matter after another editor said that granting Sensemaker’s request for exclusion would go against the purpose of the bot, emphasizing the importance of timestamps in discussion pages. Sensemaker’s manual signature did not make it easy for a user to see when each comment was made, which Fyslee, a vocal supporter of the bot, argued was counterproductive to the role of discussion spaces. Hagerman struck the earlier comments and recompiled the bot to sign automatically sign Sensemaker’s comments, again calling Fyslee’s remarks “Very insightful!” As may be expected, Sensemaker expressed frustration at Hagerman’s reversal and Fyslee’s comment – in an unsigned comment which was promptly ‘corrected’ by HagermanBot.

For Sensemaker and other editors with similar complaints, it was not clear “who gave you [Hagerman] the right to do this,” as one anonymous user who contested HagermanBot exclaimed. Hagerman responded to such rights-based arguments by linking to his bot proposal, which had been approved by the Bot Approvals Group – trying to enroll this committee as an ally in defense of the bot. At the time, Hagerman had a strong set of allies mobilized: a growing number of enthusiastic supporters, the BAG, the Signatures guideline, ideals of openness and transparency, visions of an ideal discursive space, the `{{unsigned}}` template, and a belief that signing unsigned comments was a routine act that had long been performed by humans. Yet for some reason, a growing number of editors began objecting to this regular, uncontroversial practice when HagermanBot began performing it automatically for every unsigned comment in Wikipedia.

Having failed to convince Hagerman one-on-one, Sensemaker shifted venues and brought the issue to the members of the Bot Approvals Group. Sensemaker formally asked the BAG to require an opt-out mechanism for the bot, lamenting that Hagerman could apparently “force something upon people who expressly ask to be excluded.” Many editors who had previously left their comments either unsigned or signed with non-standard signatures began to make themselves visible, showing up at Hagerman’s user talk page and other bot-related

discussion spaces to contest what they portrayed as an unfair imposition of what they believed ought to be optional guidelines. The anti-HagermanBot group was diverse in their stated rationales and suggested solutions, but all objected to the bot's operation on some level. Some objectors staunchly opposed to any user signing their comments, bot or human, and took issue with the injunction to sign one's comments using the four tilde mechanism. Sensemaker was in this group, although others did not want to sign with either their name or a timestamp at all.<sup>30</sup> Another group did not want to see a bot universally enforcing such a norm, independent of their stance on the necessity of signatures:

I don't really like this bot editing people's messages on other people's talk pages without either of their consent or even knowledge. I think it's a great concept, but it should be an opt-in thing (instead of opt-out), where people specify with a template on their userpage if they want it, like Werdnabot, it shouldn't just do it to everyone. Just my two cents. --[Rory096](#) 01:36, 11 December 2006 (UTC)

In the ensuing discussion – which was comprised of BAG members, administrators, and other Wikipedians – it became clear that this was not a debate about simply signatures and timestamps. The debate had become a full-blown controversy about the appropriateness of delegating social tasks to technologies, and a number of the participants reflected that they had entered a new territory with this issue. There had been debates about bots in Wikipedia before, but most were not about bots in the abstract. Instead, prior bot-related discussions generally revolved around whether a particular task – which just happened to be performed by a bot – was a good idea or not. If there was a consensus for performing the task, the bot was approved and began operating; if there was no consensus, the bot was rejected, or suspended if it had already been operating. In the case of HagermanBot, critics increasingly began to claim that there was something fundamentally different between humans sporadically correcting violations of a generally-accepted norm and a bot relentlessly ensuring total compliance with its interpretation of this norm. For these objectors, the burden was on Hagerman and his allies to reach a consensus in favor of the current implementation of the bot, the bot was to continue operating.

The bot's supporters rejected this, claiming that the HagermanBot was only acting in line with a well-established and agreed-upon understanding that the community had long ago reached regarding the importance of signatures and timestamps in discussion spaces. For them, the burden was on the critics to reach a consensus to amend the Signatures guideline if they wanted to stop the bot from operating. Hagerman portrayed the two supported opt-out systems (!NOSIGN! in edit summaries and <!--Disable HagermanBot--> in pages) not as ways for users to decide for themselves if they ought to abide by the Signatures guideline, but rather as ways to keep the bot from signing particular contributions to talk pages that are not actually comments and therefore, according to the guideline, do not need to be signed. These exceptions included the various informational banners routinely placed on talk pages to let editors know,

---

<sup>30</sup> In WikiWikiWeb, the first wiki created by Ward Cunningham in 1995, there is a strong norm against having timestamps in comments. Signing a comment with one's username is also seen as an option; some do, some do not. Veteran WikiWikiWeb contributors talk about how “wiki is a perpetual now,” with one essay celebrating the potential to have an argument with yourself five years ago. I do not know if any of those who opposed the use of signatures and/or timestamps brought this tradition from WikiWikiWeb.

for example, that the article is being nominated for deletion or that it will be featured on the main page the next week. HagermanBot assumed total editorial compliance with the signature guideline; the two opt-out features were to ensure more conformity, not less, by allowing editors to tell the bot when a signature would be unwarranted according to the guideline. Those who were opposed to the Signatures guideline in general could use the tedious !NOSIGN! feature to make the bot not enforce the guideline when they made comments, but Hagerman urged them to not attempt to opt-out in this manner.

HagermanBot's allies were thus able to specifically articulate a shared vision of how discussion spaces were and ought to be, placing strong moral emphasis on the role of signatures and timestamps in maintaining discursive order and furthering the ideals of openness and verifiability. Like all approved bots which came before it, HagermanBot was acting to realize a community-sanctioned vision of what Wikipedia was and how it ought to be. The Signatures guideline was clear, stating that users were not to be punished for failing to sign their comments, but that all signatures should be signed, given that signatures were essential to the smooth operation of Wikipedia as an open, discussion-based community. Yet this proved inadequate to settle the controversy, because those opposed to HagermanBot were articulating a different view of how Wikipedia was and ought to be – one which did not directly contest the claims made regarding the importance of signatures, discussion pages, and communicative conventions. Instead, those like Sensemaker advanced an opposing view of how Wikipedians, and especially bot operators, ought to act to each other in the project, a view that drew heavily on notions of mutual respect:

Concerning your emphasis on the advantages of the bot I am sure that it might be somewhat convenient for you or others to use this bot to sign everything I write. However, I have now specifically requested to not have it implemented against my will. I would not force something upon you that you expressly said you did not want for my convenience. Now I humbly request that the same basic courtesy be extended to me. - Sensemaker

For HagermanBot's allies, these objections were categorically interpreted as irrational, malicious, or indicative of what an administrator named Rich Farmbrough called "botophobia." While this seems to be a pejorative description that would strengthen Hagerman's position, it restructured the controversy and allowed it to be settled in Sensemaker's favor. In entering the debate, Farmbrough argued that while Hagerman and his allies were entirely correct in their interpretation of the Signatures guideline, Hagerman should still allow an opt-out system:

On the one hand, you can sign your edits (or not) how you like, on the other it is quite acceptable for another user to add either the userid, time or both to a talk edit which doesn't contain them. Nonetheless it might be worth allowing users to opt out of an automatic system - with an opt out list on a WP page (the technical details will be obvious to you)- after all everything is in history. This is part of the "bots are better behaved than people" mentality which is needed to avoid botophobia. [Rich Farmbrough](#), 18:22 [6 December 2006](#) (GMT).

This mediation between the seemingly incommensurable views of how Wikipedia as a socio-technical system ought to operate proved enough to resolve the controversy. Declarations of what either side was entitled to, largely articulated in the language of positive rights, were displaced by the notion of responsibility, good behavior, and mutual respect. What it meant to be a good bot operator now included maintaining good relations with editors who objected to bots, or else risk a wave of anti-bot sentiment. The next day, Hagerman agreed, and the issue was settled:

Very true. That sounds like a great idea. I'll implement those changes this evening.  
Thanks, [Hagerman](#)<sup>(talk)</sup> 23:20, 6 December 2006 (UTC)

Ok, I've implemented an opt out procedure. Thanks  
again, [Hagerman](#)<sup>(talk)</sup> 03:35, 7 December 2006 (UTC)

Excellent, an opt out option was all I was asking for. A sensible but persistent appeal to a gentleman's sense of decency and reciprocity almost always succeeds. I am glad it did so in this case too. I have done a simple test and it seems to have worked. Thank you. —Sensemaker —  
Preceding unsigned comment added by [Sensemaker](#) ([talk](#) • [contribs](#)) 16:19, 7 December 2006

Glad it's working out for you. Best, [Hagerman](#)<sup>(talk)</sup> 23:43, 7 December 2006 (UTC)

### 2.3: An Unexpected Ally

While the opt-out list may seem like a concession made by Hagerman, it proved to be one of his strongest allies in defending HagermanBot from opponents, who were arriving in numbers to his user talk page and other spaces, even after the Sensemaker/Hagerman dispute had been settled. Most editors left value-neutral bug reports or positive expressions of gratitude, but a small but steadily-increasing number of editors continued to complain about the bot's automatic signing of their comments. The arguments made against HagermanBot were diverse in their rationales, ranging from complaints based on annoyance to accusations that the bot violated long-established rights of editors in Wikipedia. As one editor asked:

Who gave you the right to do this ? It is not mandatory that we sign, AFAIK. Instead of concocting this silly hack, why not get the official policy changed? I suppose you effectively did that by getting permission to run your bot on WP. How did you manage that anyway? (I won't bother with typing the fourtildas).

It isn't a policy, however, it is a guideline. You can view its approval at [Wikipedia:Bots/Requests for approval/HagermanBot](#). Feel free to opt out if you don't want to use it. Best, [Hagerman](#)(talk) 02:29, 5 January 2007 (UTC)

As seen in Hagerman's reply to this objection, an institutional ally was useful in rebutting the objections made against his bot: the Bot Approvals Group, which had reviewed

and approved the bot according to established protocols. The Signatures guideline, including the distinction between guidelines and policies, was also invoked to justify HagermanBot's actions, as shown in both examples. It would seem that these actors in Wikipedia's policy environment – which are generally taken to draw their legitimacy from a broad, project-wide consensus – would be the most powerful allies that Hagerman could deploy in support of HagermanBot's actions and its vision of how discussion spaces in Wikipedia ought to operate. However, a much stronger ally proved to be the opt-out list through which angry editors could be made to lose interest in the debate altogether. It is this last actor that was most widely used by Hagerman and his human allies, who began to routinely use the opt-out list to respond to a wide array of objections made against the bot.

The strength of the opt-out list was its flexibility in rebutting the objections from two kinds of arguments: first, the largely under-articulated claims that the bot was annoying or troublesome to them; and second, the ideological or rights-based arguments that the bot was acting against fundamental principles of the project's normative structure. The first argument was easy to rebut, given that the opt-out list completely responded to their more practical concerns. In contrast, those making the second kind of argument called forth juridico-political concepts of rights, autonomy, and freedom. Yet the same opt-out list could be invoked in HagermanBot's defense against these objections, as it foreclosed their individual claims that the bot was violating their editorial rights. While some objectors stated they would have preferred that the bot use an opt-in list to preemptively ensure the rights of all editors, the opt-out list allowed HagermanBot to be characterized as a supremely respectful entity that was, as the new philosophy of bot building held, "better behaved than people."

#### **2.4: The formalization of exclusion compliance**

HagermanBot's two new features – the opt-out list and the `<!--Disable HagermanBot-->` tag – soon became regular practices in Wikipedian bot development. Rich Farmbrough saw the value of these non-human actors who helped settle the HagermanBot controversy, and wanted to extend such functionality to other bots; however, its idiosyncratic mechanisms were unwieldy and specific to HagermanBot. About a week after Hagerman implemented the opt-out list, he was involved in a BAG discussion about a proposed bot named PockKleanBot, which was described by its operator PockingtonDan as a "nag-bot" that would leave messages for users on their talk pages if articles in which they had edited were flagged for cleanup. It was unleashed on Wikipedia without approval by the BAG and was promptly blocked by administrators; in the ensuing discussion, many editors and administrators called for the "spam bot" to be opt-in only. However, PockingtonDan argued that the bot would not be useful without sending unsolicited messages. In response, Rich Farmbrough suggested the same opt-out solution that had settled the HagermanBot controversy. Seeing a need for extending this functionality to all possible bots, he created a template called `{{nobots}}`, which was to perform the same function as HagermanBot's exclusion tag, except apply to all compliant bots.

Most templates contain some kind of pre-written message, but the message attached to the nobots template was blank. It would not change the page for readers, but it could be added by editors and detected by bots that downloaded its source code. If a user placed the text `{{nobots}}` on their user page, any bot that supported the proposed standard would not edit that

page in any fashion. Editors could also allow only specific bots access by writing, for example, `{{nobots|allow=HagermanBot}}`. In short, `{{nobots}}` was a sign that users could place on pages to signal to certain bots that they were either welcome or not welcome to edit on that page, with no actual technical ability to restrict non-compliant bots from editing. A bot would have to be built such that it looked for this template and respected it; in the case of PockingtonBot, the BAG ruled that the bot would not be approved if it was not extended to incorporate this standard. This functionality was not implemented in PockKleanBot, as the controversy was settled by PockingtonDan giving up on the idea of the bot and promising not to develop or deploy it further.

Despite not being incorporated into PockKleanBot, the proposed template soon gained support among Wikipedian bot developers. Along with Farmbrough, Hagerman was one of the key players in developing the initial specification for `{{nobots}}`, along with Ram-Man, a member of the Bot Approvals Group and the creator of Wikipedia's first bot.<sup>31</sup> On 18 December, Hagerman announced that HagermanBot was now “nobots aware” on the template's talk page. It the first recorded bot to become what would later be called exclusion compliant, a term that Hagerman coined. After some confusion with semantics, the template was copied to `{{bots}}` and remained relatively stable for the next few months as it gained acceptance and increased use among bots. After HagermanBot, the next bot to be made exclusion compliant was AzaBot, which was created to leave user talk page messages for users who had participated in a certain specialized discussion when an outcome was reached. AzaToth submitted the proposal to the Bot Approvals Group on 20 December, which was approved by Ram-Man that same day. In his decision, Ram-Man asked AzaToth to make the bot comply with `{{bots}}`, requesting that an opt-out mechanism be implemented to “respect their wishes.” Ram-Man also asked for AzaToth to share the source code that made this mechanism possible.

AzaToth quickly wrote a seventy-five line function in the Python programming language that incorporated compliance with this new standard, publishing it to a page listing resources for Wikipedian bot developers. This was soon fine-tuned and reduced to a four-line snippet of code, which was then ported to five other programming languages. In this way, nearly any bot operator could copy and paste a function into their bot's code to achieve exclusion compliance. In the following years, members of the bot development community created software frameworks to facilitate bot programming, this code was eventually incorporated and enabled by default into many frameworks, like the apibot library introduced earlier. By 2010, when a bot developer created a table comparing various software libraries and frameworks for bot development, exclusion compliance was one of the core properties listed. Through the efforts of those in and around the Bot Approvals Group – especially Farmborough, Hagerman, and Ram-Man – exclusion compliance became a requirement the BAG instituted for many bots that went through the approval process. It was implemented first to settle existing controversies, but then eventually became a pre-emptive mechanism for inhibiting conflict between bot editors and the community. While exclusion compliance had not been formally incorporated into the Wikipedia policy on Bots, bot operators had to argue why their bot should not be required to

---

<sup>31</sup> Released in 2002, Ram-Man's Rambot created articles about U.S. cities and towns from census data, doubling the number of articles in Wikipedia in less than a month.



implement such features upon review by the BAG. Failure to implement exclusion compliance or opt-out lists soon became non-negotiable grounds for denying some bot requests.

Debates about newsletter delivery bots – which exploded in popularity as various editorial sub-communities called “WikiProjects” began to grow in 2007 – became a site of specification regarding this issue. Many bots were proposed that would automatically deliver a group’s newsletter or targeted message to all its members. When the first of these bots began operating, conflicts initially emerged between editors who felt they had received unsolicited spam and bot operators who thought they were providing a valuable service. Opt-out mechanisms were used to settle these disputes, although in many cases the bots already incorporated such features but did not make them visible to recipients. In response, a set of informal criteria was soon formed by members of the BAG to ease these proposals. One requirement was implementation of some opt-out mechanism, either via exclusion compliance or an opt-out list; another was including information about opting-out in each newsletter delivery. Such requirements settled many controversies between editors and bot operators, and soon, bot approval policies were updated to officially indicate that no newsletter bots would be approved by the BAG until they were proven to sufficiently respect the wishes of editors who did not want interference from such bots.

## 2.5 The banning of Betacommand

These requirements for exclusion compliance were only instituted for newsletter delivery bots, rather than all bots, due in part to the strong opposition from a bot developer, Wikipedia administrator, and BAG member named Betacommand. Betacommand would later go on to become one of the most infamous figures in not just Wikipedian bot development, but the project as a whole. A skilled and once-respected bot developer who adamantly believed that copyrighted images put the encyclopedia project in existential risk due to infringement lawsuits, Betacommand developed some of the most controversial bots in the project’s history, which

Key people <sup>[php.1]</sup> ↕	Name ↕	PHP Version ↕	Uses API <sup>[php.2]</sup> ↕	Exclusion compliant ↕	Admin functions ↕
<a href="#">Cobi</a>	<a href="#">wikibot.classes</a>	5	Yes	Yes	No
<a href="#">Chris G</a>	<a href="#">botclasses.php</a> ↗	5	Yes	Yes	Yes
<a href="#">Kaspo</a>	<a href="#">Phpwikibot</a>	Unknown	Partial	No	No
<a href="#">Jarry1250</a>	<a href="#">Wikibot</a>	5	Yes	Yes	No
<a href="#">Adam</a>	<a href="#">BasicBot</a> ↗	5 or 4	Unknown	Unknown	No
<a href="#">SQL</a>	<a href="#">SxWiki</a> ↗	Unknown	Yes	Unknown	Yes
<a href="#">Foxy Loxy</a>	<a href="#">PHPediaWiki</a> ↗	5	Yes	No	Yes
<a href="#">Edward Z. Yang</a>	<a href="#">Wikipedia Bot in PHP</a>	Unknown	No	No	No
<a href="#">GeorgeMoney</a>	<a href="#">Bot Framework</a>	5 or 4	Unknown	Unknown	Unknown
<a href="#">Sam Korn</a>	<a href="#">Pillar</a> ↗	5	Yes	Yes	Yes
<a href="#">Tisane</a>	<a href="#">Wikibot</a>	5	Yes	Yes	Yes

Figure 12: 2010 table of Wikipedia bot frameworks and libraries, with exclusion compliance as a key column (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

took increasingly extreme measures in automatically removing copyrighted images from Wikipedia and admonishing those who added them. Wikipedians in general were split on the need to remove copyrighted images, and a key moment leading up to the banning of Betacommand was the bot developer's refusal to implement exclusion compliance and provide an opt out list when it sent messages to those who uploaded copyrighted images. Instead, Betacommand had the bot repeatedly send multiple notifications to some opponents (who had uploaded images the bot developer sought to delete), including one case where an editor received 50 unsolicited notifications.

The exclusion compliance standard and the norms around bot development established in the HagermanBot controversy would prove crucial to this case, but such standards and norms did not seamlessly flow into Wikipedia's broader administrative dispute resolution apparatuses. Instead, the banning of Betacommand involved debates and discussions that lasted over two years and generated over half a million words of comments across a variety of spaces and processes in Wikipedia. The culmination of the Betacommand case further established norms, meta-norms, and precedents for the relationships between bot developers and other Wikipedians. The case was ultimately referred to Wikipedia's Arbitration Committee, who found that such a repeated refusal to implement an opt-out list was "unreasonable" and suggested that Betacommand was not able to participate in Wikipedia as a bot developer according to the project's broader understandings of consensus and collaboration. Part of ArbCom's "remedies" was for Betacommand "to establish an 'opt-out' list for BetacommandBot without imposing any unnecessary conditions on the right to decline to receive notifications."

Betacommand refused to comply with the various decisions remedies reached by ArbCom. The bot developer was first placed under "parole," required to gain a consensus for performing any task (manual or automatically made) that affected more than 25 pages to the Village Pump – a large and crowded discussion venue for Wikipedia-wide issues. Betacommand was also restricted from making more than forty edits over any ten minute period of time. After repeatedly violating this prohibition and making suspected automated edits about copyrighted images, Betacommand was restricted from developing or operating any bots in Wikipedia and given a "topic ban" against taking any actions related to copyright images in any way. Some of Betacommand's other less controversial bots were handed over to other veteran bot developers, and Betacommand initially left the encyclopedia project. However, Betacommand soon returned and repeatedly violated these prohibitions, running what were suspected to be unauthorized bots performing similar kinds of tasks related to copyrighted images. In response, Betacommand was given one of the project's few "community bans" in 2009, indefinitely forbidding the now-former Wikipedian from participating in the project in any capacity whatsoever. In 2011 and again in 2014, Wikipedians claimed to have found newly registered accounts that were suspected "sockpuppets" of Betacommand, making fast-paced edits in a way similar to Betacommand's old bots. After investigations conducted through the "SPI" (or Sockpuppet Investigations) process, these accounts were both found to be operated by Betacommand and were similarly blocked from editing any pages in the wiki in any way.

### **3. Conclusion**

In the previous chapter on bespoke code, I discussed the role that the individual bot developer (and the infrastructure they support) plays in the continued operation of a bot. Instead of just seeing a bot as an autonomous software agent that has some effect on Wikipedia, the previous chapter analyzed bots as software agents that must be developed and continually operated by people. In the case of HagermanBot, I further develop this argument about bots as human-machine assemblages by showing how HagermanBot is a team effort to solve a specific problem in Wikipedia around unsigned comments. The bot's software agent does all of the tedious work of signing and timestamping comments, but the bot's operator ensures that it continues to operate, just like in my account of AfDStatBot. Yet Hagerman had to do more than the infrastructural work of keeping the server hosting the software agent online. He also had to negotiate with other Wikipedians to ensure that the software agent wasn't banned from Wikipedia due to the objections of those who strongly disagreed with the task it was performing (or if not banned, contributed to a growing resentment of bots, characterized as 'botophobia'). This case expands my analytical frame for studying bots as more than their source code further than in the previous chapter, by discussing how the collective entity of a software agent and its human operator interacts with the rest of the Wikipedian community.

The case of HagermanBot further illustrates that the negotiation of a bot's source code is not necessarily a purely deliberative process, such as one where Wikipedians first discuss the kind of norms they want to be algorithmically enforced, then build software agents to implement those decisions. Instead, this case shows the strengths of analyzing bots as ongoing projects in which developers and non-developers participate in the construction of a common worldview, using both computational and communicative means to express themselves. The history of exclusion compliance post-HagermanBot calls attention to how these computational and communicative modes of expression can be mutually co-constitutive, building on each other as these Wikipedians increasingly specified out how such a standard would operate for bots and Wikipedians.

This case also chronicles a turning point in Wikipedians' own understanding of what it meant to delegate tasks to a bot and a bot developer. With the rise of exclusion compliance, certain programmatic features of automated software agents came to be seen as expressing a normative ideal about how bots ought to act in relation to the community. The persistent notion that 'bots are better behaved than people' – which Hagerman first implemented in the form of a proprietary opt-out list – became standardized in the `{{bots}}` template, which could be used to express an intent to opt-out of any bot. Compliance with this standardized template came to be articulated in AzaToth's software code, which was translated into a number of programming languages such that any bot operator could easily make their bot articulate this notion of respect. Including this code gained the moniker of "exclusion compliant" and this condition became regularly incorporated into BAG bot approval discussions. Other bespoke code which structured the operation of bots began to incorporate the concept, from the infoboxes bot developers were to use to describe features of their bot to the software libraries built for bot development that made bots exclusion compliant by default. Such a concept then became even more cemented into Wikipedia's policy environment when Betacommand was ultimately banned for failing to incorporate some kind of opt-out procedure along these lines.

Finally, the HagermanBot case is a compelling way to introduce a different way of conceptualizing automata, going beyond the trope of technology as independent agents that technocrats create, then unleash on society to some effect (intended or unintended). Bots in Wikipedia are ongoing projects that can provoke substantial reflection, negotiation, and collective sensemaking about what the software agent will do, how it will do it, and why what it is doing matters – and even occasionally provoking the meta-level questions about who gets to answer those questions. This case problematizes discourses of automation which alternatively celebrate or lament a ‘rise of the machines’ trope, casting bots and humans as separate and distinct competitors. As I critiqued in the previous chapter, such arguments typically rely on this firm distinction to make two kinds of arguments about automation: first, automata replace humans, making humans obsolete; second, automata ruthlessly enforce rigid procedures on humans, dominating them. Yet far from being the kinds of rogue autonomous agents that much of the literature on technology and automation imagine, the *algorithms-in-the-making* approach I take emphasizes how bots are one of the many ways in which Wikipedians continually work out what kinds of governance work they think ought to be performed in the project. Bot development in Wikipedia is not an isolated, separate part of the project that impacts it from behind-the-scenes. Rather, it is tightly linked to broader governance and moderation practices across Wikipedia.

Ultimately, bots defy simple categorizations: they are both humans and software, social and technical, ideological and material, as well as assembled and autonomous. One-sided determinisms and constructionisms, while tempting, are insufficient to fully explain the complicated ways in which these bots have become vital aspects of how Wikipedia works. The case of HagermanBot and exclusion compliance initially tells the kind of “artifacts have politics” (Winner, 1986) story about how a technocrat’s understanding of the world controversially reified through automation, but there is a more nuanced dynamic at play. In taking an algorithms-in-the-making approach, the question of “who or what is in control of Wikipedia?” rests upon a more complex question of how control operates across a diverse and multi-faceted socio-technical environment. In exploring moments of controversy over the delegation of work to automated software agents, bots like HagermanBot come to appear more fluid and indeterminate – at least for the people who were able to raise such objections in Wikipedia’s administrative apparatuses. For many Wikipedians, bots are not so much ruthless automata taking over their project, but instead sites for explicating and negotiating their ideas about what work is to be done and how that work ought to be done.

## **Section 2: Specialized administrative processes in Wikipedia (Chapter 4, 5, and 6)**

### **1. Summary of section 1 (chapter 2 & 3)**

The chapters in this section expanded the analytical frame of bots far beyond the standard focus on the source code of an automated software agent. In chapter 2, I discussed my own experiences as a bot developer through the concept of bespoke code to illustrate how the bot's operator plays a crucial role in keeping a bot operational. Put simply, a bot operator who is no longer able to support a bot's computational infrastructure is no longer a bot operator. Bots are therefore a project, and one that extends far beyond the initial design and development of software code. Even "completed" bots are a team effort by a bot's software agent (which performs the tasks according to its source code) and a bot's operator (who ensures that the software agent is able to perform such tasks). This analytical frame helps us think about many issues around governance and code, such as how bot operators far more flexibility and autonomy in developing and deploying code than in traditional server-side software, but also have more responsibility in continually operating and maintaining the bot's computational infrastructure.

In this chapter, I then discussed the case of HagermanBot and exclusion compliance to further expand this notion of what the bot is as a kind of project. In this expansion of the analytical frame, a bot is a project to decide what kinds of tasks ought to be automated and how a bot ought to go about performing those tasks. In Wikipedia, these decisions are certainly not limited to the bot's operator, due to the existence of an administrative apparatus that requires all bots be approved by the Bot Approvals Group. While Hagerman, as the bot's operator, was the only person with access to modify the bot's source code and change the behavior of the corresponding software agent, others heavily participated in the broader project of producing a shared understanding of how HagermanBot ought to go about performing its somewhat controversial task. The compromise that resolved the HagermanBot controversy produced a new set of norms, standards, source code, and policies – all of which were aligned around a common set of ideas about what the Wikipedia community of editors was and ought to be. When these ideas were repeatedly violated by a later Wikipedian who operated automated software agents without building a consensus for the tasks they performed, the software agent and its developer were ultimately banished from Wikipedia.

These two chapters sketch out a way of understanding bots as one of the many different ways in which Wikipedians work out various ideas about what they want their community and their encyclopedia to be. A bot typically has only a single human operator (some major bots are run by teams), but every one of Wikipedia's hundreds of currently-active bots is made possible through an effort that extends far beyond the bot's operator. This is not to diminish the role of the bot's operator, but rather to call attention to how the work of programming a bot's software agent is tightly bound up in deciding what kinds of tasks ought to be performed and how. Even a bot that performs a task which is universally seen as "obvious" only has such an uncontroversial status because it rests upon a previous consensus that had been built in support of performing that task in the way that the bot performs it. For bots that are not so "obvious," the work of building a bot includes building a consensus for performing the task at hand – which can take far more work than programming the bot, as Hagerman found. This argument also calls

attention to how the bot operator can serve as an intermediary between opposing factions, working to resolve the tensions between different, possibly incompatible ideas about what Wikipedia is and ought to be.

At a broader level, this way of looking at bots as projects of collective sensemaking is how I take an algorithms-in-the-making approach to bots in Wikipedia. As I reviewed in previous chapters, much of the ‘critical algorithms studies’ literature focuses on algorithmic systems once they have already been designed, developed, and deployed. The goal of such scholarship is often to ‘open up the black box’ by seeking ideological assumptions or power relations that are ‘embedded’ in code; code is instrumental, a means to exercise technocratic control. My algorithms-in-the-making approach comes on the scene much earlier, which calls attention to a different set of issues. Lessig’s “code as law” (Lessig 1999; 2006) metaphor – which has become a dominant way of understanding how code operates as a form of governance – becomes something quite different when looking at algorithms-in-the-making. For Lessig, code is law because code is enforced with the unquestionable authority of the sovereign, having a force that he suggests is as powerful as the laws of physics. Yet with Wikipedia’s bots, code is law in that it is more like a medium in which people express and negotiate issues of public policy. Taking an algorithms-in-the-making approach to bots like HagermanBot is useful because it provides a way to empirically investigate the broader assumptions, values, and priorities that are expressed and negotiated through the programming of algorithmic agents.

Even something as seemingly minor as ensuring that comments in discussion spaces are signed and timestamped are bound up in strong normative visions about how and why Wikipedia works – such as the belief expressed by Hagerman and his allies that communicative practices like signatures and timestamps were important parts of Wikipedia’s discussion and consensus based editorial model. Yet this vaguely-Habermasian ideal had to be reconciled with competing worldviews that valued editorial autonomy, respecting the individual editor’s choice to not engage in those widely-accepted communicative practices if they so desired. Ultimately, the compromise that emerged was not just a consensus about the necessity of signatures and timestamps in comments; it was about how members of a community ought to interact with each other in a common space. The analytical frame of the bot as an ongoing sociotechnical project is useful when thinking about the many different Wikipedia bots that support specific individual tasks related to writing and discussing encyclopedia articles, like signing unsigned comments, flagging articles with broken links, fixing spelling mistakes, removing spam and vandalism, or updating articles using structured databases. For example, spellchecking bots may seem like an obvious and uncontroversial task to perform, until the question is asked about what national variety of English will be used for its dictionary – an issue that raises many high-level questions about what Wikipedia is as an encyclopedia and a global community of editors. On this issue, I do not wish to imply that all of Wikipedia’s high-level normative issues are resolved exclusively through bots. Rather, I emphasize that bots are one of many core ways in which Wikipedians make such decisions about what Wikipedia is and ought to be.

## 2. Overview of section 2

To further explore the role of bots as projects of collective sensemaking and decision-making, the chapters in this next section explore more cases about how bots are projects in which Wikipedians work out more-and-less shared understandings about what Wikipedia is and ought to be. In these chapters, I move away from bots that perform more specific tasks about writing and discussing individual encyclopedia articles and instead discuss bots that operate in Wikipedia's specialized administrative spaces. These administrative processes are formalized spaces that exist largely behind the scenes of Wikipedia, where veteran Wikipedians make decisions about which articles ought to be deleted or protected from public editing, which editors ought to be banned from editing for violating policies, which articles ought to be featured on the main page, and more. Just as HagermanBot was a project to build a consensus around a particular normative understanding of how Wikipedians ought to participate in discussion spaces, these "clerk" bots (as some of their operators call them) are projects to build a consensus around a particular normative understanding of how Wikipedians ought to participate in specialized administrative processes. In order to discuss the role of bots and other bespoke code in these spaces, I first intentionally de-center bots and instead detail these administrative processes using a variety of ethnographic, archival, and statistical methods. After introducing various aspects of the historical development and everyday operation of these administrative processes, I then discuss the role that bots have played in making this way of governing contributors possible.

In giving these accounts of Wikipedia's administrative processes, I further show how bespoke code can be a medium in which developers and non-developers work out and implement ideas about what kinds of decisions they want to make and how they want to make them. The impact of such bots is not only in the kinds of work their software agents have been delegated (which is certainly substantial). In addition, I argue that we must also look to how certain software agents are authorized and accepted by Wikipedians as furthering the mission or spirit of the encyclopedia project, while others are rejected and banned for not aligning with these ideas. In order to show how bespoke code supports certain ideas about what Wikipedia is, I relate various historical and contemporary accounts of bots as they are situated in relation to existing editorial and administrative practices in Wikipedia. The history of bots in Wikipedia is decidedly **not** a story of machines replacing and displacing humans – as is often imagined in discussions of automation – and so to present bots at the front and center of these accounts would be somewhat misleading. Rather, bots come on the scene as one of many different, simultaneously existing ways in which Wikipedians work out what they want their project to be.

Throughout Wikipedia's 15 year history, processes for making decisions about commonly reoccurring issues have been continually developed by Wikipedians to deal with problems of scale, which many Wikipedia researchers have detailed (Bruckman & Forte, 2008; Halfaker et al., 2013; Konieczny, 2010; Pike, Joyce, & Butler, 2008; Tkacz, 2015; Wattenberg et al., 2007). The core contribution I make is in showing how these processes are frequently supported, extended, and contested through the development and deployment of bots and other bespoke code. To make this argument, I first ethnographic, archival, and statistical methods to document the prevalence of these formalized administrative spaces in Wikipedia, then discuss

how bots and other bespoke code are an integral part of how these administrative spaces operate. Processes like “Articles for Deletion” or “Requests for Page Protection” are often invisible or opaque to non-Wikipedians, particularly newcomers, who often find themselves thrust into a space that operates quite differently than the anarchistic “anyone can edit” reputation that Wikipedia has been attributed in mass media over the past 15 years. As many researchers and commentators of Wikipedia have noted, the encyclopedia project has slowly developed more and more bureaucratic-like structures to deal with issues of scale, particularly as Wikipedians’ priorities shifted towards quality and reliability.

I extend this literature by giving specific mechanisms to this account, using a focus on the articulation work delegated to automated software agents. I give an overview of these specialized administrative processes, showing how prevalent they are in Wikipedia today and how bots support this work in particular ways and not others (chapter 4). Then I discuss how such bot-supported processes have become part of Wikipedia’s organizational culture (chapter 5): they are one of many ways in which veteran Wikipedians interact and relate to each other, and familiarity with such processes is learned as a part of becoming a member. Finally, I discuss the history of one of the longest administrative processes in Wikipedia, in which Wikipedians decide which articles ought to be deleted. I detail how this process was first established, and then show how various forms of articulation work began to be delegated to clerkbots. In all, these three chapters give specific ways of taking an algorithms-in-the-making approach to Wikipedia, showing how bots are ongoing projects that are deeply bound up in what it means for Wikipedia to be both an encyclopedia and a community dedicated to curating that encyclopedia.



## **Chapter 4: Articulation work**

### **1. Introduction**

#### **1.1 Chapter overview**

This first chapter of the second section introduces the specialized administrative processes that are now dominant in Wikipedia, then shows how they are supported through bots and other bespoke code. I first show how these centralized spaces are a core way in which Wikipedians perform commonly reoccurring tasks, including making decisions about particular encyclopedia articles as well as issues that span the encyclopedia project. I give an overview of these administrative processes using a variety of ethnographic, archival, and statistical methods, then discuss them in the context of the bespoke code that has been designed, developed, and deployed to support such processes of decision-making in certain ways and not others. Finally, I discuss these processes and the bespoke code that support them using the concept of “articulation work,” which refers to all the “supra-work” that takes place as a group performs a particular task: dividing a task into smaller tasks, allocating tasks to different members, recomposing completed tasks into a common whole, keeping members on task, giving context and support, and so on. Articulation work is a further specification of the analytical frame of the bot, which the previous chapters have established as an ongoing project in negotiation and collective sensemaking about community norms and ideals (not just the individual delegation of work from a human developer to a software agent). This focus on articulation work is a core way in which I further develop the algorithms-in-the-making approach, as it gives analytical specificity to how bots are able to serve as a focal point for Wikipedians to work out these kinds of high-level normative issues and concerns.

##### **1.1.1 Methods for studying Wikipedia’s administrative processes**

In the first part of this chapter, I introduce and discuss Wikipedia’s administrative processes using a broad set of methods and approaches, which are necessary to capture the diversity of the many different kinds of phenomena at work in Wikipedia. Throughout this chapter, I present reviews of existing empirical research on Wikipedia, ethnographic texts from my experience as a veteran Wikipedia, and descriptive statistics I generated using the Wikimedia Foundation’s analytics cluster. I begin with an introduction to these kinds of administrative processes using a contemporary vignette that speaks across the issues in this chapter and the follow two chapters. I relate a case in 2015 in which I responded to a request from someone who wanted to know how they ought to go about performing a particular kind of task. As a highly-active veteran Wikipedian, I was excited to extensively detail what they ought to do in order to ensure that they edited Wikipedia in the ‘correct’ way. The response I received made me reflect about the way my status as a Wikipedian shapes how I not only participate in the encyclopedia project, but understand what Wikipedia is a site of cultural production. Following this vignette, I present my own reflection as a highly active veteran Wikipedian as to why I personally believe that such a structure is valuable within the community’s existing set of values and ideals – rather than simply being a difficult waste of time, as they are sometimes critiqued.

After presenting my own experience speaking to these processes, I then review previous research exploring processes in Wikipedia. Many scholars have studied various important specific processes, and such literature gives great context into the contemporary (or contemporary at the time of their research) operation of such ways in which Wikipedians make decisions. I extend this literature by theorizing this phenomenon more generally in the context of the bespoke code in which it is supported. Next, I conduct a quantitative analysis indicating how far more text is added to Wikipedia's largely bespoke administrative processes than both encyclopedia articles and article-specific talk pages combined. I further quantitatively explore the contributions of registered bot accounts across various spaces in Wikipedia, finding that in terms of text added, they have far more of a proportional presence in these behind-the-scenes spaces than in other aspects of the encyclopedia project. I then give an overview of some of the specialized processes in which bots are most active. I first present a quantitative analysis of the number of bot edits made to various pages in an administrative process, then I describe how some of these pages operate using my ethnographic experience as a veteran Wikipedian.

### **1.1.2 Discussion: bots as delegations of articulation work**

The goal of this chapter is to use a variety of methods to introduce these specialized processes and venues within the context of the bespoke code that supports them, then discuss these bots as bound up in a kind of “articulation work” (Strauss, 1985; Suchman, 1996). Articulation work is a concept that has been frequently discussed in studies of IT in organizations and in the Science and Technology Studies literature, and this concept calls attention to the work involved in specifying how a task is to be accomplished. There is much discussion about automated software agents that are (ostensibly) independently making decisions that have profound societal impacts, whether this in the filtering of Facebook's news feed (Bakshy et al., 2015; Tufecki, 2015) or in predictive policing (Doctorow, 2009). However, my algorithms-in-the-making approach shows that at least in the case of Wikipedia, algorithmic systems must be understood in the context of the broader sociotechnical systems in which they operate. These bots in Wikipedia are typically not delegated autonomous, independent decision-making authority; they are instead delegated articulation work and are more subtly woven into the fabric of Wikipedia's administrative apparatuses. Clerk bots perform articulation work in that they specify particular ways in which Wikipedians are to participate in specialized administrative processes, just as clerks in courts of law specify particular ways in which individuals can participate in the legal system. Someone who participates in an administrative process in a way that the clerk bot cannot ‘see’ risks having their actions ignored by Wikipedians who rely on these software agents to aggregate and consolidate actions into orderly lists, queues, and stacks. Because such bots are projects to delegate articulation work (and not so much the decisions about which pages to delete, which users to ban, etc.), they can provoke and resolve debates about how Wikipedians ought to collectively make decisions.

Strauss's concept of articulation work and its subsequent application in the fields of Computer Supported Cooperative Work draws attention to several issues that I address in subsequent chapters in this section on administrative processes. Chapter 5 discusses the invisibility of articulation work, which speaks to issues of socialization and membership. Many administrative processes are difficult for newcomers to locate and participate in, particularly because many were created for veterans to quickly and efficiently make decisions in the face of

an exponential growth of allegedly low-quality contributions by non-veterans. However, I also discuss initiatives seeking to use these same kinds of strategies to build safe spaces for mentorship and newcomer socialization. In Chapter 6, I discuss the history of administrative processes around the deletion of unencyclopedic articles, showing how the process came to be increasingly formalized and specified as bots and other bespoke code were developed and deployed to perform articulation work. Human Wikipedians still make the core decisions about whether articles ought to be kept or deleted, but they are supported by bots and other bespoke code, which works to support this decision-making process in particular ways and not others.

## **2. Bots behind the scenes in Wikipedia**

### **2.1 The edit request process**

#### **2.1.1 Opening vignette: a simple request**

One day, I received an e-mail from a professor I know at Berkeley who wanted some advice about the article written about him on the English-language Wikipedia. The article that others wrote about him was factually accurate, but there were some things he thought were oddly worded and might need to be rephrased. The professor wasn't sure if he wanted to edit the article – he knew there might be conflict of interest issues in editing his own Wikipedia article, but also he thought he might actually prefer the odd wording. But either way, he asked me how someone in his position would go about correcting their own Wikipedia article, if that was something they wanted to do. This professor knew quite a bit about Wikipedia – enough to look through the revision history and learn who added the awkward phrases and when, and enough to know that editing his own article might not be appropriate. Yet like most people who ask me for advice on Wikipedia, he didn't know enough about the project's internal norms and processes to know for sure how to proceed – which is why he came to me, someone who has both studied various aspects of Wikipedia's internal operations, as well as a long-time Wikipedian, a member of this self-identified “community.”

When I first read the professor's e-mail, I got excited and sent a long reply about how exactly he should proceed if he did want to make a change. First, I told him he should register an account that identified him with his legal name, as Wikipedians generally prefer people with conflicts of interest to declare them outright. Then, he should go to the designated “talk” page for the article, edit the talk page, and add a new message that identified the issue and suggested what he thought was an appropriate change. Finally, because this professor's article was likely not the most visited page in Wikipedia, he should add a special trace to make sure someone saw that request: at the top of his message, he ought to add `{{request edit}}`. This special text (called a template) would be parsed by an automated software agent named AnomieBot, which was continually scanning for new talk page messages containing this template text. In a matter of minutes, when the bot's script next ran on a server cluster in Ashburn, Virginia, it would summarize the professor's edit request and put it into a centralized queue in a special administrative space in Wikipedia, alongside all the other pending requests of this same type.

The bot would provide some information about his request, such as the date of the request and if the page is locked down from public editing, which it displays more prominently in yellow (Figure 16).

Sometime later (hopefully not too much later), a Wikipediaian who was looking for something to do would check that list, navigate to the talk page for the professor’s article, and give their thoughts about whether the edit request was a good idea or a bad idea. This person would probably be someone who spent a lot of their time on the English-language Wikipedia responding to edit requests, rather than any number of other tasks they could possibly do. If the Wikipediaian agreed with the professor (which would be likely, I thought) they might edit the article themselves to fix it. Or they might also simply tell the professor to go ahead and implement the changes himself. However, if the Wikipediaian didn’t agree, they would explain why, and there might be a longer discussion between the two of them on the talk page about the issue. With any decision made, the Wikipediaian should leave a slightly different template based on what their decision was, so that the bot would remove the professor’s edit request from the main queue and place it into an archive. I thought about telling the professor that if he found himself getting into a conflict with the Wikipediaian who responded to his request, he could add `{{rfc}}` to the talk page to more broadly request the comments of other Wikipediaians. However, that “Requests for Comments” process was reserved for more controversial issues, as many more Wikipediaians monitored that specialized venue on a regular basis. I didn’t think it was likely he would get into such a controversy, so I left that out.

### Current requested edits [\[edit\]](#)

87 COI edit requests <span style="float: right;">v · h</span>			
Page	Tagged since	Protection level	Last protection log entry
<a href="#">Mitch Caplan (request)</a>	2015-05-19 15:51	Not protected <a href="#">(log)</a>	
<a href="#">Rick Ross (consultant) (request)</a>	2015-05-19 10:40	Semiprotected <a href="#">(log)</a>	Protected by <a href="#">Brandon</a> on 2011-06-20: "Violations of the <a href="#">biographies of living persons policy</a> "
<a href="#">Battle of Ramadi (2014–15) (request)</a>	2015-05-19 04:20	Not protected <a href="#">(log)</a>	
<a href="#">User:ALNNAL (request)</a>	2015-05-18 21:16	Not protected <a href="#">(log)</a>	
<a href="#">Tipper Gore (request)</a>	2015-05-18 20:30	Not protected <a href="#">(log)</a>	Protected by <a href="#">Alison</a> on 2011-04-08: "Violations of the <a href="#">biographies of living persons policy</a> "
<a href="#">LinkedIn (request)</a>	2015-05-18 15:51	Not protected <a href="#">(log)</a>	

Figure 13: The top of the edit request list as of 19 May 2015, with 87 requests pending. (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

While I was quite enthusiastic about explaining this highly-structured and automated administrative process, the professor’s reaction couldn’t have been more different to mine. The professor – as someone with substantial experience authoring many different kinds of reference works, who simply wanted to have some say about how his Wikipedia article was worded – was immediately disinterested by the amount of effort that would be involved for such a trivial change. Reflecting on this interaction, I realized my initial enthusiasm for sharing the edit request process with the professor cannot be divorced from my status as a Wikipedian, where I encounter these kinds of bot-supported administrative processes on an average, everyday basis. Such processes are not just used to respond to edit requests from people with conflicts of interest, as there are hundreds of specialized processes and venues in the project dedicated to a variety of different tasks.

### 2.1.2 My personal reflection on such processes

Interacting in these specialized spaces is a core mode of participation for many of the highly-dedicated Wikipedians who spend dozens of hours a week working behind the encyclopedia project’s proverbial curtain.<sup>32</sup> I know this as someone who not only edits encyclopedia articles, but also is part of a much smaller, tight-knit, and less-visible set of people in the project who collectively refer to themselves as “the community” – with all the insider/outsider implications such a term implies. As one of the insiders, I recognize all the elements involved in this aspect of governance work: the template traces on an individual page linking up to a centralized, bot-curated noticeboard; a set of broad principles applied by whoever decides to take the next issue on the queue. They individually and collectively make sense to me in a way they likely do not to those who do not spend a substantial portion of their lives behind the scenes of Wikipedia. This kind of process has become not only natural and routine to me, but something I treat with much meaning, significance, and reverence, even if I disagree with certain aspects about how it operates (and I certainly do).

From my insider perspective, I wrote out my own justification of the process and shared it with other Wikipedians – not to convince the professor about its superiority or necessity, but to make my own assumptions about the algorithmically-supported administrative processes explicit. The first layer is that of what Wikipedians call *policy*, which includes (but extends beyond) documenting what kinds of actions are and are not prohibited. Wikipedians have come to a general agreement about how to handle similar kinds of cases and issues, although this certainly evolves over time. Even if I personally disagree with certain aspects of how the community has decided how those cases are to be handled, I know that I ought to act according to those general principles.

---

<sup>32</sup> Wikipedians spend a substantial amount of time contributing; a 2011 survey ran by the Wikimedia Foundation found that 29% of respondents reported spending 1-2 hours a day contributing to Wikipedia, 14% reported 3-4 hours per day, and 6% reported more than 4 hours per day. ([van Liere and Fung 2011](#))

For example, the community already has a generally shared belief that it is usually a bad idea for people to edit Wikipedia articles about themselves.<sup>33</sup> However, the existence of the conflicts of interest policy doesn't automatically mean that people will adhere to this policy, particularly those who do not know the rules at all. As an editor, I constantly see new and unregistered accounts violating Wikipedia policy in the course of doing the work that I think is important. I then feel obligated to make things right, but I also want to get back to what I had originally come to an article to do. The existence of policy pages is already a big help in doing this, as I can just link to WP:COI and send them a short message telling them to read it and comply. However, those kinds of issues often aren't solved with a 'read the f-ing manual' approach; they often require extensive back-and-forth conversations. Furthermore, given the extensive division of labor and specialization that takes place in Wikipedia, I know that I'm not the best person to talk about conflict of interest with a newcomer. I probably am one of the better people to talk to about article deletions or references, but I don't keep up with the WP:COI policy nearly as much as I do other areas of the encyclopedia. And it does take work to keep up with a policy – not only the current text, but also the way that it is canonically interpreted and applied.

This is where the second layer of *process* comes in. Wikipedians have developed a wide variety of centralized spaces and standardized processes for coordinating all different kinds of work that reoccurs in the project, even those that don't involve such yes-or-no decision-making: copyediting, adding references, merging duplicate articles, deleting un-encyclopedic articles, blocking problematic users, resolving interpersonal disputes, and many, many more tasks (Bruckman & Forte, 2008), which make up what other Wikipedia researchers have called "the hidden order of Wikipedia" (Wattenberg et al., 2007). Because there is a project-wide policy about how to deal with common issues like conflicts of interest, and because there are people who specialize in those particular issues, then it is helpful for veterans to have some kind of centralized space to aggregate all of those same kinds of decisions. This is helpful both for those who specialize in a particular area (who have a common site for working these issues out) and for those who do not (who have a way to send relevant issues to specialists). So when I want to do work on article deletion decisions, it is helpful to have a single page where I can see all the articles currently nominated for deletion, with an efficient way for me to participate in as many of those discussions as I want. But since I often don't feel like working out conflict of interest issues, it is also helpful for me to have some kind of process to send that editor to a veteran Wikipedian who does want to work on conflict of interest issues. Correspondingly, when a veteran Wikipedian who cares about conflict of interest but not article deletions encounters a potentially non-notable article, they can send it to a space that I and other deletion specialists will notice.

The final layer is *infrastructure*. If there is a project-wide process for a commonly reoccurring task, where similar kinds of issues about individual articles can be brought to a single space where those who specialize in such kinds of issues can make decisions about them,

---

<sup>33</sup> Which is something that I personally do not believe is as large of an issue as many other Wikipedians do, but I understand that I am in the minority on this issue. Coming to terms with the fact that the "community consensus" may not always coincide with your own views is another part of becoming a Wikipedian.

then in Wikipedia it just makes sense to me that this process will be supported using a particular set of infrastructure. It doesn't make sense to re-invent the wheel, so the process will be not only be made up of the policy as a specification of more abstract norms and principles, nor the policy as a particular procedure for making decisions in accordance with those more abstract norms and principles, but a material-semiotic assemblage (Law, 2008) that is easily recognizable to many veteran Wikipedians. It will likely be comprised of the following heterogeneous entities, which will be linked together to constitute a single case. The first element is a wiki page, which are the source of the commonly reoccurring issue. These can be articles, talk pages, user pages, or other processes, but in the edit request example, it is the talk page for the professor's article containing a request to be answered. The second element is a template script, where editors can quickly tag articles or talk pages by adding those curly-bracket codes. In the processors case, it is `{{edit request}}`. The third element is a centralized, designated wiki page in a special section of Wikipedia for discussing a particular kind of issue, sometimes called "noticeboards." Then the last element is some kind of automated software agent that will scan in near real-time for recently tagged or untagged pages and update the centralized page accordingly, possibly adding some contextual information. In this way, any commonly reoccurring issue can therefore be escalated from any page in Wikipedia to a specialized process. Perhaps if you were to redesign Wikipedia's platform from scratch, it might be a good idea to improve the user interface behind this page-template-noticeboard-bot assemblage, but the "stock" version of MediaWiki is notoriously thin, which is why bots and other bespoke code have come on the scene to make these particular kind of administrative processes possible.

## **2.2: Previous literature and popular commentary on Wikipedia behind the scenes**

### **2.2.1: Editing articles versus editing discussion pages**

In Wikipedia, automated software agents perform a significant amount of work in writing, curating, and policing encyclopedia articles, as a number of academic and popular accounts have emphasized. ClueBot NG, a quality control bot focused on "counter-vandalism," has made over 2 million edits on Wikipedia as of July 2015, reverting about 2,500 to 5,000 unconstructive edits every single day (Geiger & Halfaker, 2013). Ram-Bot, the first known Wikipedia bot developed in 2002, nearly doubled the number of articles in fledgling encyclopedia by scraping public domain census data and creating an article about every city and town in the United States (Ayres et al., 2008, p. 8). Lsjbot, developed by a Swedish Wikipedian, is responsible for authoring 2.7 million articles in a variety of non-English languages, similarly scraping databases on living creatures and towns (Jervell, 2014). Scholars have focused heavily on the legal and cultural implications of automatically-generated encyclopedia articles, often arguing that these bots demand that we rethink notions of authorship and accountability. Kennedy (2010) argues that Wikipedia's article writing bots illustrate something literary scholars have acknowledged for decades: "writing happens as an interactive process that involves exchanges between multiple agents, texts, and influences" (Kennedy, 2010, p. 308).

While these article writing bots are certainly important for Wikipedia, Kennedy's argument about the need to examine the interactivity of writing applies even stronger when

examining a less visible type of automated software agent in Wikipedia: bots that operate in the project's specialized discussion spaces and administrative processes. As qualitative and quantitative researchers of Wikipedia have long demonstrated, there is a substantial amount of coordination and administrative work which takes place behind the scenes in Wikipedia. Many Wikipedia researchers have focused on the "talk pages" associated with every individual encyclopedia article, accessed by clicking the "talk" link in a tab at the top of the article, next to the "edit" link. (Laniado, Tasso, Kaltenbrunner, Milano, & Volkovich, 2011; Schneider, Passant, & Breslin, 2011; Viegas, Wattenberg, Kriss, & van Ham, 2007). In addition to these article-specific spaces for discussing what the content of each article should be, there are also specialized project-wide spaces where commonly reoccurring decisions are made.

For example, decisions about whether an article ought to be deleted for being unencyclopedic or protected from public editing due to excessive vandalism take place in their own dedicated venues, rather than each article's talk page (Taraborelli and Ciampaglia 2011; Lam, Karim, and Riedl 2010; Schneider, Passant, and Decker 2012).<sup>34</sup> Researchers have also studied the processes for promoting Wikipedians to administrators, which gives a set of technical privileges in the MediaWiki software and social obligations in the community (Burke & Kraut, 2008; Derthick et al., 2011) as well as processes for reviewing the quality of articles (Morgan, Gilbert, McDonald, & Zachry, 2014). These venues make up what some researchers have called "the hidden order of Wikipedia" (Wattenberg et al., 2007), supporting the more visible work of writing and editing encyclopedia articles. As Wikipedia has expanded in article content, coordination and administrative work has outpaced the work of writing encyclopedia articles – a trend beginning in 2005 as Wikipedians shifted and began to focus more on article quality than quantity (Halfaker et al., 2013b).

In discussing the project's specialized venues and processes with veteran Wikipedians, I consistently hear that while these processes may be complicated and somewhat impenetrable to newcomers, they are crucial to the project's continued operation in its current state<sup>35</sup>, both when it comes to making decisions about the content of articles and broader issues. Figure 17, a table of pages relevant to the "Wikipedia community," includes some pages on policies, guidelines, and tutorials, but it includes far more specialized spaces, including processes, groups, noticeboards dashboards, and indexes. The assemblage of norms, processes, templates, categories, queues, centralized pages, and bots I discussed in the edit request process is found in dozens of similarly-structured processes that operate across the project. Wikipedia researchers have documented and researched a variety of issues relating to these specialized processes and venues – which Konieczny (2010) has labeled an "adhocracy," drawing from the term coined by Toffler (1970).

Such research generally argues that these kinds of decentralized structures are how highly-active contributors to the "anyone can edit" encyclopedia project work to implement and enforce a more unified understanding of what an encyclopedia ought to look like and how it

---

<sup>34</sup> When an article is involved in one of these specialized venues, it is standard practice to place a notice on the article's talk page (and sometimes at the top of the article itself) to direct editors to this venue.

<sup>35</sup> However, I have talked to many Wikipedians who strongly disagree with the status quo for a wide variety of reasons, arguing that the highly-automated bureaucracies need to be reformed.



ought to be written. Processes exist for deciding which articles to delete for being on “unencyclopedic” topics (Geiger & Ford, 2011; Schneider et al., 2012; Taraborelli & Ciampaglia, 2010), which articles ought to be featured on the main page (Wattenberg et al., 2007), and which user accounts and IP addresses ought to be blocked from editing due to breaking the project’s policies (Geiger & Ribes, 2010a). Processes also exist to structure practices of dispute resolution, newcomer socialization, reporting violent threats to law enforcement, vetting potentially unreliable references, locking down pages due to “edit wars” and vandalism, and more. In fact, just in the English-language Wikipedia alone, there are so many different kinds of specialized processes and venues supported in this way that even I do not know them all off hand. To my knowledge, hardly any Wikipedians do, just as no single person can even come close to having read all the encyclopedia articles and discussion pages in the project. Furthermore, like any organizational structure or bureaucracy, these processes are subject to revision and reconfiguration over time. New processes are developed, sometimes split off from an existing process that is too crowded, and occasionally a process stagnates into obscurity.

<span>V</span> · <span>T</span> · <span>E</span>	<b>Wikipedia community</b>	[hide]
<b>Welcome to Wikipedia</b>	Community portal (Bulletin board) · Regional notice boards · News (The Signpost · Goings-on · In the media) · Meetups · Mailing lists · Awards (Reward board) · Statistics (Milestones) · The Wikipedia Library · Centralized discussion · Village pump (Idea lab · Policy · Proposals · Technical · Miscellaneous)	
<b>Contents and grading</b>	Requested articles · Most-wanted articles · Images needing articles · Articles needing images · Articles for creation (Creation · Help) · Vital articles · Today's articles for improvement · Peer review · Good article nominations · Featured article candidates (Lists · Pictures · Portals · Topics) · Article translation (Pages) · Main Page (Errors)	
<b>WikiProjects and groups</b>	Directory · Culture and the arts · Geographical · History and society · Science, technology and engineering · Wikipedia assistance and tasks · Patrols (Recent changes) · Counter-Vandalism Unit · Version 1.0 Editorial Team · Accessibility	
<b>Maintenance tasks</b>	Open tasks · Cleanup (Vandalism) · Database reports · 'As of' · Page Curation · Recent changes · Backlog · Controversial issues · Dusty articles	
<b>Administrators and noticeboards</b>	Dashboard (Admin) · Admin requests · Administrators' (Incidents · Edit warring · Vandalism) · Requests for closure · Revision deletion · Oversight (Request) · Page protection · Sockpuppets · User permissions · Open proxies · Usernames (Changing · Title blacklist) · OTRS · Bureaucrats' (Requests for adminship) · Arbitration Committee (Requests · Enforcement)	
<b>Content dispute resolution</b>	Editor assistance · Requests for comment · Third opinion · Dispute resolution noticeboard (Biographies of living persons · Conflict of interest · External links · Fringe theories · Neutral point of view · No original research · Reliable sources) · Mediation	
<b>Other noticeboards and assistance</b>	New contributors' (Teahouse) · Asking questions (Help desk · Reference desk) · Requests for help (Adopt-a-user · Online Ambassadors) · Copyright assistance (Copyright investigations · Text problems · Media questions · Non-free review) · Paid editors · Resource requests · Mergers (History mergers) · Moves (Page importation) · Spam (Blacklist · Whitelist) · Bot owners' · Education (Incidents) · New pages patrol · General sanctions · Editor sanctions · Long-term abuse	
<b>Deletion discussions</b>	Guide (Admin) · Today · Articles · Templates · Files (Possibly unfiled) · Categories · Redirects · Miscellany · Speedy · Proposed (BLP · Books) · Review (Undeletion) · Arguments to avoid · Arguments to make · Article Rescue	
<b>Directories and summaries</b>	Departments · Edit summary legend · Editor's index · Essays · FAQs · Glossary (Abbreviations) · Help · Manual of Style (Simplified) · Rules (Five pillars · Policies · Guidelines) · Shortcuts · Template messages (Sources · User talk) · Tips (Today) · Tools · Wikis · Wiki markup	

Figure 14: A list of specialized spaces and venues that can be found at the bottom of every page included in this list in an expandable table titled "Wikipedia community" (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

### 2.3. Quantifying behind the scenes coordination work by Wikipedians

In this section, I present statistics about the use of administrative processes and other specialized spaces, which I collected using the Wikimedia Foundation’s analytics server. These statistics indicate that a substantial amount of coordination work in Wikipedia takes place outside of the individual article talk pages attached to each encyclopedia article – which have long been studied as key sites of coordination, negotiation, and decision-making. This statistical analysis relies on two related aspects of Wikipedia that are not as common in some other online environments. First, almost all activity on the site takes place through editing wiki pages: discussing the content of encyclopedia articles takes place through editing “talk pages,” sending messages to other editors takes place through editing “user talk pages,” meta-level discussion spaces and processes about Wikipedia take place through editing the confusingly-named “Wikipedia pages,” meta-meta-level discussion about these specific meta-level pages takes place through editing “Wikipedia talk pages,” and so on. Second, these different kinds of pages are separated into distinct, non-overlapping “namespaces,” which is a feature Wikipedians built into the MediaWiki platform. A page in a particular namespace will be prefixed with the name of the namespace and a colon. Encyclopedia articles have no prefix and are represented in MediaWiki’s database as namespace “0.” For example, the project’s policy on conflict of interest is written at “Wikipedia:Conflict of Interest” (also linked as “WP:COI” for short) and discussions about this policy take place at “Wikipedia talk:Conflict of Interest.” One process for deciding whether articles ought to be deleted from Wikipedia takes place at “Wikipedia:Articles for Deletion,” which can also be accessed at the shortlink “WP:AFD.” Wikipedians rely on these namespaces to keep different kinds of activity separate, and pages created in the wrong namespace are subject to immediate renaming.

This separation of different kinds of activity into namespaces supported my analysis of the different kinds of activities that take place in Wikipedia, as well as the roles of bots in such spaces. In the month of May 2015, approximately 1.47 billion characters were added to all pages in the English-language Wikipedia (Figure 18). Of these, 540 million characters (or 37%) were added to encyclopedia articles, 83 million characters (6%) were added to article talk pages, and the remaining 800 million characters (57%) were added to wiki pages that were neither encyclopedia articles or article talk pages. Instead, this content was added across namespaces dedicated to specific kinds of coordination and meta-level activities that go beyond individual articles. These include interpersonal communication and notifications from bots (user talk, 25% of all characters added), pages dedicated to policy, process, and meta-level discussions (Wikipedia and Wikipedia talk, 15% of all characters added), and personal pages where editors describe themselves, write essays, and draft articles (user, 14% of all characters added). Bots are also heavily active in these communication, coordination, and governance spaces: in May 2015, while bot accounts were responsible for less than 1% of the text added to encyclopedia articles, 59% of text added to pages in the Wikipedia namespace and originated from a bot account (Figure 19).

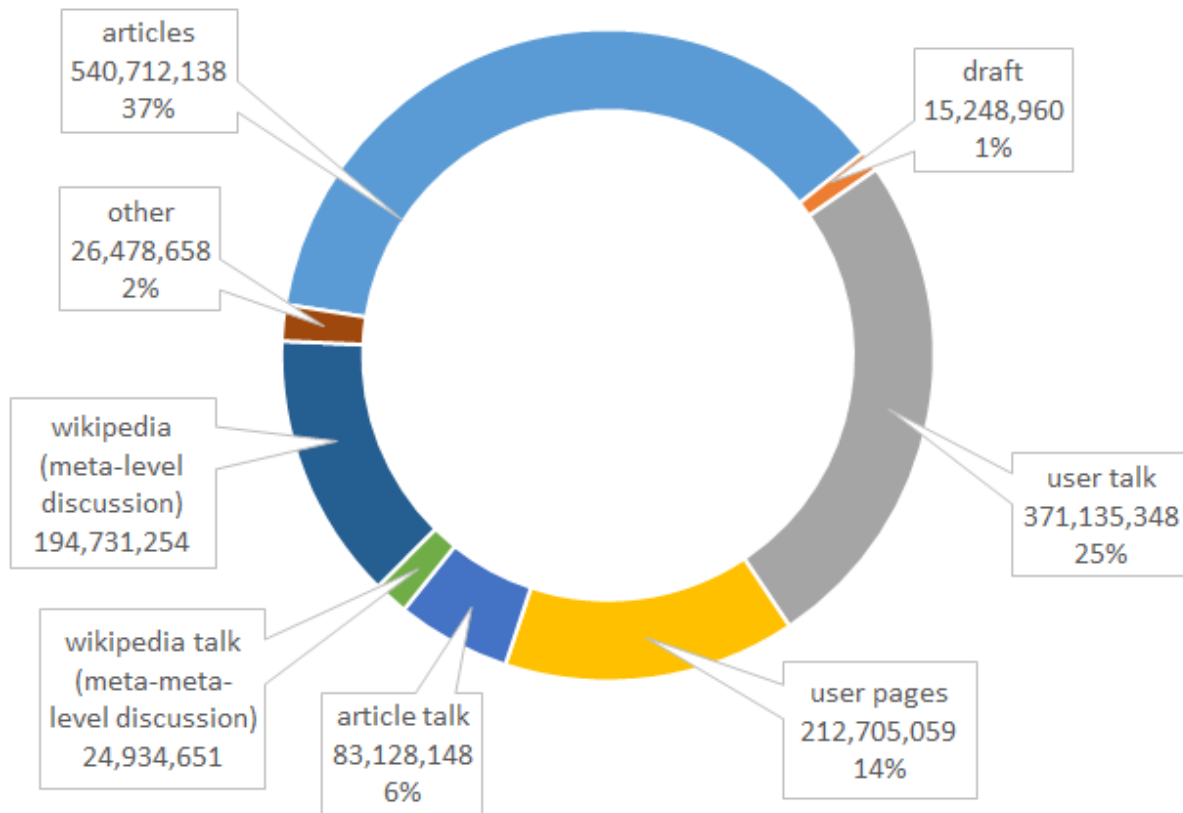


Figure 15: Total number of characters added to English-language Wikipedia pages from May 1st to 31st, 2015. Includes bot and non-bot accounts, broken out by namespace of page.

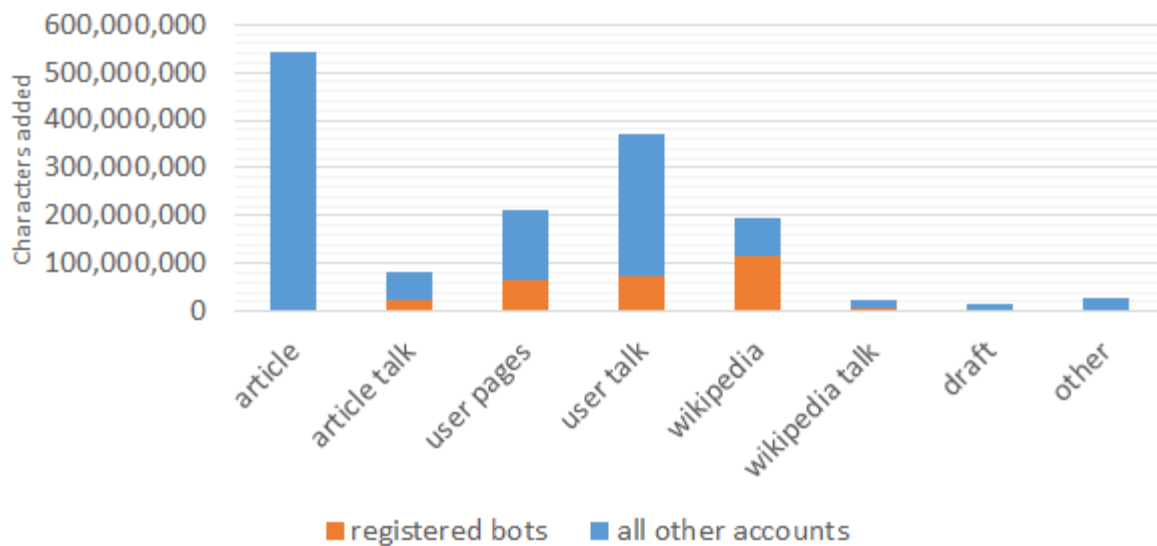


Figure 16: Total number of characters added to en.wikipedia.org from bot and non-bot accounts from May 1-31st, 2015, broken out by namespace.

Second, I analyzed the number of bot edits made to various specialized venues in Wikipedia across the entire history of the project (Table 2). Such a table speaks to the diversity I have personally observed as to the relative role of bots in different specialized venues. Not all administrative processes are equally bespoke. The two pages with the highest number of bot edits – administrator intervention against vandalism and usernames for administrator attention – are fast-paced queues in which bots and humans submit cases for administrator review. A different set of bots then process such requests, keeping the page properly ordered such that administrators can efficiently and easily evaluate such cases. In contrast, the help desk is a different kind of centralized space, where bots do not post responses to requests, but are instead involved in keeping the page ordered and ensuring that questions that have been answered are removed from the main page. Bots in the help desk also link this wiki page to an IRC channel where Wikipedians who are willing to respond to such questions “idle.”

**Table 2: Bot edits to various specialized venues**

<b>Noticeboard or other specialized venue</b>	<b>Number of bot edits</b>
Wikipedia:Administrator intervention against vandalism	431785
Wikipedia:Usernames for administrator attention	115357
Wikipedia:Help desk	32072
Wikipedia:Requests for page protection	27807
Wikipedia:Administrators' noticeboard/Incidents	15866
Wikipedia:Administrators' noticeboard	8146
Wikipedia:Biographies of living persons/Noticeboard	6551
Wikipedia:Media copyright questions	5075
Wikipedia:WikiProject Articles for creation/Help desk	4761
Wikipedia:Village pump (technical)	4246
Wikipedia:Miscellany for deletion	4016
Wikipedia:Dispute resolution noticeboard	3530

Wikipedia:Administrators' noticeboard/Edit warring	3526
Wikipedia:Village pump (policy)	3335
Wikipedia:Village pump (miscellaneous)	2947
Wikipedia:New contributors' help page	2911
Wikipedia:Editor assistance/Requests	2561
Wikipedia:Village pump (proposals)	2547
Wikipedia:Reliable sources/Noticeboard	2461
Wikipedia:Conflict of interest/Noticeboard	2237
Wikipedia:Requests for undeletion	2182
Wikipedia:Copyright problems	2160
Wikipedia:Templates for discussion	2065
Wikipedia talk:WikiProject Spam	1786
Wikipedia:Main Page/Errors	1396
Wikipedia:WikiProject on open proxies	1332
Wikipedia:Fringe theories/Noticeboard	1104
Wikipedia:Arbitration/Requests/Enforcement	1069
Wikipedia:Neutral point of view/Noticeboard	1048
Wikipedia:Bureaucrats' noticeboard	1026

### 3. Discussion: bots as articulation work

#### 3.1 Bots as “clerks” for a process, rather than independent decision-makers

The statistics I presented about the amount of text added to pages or edits to pages indicate the presence of bots in such spaces, but such graphs do not specify the specific roles that bots play in the specialized venues and processes found in pages in the Wikipedia namespace. In fact, the number of characters added could be misleading if it is assumed that bots participate in the same way that humans do – which they do not. These administrative

support bots in Wikipedia are not the kind of “chatbots” with such natural language processing capabilities that they autonomously debate standards of notability or the reliability of sources with human Wikipedians. Such bots are not delegated autonomous, independent decision-making roles about content once reserved exclusively for humans. Bots like AnomieBot are not deciding whether the professor’s edit request is a valid or invalid edit to the encyclopedia – although other bots dedicated to anti-spam and counter-vandalism activity certainly do have such autonomous decision-making ability. Yet the bots that operate in the project’s administrative spaces and venues are just as important when it comes to how Wikipedia operates as a highly-organized site of cultural production. These bots are developed and deployed with the explicit goal of helping the more human contributors to Wikipedia quickly, cleanly, and efficiently participate in these governance-related tasks. In being delegated coordination work supporting a particular administrative process, these “clerks” – as some of their developers call them – make possible particular kinds of formalized institution within the allegedly-anarchistic “anyone can edit” environment of the English-language Wikipedia.

These bots and other bespoke code are the product of a kind of design practice in which the default affordances of wiki pages were dramatically extended to support particular kinds of governance work. These administrative support bots involve a particular kind of redesign, focused on supporting one of many possible understandings of how a certain type of governance work ought to be conducted in Wikipedia. While these bots do increase the usability of the specialized page where such decision-making occurs, these efforts are about far more than simply making wiki pages in general more usable spaces. In many cases, they make it far easier for veteran Wikipedians to engage in the kind of complex, multi-faceted work involved in the governance of Wikipedia. This can make it far more difficult for newcomers to participate – not because bots are inherently difficult to deal with, but rather because bots support more complex kinds of governance practices in Wikipedia. Furthermore, each of these bots are developed with different visions of what this meta-level work in Wikipedia is and ought to be, as well as how this work is to be supported through automation. Some smaller and less controversial processes have remained relatively stable for years and involve only a single bot, like the edit request process supported by AnomieBot, or the Mediation Committee’s process for dispute resolution, also supported by AnomieBot. The edit request process and the Mediation Committee are quite different than the Articles for Deletion process that I examine in chapter 6, for example, which has been supported by 39 different bots that have assisted with various tasks at different times in the 10 years that this process has existed in Wikipedia.

The infrastructures built to support these specialized processes undergird a wide range of different, simultaneously-operating understandings about what Wikipedia – as an encyclopedia, community, organization, bureaucracy, public, institution, project, or any number of other mass nouns – is and ought to be. My analysis extends Tkacz’s argument that studying the project’s various specialized processes and venues “doesn’t allow one to locate in Wikipedia a new organizational archetype; there is no generalizable Wikiocracy. Rather, it is the singularity of different organizational forms that such an approach accentuates” (Tkacz, 2015, p. 135). Those involved in newcomer socialization and mentorship have quite different assumptions, priorities, and goals compared to those involved in fast paced quality control, as can be seen in how these Wikipedians have differently developed and deployed bots in their spaces. In using bots and other bespoke code to extend the functionality of particular wiki pages

inside of the broader MediaWiki platform, these self-selected groups of bot developers and non-developers have made different decisions about not only what kinds of work ought to take place in Wikipedia, but how that work ought to be accomplished. However, there are some commonly-reoccurring elements – most notably the use of templates that can be inserted into articles in ways that are invisible to readers but legible to both bots and humans who know how to follow their traces.

### 3.2: The bureaucracy of crowds

These kind of highly-structured and specialized bot-supported workflows are one of the more invisible yet important aspects to the contemporary operation of one of the world's largest and most visited websites and reference works. Wikipedia researchers have long made such arguments about these processes in the previously-reviewed literature studying individual processes. In fact, one of the first academic papers on such a process is titled “Don't Look Now, But We've Created a Bureaucracy: The Nature and Roles of Policies and Rules in Wikipedia” (Butler et al., 2008), capturing the tension between Wikipedia's popular representation and its increasing formalization. Yet due in part to Wikipedia's “anyone can edit” model and its positioning against the then-dominant *Encyclopedia Britannica*, the project is still depicted in mass media (and some academic circles) as an anarchistic battleground of truth, one that has rejected the formal institutional structures which have long governed the production of knowledge and cultural content.<sup>36</sup>

This allegedly anti-institutional, anti-expert orientation remains a cornerstone of Wikipedia commentary, alternatively critiqued as a “cult of the amateur” (Keen, 2007) or celebrated as “the wisdom of crowds” (Surowiecki, 2004). For those who do not know how to follow the traces used by veteran Wikipedians to coordinate activities behind the scenes, Wikipedia can easily appear as a kind of spontaneously-emergent “self-organizing” endeavour. Such highly economic accounts of the project emerged as Wikipedia first gained widespread popularity and notoriety in 2004-2006, with scholars and commentators making analogies casting Wikipedia as a kind of prediction market, as in Yochai Benkler's “the wealth of networks,” (2007) which celebrates the “information goods ... produced by the coordinate effects of the uncoordinated actions of a wide and diverse range of individuals” (5).

For the overwhelming majority of people who read Wikipedia – and might occasionally contribute by fixing the wording of a sentence or correcting an out of date fact – participation can indeed feel like adding a drop into a vast undifferentiated ocean, which somehow ends up looking like a relatively well-ordered and high-quality reference work. The discourse of efficient labor distribution common in open source software – “with enough eyes, all bugs [or typos] are shallow” – is sometimes deployed in a way that implies there does not need to be any formalization or process at all for coordinating all of those eyes as they scan through all of the changes made to Wikipedia every second of every day. Furthermore, with the project's universal non-profit mission cast as an all-encompassing cosmopolitan vision of “a world in which every single person on the planet is given free access to the sum of all human knowledge,” it is easy for readers, donors, and occasional editors to imagine themselves as

---

<sup>36</sup> For a critical survey and analysis of these discourses, see (Van Dijck and Nieborg 2009; Tkacz 2013).



members of the “Wikipedia community.” This is particularly the case given the dramatic expansion of the term “community” by corporate owners of massive social media platforms to describe their sites and “their users” – as in YouTube’s use of “community standards” to describe their moderation policies, see (Crawford & Gillespie, 2014).

### **3.3: The concept of articulation work**

While researchers have studied how Wikipedia contributors participate in some of these established processes for deciding which articles ought to be deleted or which contributors should be granted administrative privileges, most have focused less on how these bots perform valuable “articulation work” (Strauss, 1985) in establishing these processes as a set of specific tasks oriented towards a particular end. As Strauss defines it, this articulation work is “involved in organizing both the tasks and relationships to them of the people who perform them” (4). ‘Articulation’ refers less to the meaning of the term as a kind of eloquence in speech and more to its meaning as being composed of joints, like bones in a skeleton.

Strauss stresses that tasks are not “automatically articulated” (2), which means that the division of work does not spontaneously emerge, but instead involves specific actors whose work is often made invisible. To play on Strauss’s phrase, when bots use automation to help articulate an administrative process, such “automatic articulation” involves a substantial amount of work, where developers and non-developers design, develop, and deploy a bot to automate the task in specific ways and not others. If this articulation work remains invisible, then it is far easier to embrace the many economic accounts of Wikipedia’s success, which often focus on the “anyone can edit” property of wikis and position the project’s contributors as atomistic agents whose work is abstractly coordinated and aggregated in market-like mechanisms, like those who place bets in a prediction market (Surowiecki, 2004; Tapscott & Williams, 2006). Benkler describes this model as one in which “information goods ... [are] produced by the coordinate effects of the uncoordinated actions of a wide and diverse range of individuals” (5), while Elliott describes the template tagging process in Wikipedia as “stigmergy,” (Elliott, 2006) drawing an analogy with the biological phenomena where ants share information by leaving trails of pheromones for others to follow.

I focus on the invisible articulation work that makes this form of participation in Wikipedia possible, such that even those who are somewhat familiar with how to participate in authoring encyclopedia articles in the project can attribute a naturalistic ‘self-organizing’ quality to the project. It is a similar kind of error that might result if a researcher was able to plot the movements of every person in a city on a blank grid, discovering that people have self-organized into several major arteries – which just happen to line up with major roads and public transit routes. Bot development is a kind of infrastructural work that can easily be made invisible, and if this work remains invisible, then it is easy to assume that there is some natural law or property of emergence which is responsible for the observed regularity in Wikipedia. For example, the process in which vandals and spammers are identified and banned is one I explored in depth in previous work (Geiger & Ribes, 2010), which initially seems like one where there is no explicit coordination. Humans are presented with individual decisions to make about whether specific edits are malicious or not, and once four of a user’s edits are judged to

be malicious, the case (and this evidence) is sent to an administrator to review. The user is banned if the administrator agrees that the user has made four malicious edits.

These tasks are issued and aggregated by automated software agents rather than humans, but that does not somehow mean there is no coordination at work in such a process. First, such automated systems are developed by humans who have specifically decided that they wanted to have a process for banning of vandals and spammers which operated in such a way. Just because automated systems perform this coordination work does not somehow make it any less coordinative. Second, the humans who make such individual decisions cannot be described as taking “uncoordinated actions” any more than the members of any more traditional bureaucracy can. Veteran Wikipedians are generally quite aware of their role within such these processes, particularly with counter-vandalism and anti-spam work. The bespoke tools in which Wikipedians receive such tasks are power tools that are difficult for outsiders to find, and the leading tool requires that the user get approval to use it. Many Wikipedians involved in this process call themselves “vandal fighters” and have a strong collective identity within Wikipedia, often drawing on police imagery. They have their own discussion spaces, chat rooms, an “academy” for training new vandal fighters, and even emblems that some put on their user pages to indicate membership in the “Counter-Vandalism Unit.”

The particular streamlined decision-making processes Wikipedians have developed for and by themselves are not inevitable or universal consequences, but are deeply bound up in particular assumptions about what it means for Wikipedia to be “the free encyclopedia that anybody can edit” (as its slogan declares). As Suchman argues, studying articulation work supported by information technology is part of a broader project in understanding the situated nature of IT in organizations, arguing that scholars:

need to reconceptualize systems development from the creation of discrete, intrinsically meaningful objects, to the ongoing production of new forms of working practice ... the production of new forms of technologically mediated practice relies not only on processional design practice, but on the articulation work that is necessary if an artifact is to be integrated effectively into the situated activities of its use. (Suchman, 1996, pp. 407–8).

I conceptualize these processes as supported by bot-articulated work because Strauss’s concept of articulation work (and its further development by scholars in the field of Computer-Supported Cooperative Work) calls attention to how the administrative processes across the English-language Wikipedia are situated practices. They have developed in certain ways and not others as part of broader conversations among Wikipedians as they worked out issues which were important to them and what role (if any) automation ought to play in the project. This articulation work is one in which highly-active Wikipedia contributors – including bot developers and non-developers – work out new ways to extend the affordances of the wiki medium beyond editing flat text files in the course of working out new understandings of what participation in Wikipedia ought to be. The processes I discuss and reference throughout this section – including three I feature in screenshots on the following pages, Figures 6-11 – illustrate a diversity of approaches around what it means to resolve the tension between the

project's two foundational goals of the wiki- and the -pedia: how ought we create and curate a high-quality encyclopedic text that anyone can edit? (Reagle, 2010)

#### 4. Conclusion

In this chapter, I have given an overview of the administrative processes that exist in the English-language Wikipedia, discussing how much of the “articulation work” (Strauss, 1985) involved in their operation relies on bespoke code, most notably bots. I showed how these administrative processes are supported by bots and other bespoke code, which are not so much delegated the entire administrative task as they are the articulation work supporting the more human Wikipedians who perform various aspects of that task. Such an approach to performing governance work in Wikipedia dramatically departs from longstanding depictions of Wikipedia in popular culture and mass media. Examining various bot-supported venues for administrative decision-making across the English-language Wikipedia gives a quite different view of the project from the more dominant depictions of Wikipedia as an anti-institutional, anti-expert space of knowledge production.

Wikipedians have built highly-automated processes that work to regulate and structure decisions about content and conduct. Such processes are prevalent in the project and they are important to study for several reasons, first of which is the direct impact they have on decisions Wikipedians make about the content of Wikipedia's articles. Yet beyond impact, such processes indicate that highly-active, veteran Wikipedians have built both social and technical structures in ways that generally make sense for them. In fact, these ways may be at odds with the project's public-facing rhetoric about being the encyclopedia that “anyone can edit” – as the opening vignette about my experience with the professor over the edit request process indicates. In the next chapter on membership and socialization, I explore this issue in depth. I show how as of 2015, what it means to be a Wikipedian involves a substantial amount of expertise about how to navigate and participate in these administrative processes. I personally began editing Wikipedia over ten years ago, in 2004, when such formalized processes were the kind of thing that Wikipedians chastised the *Encyclopedia Britannica* for having. What kinds of consequences does such a bot-supported system for decision-making have for newcomers in Wikipedia? And what are the veterans Wikipedians who care strongly about in mentoring and newcomer support– who repeatedly tell me that they would likely be banned if they joined Wikipedia today and edited as they did when they joined years ago – doing in response to such a situation?

This issue about newcomers raises a second related, but distinct issue about bot-articulated processes. As these specialized venues are supported by bots and other bespoke code, such software is developed and designed with particular visions of how a given process should be computationally-supported. As I examine with cases in the following two chapters, studying the active, ongoing development of such software makes visible and explicit particular normative assumptions about what participation in Wikipedia is and ought to be. Each of the specialized venues for decision-making in Wikipedia have been differently designed with various goals and priorities: processes for fast-paced quality control are supported by bots in a different way than processes for interpersonal dispute resolution, and both are supported by bots in a different way than processes for newcomer socialization and mentorship are.

It is easy to take the current existence of any bureaucracy for granted, assuming that the way it operates is simply the way it always has and always will. Furthermore, it is just as easy to tightly couple a technological infrastructure with the organizational process and/or normative assumptions that are supported or articulated by that infrastructure – for example, black-boxing all the elements that make up the edit request process into a single monolithic entity. Yet as I show with several cases in both the following chapter on socialization and the subsequent chapter on the history of the Articles for Deletion process, an algorithms-in-the-making approach shows the complex interplay at work between what are sometimes too easily separated as ‘technical’ or ‘algorithmic’ on one side and ‘social’, ‘organizational’, or even ‘ideological’ on the other side.

## Chapter 5: Socialization and Membership in Administrative processes

### 1. Introduction

#### 1.1 Chapter overview

In this 5th chapter of the dissertation, following the chapter on administrative processes as articulation work, I discuss the relationship between these increasingly formalized bot-supported processes and issues of participation, socialization, and membership in Wikipedia. I discuss these issues in the context of literature on IT in organizations, arguing that the bots supporting Wikipedia's administrative processes are deeply woven into the "fabric" of the project's organizational culture (Orlikowski & Scott, 2008; Zammuto et al., 2007). In this chapter, I focus on issues of organizational culture like socialization and membership to build on the previous chapter's specification of articulation work, which was a concept introduced to help specify how bots are projects of collective sensemaking by Wikipedians. For veterans, the automation of articulation work in specialized administrative spaces is incredibly helpful in supporting their average, everyday work – bots aggregate commonly-occurring requests into centralized spaces. At a higher level, this automation is also a way that Wikipedians decide how they want to make decisions about issues like article deletions – as shown in the history of the articles for deletion process.

I begin with a vignette that illustrates my own status as a veteran Wikipedian: upon creating a new encyclopedia article, I was immediately thrust into a set of bot-supported administrative processes, in which the article was initially deleted by an administrator within minutes after I created it. Due to my existing understanding of the way Wikipedia's deletion processes operate and my existing relationships with other Wikipedians, I was successfully able to get the article undeleted so that I could continue the more routine work involved in writing and expanding the article.

However, as I argue in the rest of this chapter, such a process is often opaque and confusing to newcomers, and runs counter to the early accounts of socialization in Wikipedia that celebrated a successful model of "legitimate peripheral participation" (Lave & Wenger, 1991). This model is one in which newcomers gradually take on tasks that are increasingly complex and high-stakes, as newcomers ostensibly do not need to know how to participate in high-level processes to begin contributing to an encyclopedia article. Yet because these administrative processes are increasingly a core way in which regularly-occurring tasks are coordinated in Wikipedia, newcomers are now often required to participate in these administrative processes as soon as they make a contribution. Today, becoming a Wikipedian involves not just learning the project's discourses and norms of participation, the wiki's complicated user interface, or specialized reference and writing skills, which many Wikipedia researchers have analysed (e.g. Bruckman, Bryant, & Forte, 2005). In addition, newcomers to Wikipedia must become familiar with the highly-automated processes in which decisions are made about a wide variety of work across the encyclopedia project. I end with a discussion of a bespoke specialized venue created in response to this issue, where Wikipedians have used the same kind of bespoke code that has structured administrative processes to create safe mentoring spaces for newcomers to get help.

## 1.2: Introductory vignette: “Save saveMLAK!”

On the morning of August 9<sup>th</sup> 2013, I was sitting in the auditorium of Hong Kong Polytechnic University, where the opening ceremonies of Wikimania 2013 were being held. Wikimania is not an academic conference but an annual hybrid convention/conference for those active in Wikimedia Foundation projects (including but not limited to Wikipedia). The keynote speaker was Makoto Okamoto, the founder of a wiki called saveMLAK, dedicated to coordinate responses and efforts to the 2011 Tohoku earthquake and tsunami. With my laptop out, I looked for an article on saveMLAK on the English-language Wikipedia, but there was none. So I did what Wikipedians do and created one – starting off with a very barebones “stub” and hoping that others would help me expand it during the keynote.

At 9:43am local time, I created the saveMLAK article, which was expanded by another Wikipedian (also in attendance) three minutes later. I got to work on adding an “infobox,” but when I went to add this at 9:47am, I found that someone had used a semi-automated tool named Twinkle to put a giant red notice on the article. This template, `{{db-web}}`, is part of the “criteria for speedy deletion” (CSD) process, one of three different processes Wikipedians have for deleting articles from Wikipedia. According to long-established policy and process, any editor can “CSD” or “speedy” an article they believe fits one of several dozen criteria by tagging it with certain templates. These templates generate all the text Wikipedians used to manually write when arguing for deleting an article according to the project’s notability policies, and the templates make such nominations visible to a large set of human and algorithmic users who know how to follow this trace. The `{{db-web}}` template (rendered as Figure 18) was left on the SaveMLAK article, contains text arguing that the article about a website fails the A7 criteria in the CSD process, which demands that articles “credibly indicate the importance or significance of the subject.”<sup>37</sup> As part of the CSD process, those who tag articles for speedy deletion are supposed to notify the original author of this so they can properly respond, which is a process that has also been automated in the workflows of Wikipedians who engage in this kind of quality control work. Accordingly, a few seconds after I saw the tag on the SaveMLAK article, I received a new message on my talk page, pre-written but signed by the Wikipedian who tagged my article for speedy deletion (Figure 19).

Once an article is tagged in this way, it will then be automatically aggregated to a few different centralized spaces where administrators can review articles that have been recently CSDed. Administrators have the technical privilege in the software to unilaterally delete (or undelete) any page, but the CSD process follows a “four eye principle” (like many in Wikipedia). Administrators are only authorized to delete articles if someone else has first independently evaluated it and deemed it worthy of speedy deletion, indicated by tagging it with a CSD template. However, if two Wikipedians believe that the article should not be speedy deleted, then that is considered sufficient cause to take it out of the CSD process, possibly putting it instead in the more rigorous Articles for Deletion process.

---

<sup>37</sup> A majority of speedy deleted articles are tagged with templates containing A7 rationales. (Geiger and Ford 2012)

*This article may meet Wikipedia's [criteria for speedy deletion](#) as an article about a website, blog, web forum, webcomic, podcast, browser game, or similar web content that does not [credibly indicate](#) the importance or significance of the subject. See [CSD A7](#).*

If this article does not meet the criteria for speedy deletion, or you intend to fix it, please remove this notice, but **do not remove this notice from pages that you have created yourself**. If you created this page, and you disagree with the given reason for deletion, you can click the button below and leave a message, explaining why you believe this article should not be deleted. You can also [visit the talk page](#) to check if you have received a response to your message.

**Contest this speedy deletion**

Note that once tagged with this notice, this article may be deleted at any time if it unquestionably meets the speedy deletion criteria, or if an explanation posted to the talk page is found to be insufficient.

*Nominator: Please consider placing the [template](#):*

```
{{subst:db-notability-notice/SaveMLAK/header=1}} ~~~~
```

*on the talk page of the author.*

**Note to administrators:** this article has content on its [talk page](#) which should be checked before deletion.

This page was [last edited](#) by [Hell in a Bucket](#) ([contribs](#) | [logs](#)) at 01:47 UTC ([21 months ago](#))


SaveMLAK is a [wiki](#) designed to coordinate responses to the [2011 Tōhoku earthquake and tsunami](#), focusing on protecting [museums](#).

**SaveMLAK**

Available in Japanese, English

Figure 17: The speedy deletion / CSD notice on the SaveMLAK article, 9:47am (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

### Speedy deletion nomination of SaveMLAK

 A tag has been placed on [SaveMLAK](#) requesting that it be speedily deleted from Wikipedia. This has been done under [section A7 of the criteria for speedy deletion](#), because the article appears to be about web content, but it does not indicate how or why the subject is important or significant: that is, why an article about that subject should be included in an encyclopedia. Under the [criteria for speedy deletion](#), such articles may be deleted at any time. Please read more about [what is generally accepted as notable](#).

If this is the first article that you have created, you may want to read [the guide to writing your first article](#). You may want to consider using the [Article Wizard](#) to help you create articles.

If you think this page should not be deleted for this reason, you may **contest the nomination** by [visiting the page](#) and clicking the button labelled "Click here to contest this speedy deletion". This will give you the opportunity to explain why you believe the page should not be deleted. However, be aware that once a page is tagged for speedy deletion, it may be removed without delay. Please do not remove the speedy deletion tag from the page yourself, but do not hesitate to add information in line with [Wikipedia's policies and guidelines](#). If the page is deleted, and you wish to retrieve the deleted material for future reference or improvement, you can place a request [here](#). [Hell In A Bucket](#) ([talk](#)) 01:47, 9 August 2013 (UTC)

Figure 18: The speedy deletion notice I received for the SaveMLAK article (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

The way this ideal is implemented in practice is that anyone except for the article's original creator is allowed to remove the CSD template tag without discussion or justification, and administrators are not supposed to delete pages that have had the tag removed by someone who is not the article's original creator. The article's original creator is presumed to always agree that the page should not be deleted, so they are not allowed remove the tag; doing so can result in admonishment and a temporary block if repeated.

When I saw the CSD A7 notice appear on the page, my heart sunk. I didn't even have to read it, as I knew exactly what it said. I also knew exactly what I had to do – and that I might only have seconds to do it. As the article's original creator, I couldn't legitimately remove the deletion nomination tag, but I could add another template-based tag – `{{hangon}}` – that would signal two things to any administrator going through the speedy deletion process. The first signal was more explicit, telling them that I was actively working on expanding the page: after adding this template, a pre-written message saying as much would appear at the top of the article and would be visible on their screens if they were using most of the popular in-browser and standalone bespoke tools that Wikipedians have developed to automate various parts of this process. Yet I had a second, the more subtle motivation, hoping that in properly demonstrating correct usage of such a template within the established workflow of this process, I would be made legible as a Wikipedian who knew the CSD process and should be given some more leeway – unlike most of the people who were creating articles that they were deleting.

As I added the `{{hangon}}` tag in the proper place and clicked the “submit” button at 9:48am, an error message appeared in my browser: the article I was editing no longer existed, as it had been deleted. “Of course,” I thought. Below this error message, I saw the “Start the SaveMLAK article” link that would let me recreate the article if I so desired, but I knew that would be the last thing I should do at this moment. I needed to get an administrator to undelete the article, or else the re-created article would be CSDed again and possibly “salted” – when the deleted page is protected from editing so no non-administrators can create a new version. Normally, what a non-administrator like me would do is go through the Deletion Review (DRV) process, where I'd write up my case, submit it, have it enter a queue, wait for an administrator to process it, have some back-and-forth with them, and so on. But I was in a thousand-person auditorium filled with Wikipedia's upper echelon, all listening to a captivating presentation about this article that just got deleted. Hundreds of people in the room had the technical privilege to undelete the article using their administrative accounts, and any one of them would be authorized to unilaterally do so without needing to give even a justification, given the procedures specified in the CSD policy.

Had the article gone through the more rigorous Articles for Deletion process, an administrator could only undelete it after it went through Deletion Review, but any admin can reverse a CSD. So I posted about the deletion to Twitter with the #Wikimania hashtag attendees had been using, and a friend of mine who was an administrator saw the tweet and promptly undeleted the article in accordance with the CSD process at 9:51am, three minutes after it was deleted. Another Wikimania attendee notified the admin who deleted the article about the undeletion on their user talk page (not required, but done as a matter of courtesy), and a polite discussion took place there. The deleting admin stated they were authorized under policy and process to have deleted the saveMLAK article when they did, but agreed that it might be better



to wait more than a few minutes before deleting new articles. I replied, telling the deleting admin that they could nominate it at AfD if they felt it was still underdeveloped in a few hours. Soon after, I received a “barnstar” for these efforts from a Wikimania attendee (Figure 20) – a token of appreciation Wikipedians send each other for doing good work (see Kriplean, Beschastnikh, and McDonald 2008), which recognize kinds of behaviors that the sender believes are in line with Wikipedia’s broader mission.



Figure 19: The barnstar I received for work on the SaveMLAK article (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

## 2: Specialization and socialization

### 2.1 Articulating trials of strength

I introduce this vignette to illustrate one way in which administrative processes operate alongside average, everyday modes of participation in Wikipedia, such as creating a new encyclopedia article. It may seem quite reasonable that a newcomer who seeks to perform a complex, high-level task – like editing a highly controversial article or changing one of Wikipedia’s policies – might find themselves in a specialized venue full of veteran Wikipedians who have established particular ways of making decisions (along with non-standard jargon and ubiquitous acronyms). However, as this vignette illustrates, administrative processes can come on the scene almost immediately to dramatically raise the stakes of a task like creating an encyclopedia article. I discuss this in relation to the previous chapter’s discussion of “articulation work” (Strauss, 1985; Suchman, 1996) in relation to Wikipedia’s administrative processes, as this concept is also relevant in this chapter’s focus on socialization and membership.

Strauss stresses that tasks are not “automatically articulated” (2), which means that the division of work does not spontaneously emerge (especially not according to some universal, context-independent principle of organizing). Instead, specific actors work to articulate tasks in specific ways and not others, which makes certain kinds of actions easier and harder – a process that Strauss notes is often made invisible. This invisibility also affects newcomers who are seeking to perform seemingly-common tasks, such as adding content to the “free encyclopedia that anyone can edit” – as Wikipedia’s slogan describes itself. As I discuss throughout in this chapter, my insider status in Wikipedia as a veteran editor has developed as I have learned how to follow the traces left by others and leave my own traces when participating in a process like speedy deletion.

The roles of bots and other bespoke code in this process is incredibly important, particularly given the role it plays in helping veteran Wikipedians quickly and efficiently find new articles to review, then delete them if they are judged to be unencyclopedic. My experience having my article nominated for deletion and then deleted within minutes of creating it is not abnormal, either for newcomers or veterans. In a non-peer reviewed 2013 report I conducted for the Wikimedia Foundation, I found that of articles nominated for speedy deletion, the median time between the creation of the article and its nomination was two minutes; of articles deleted, the median time between creation and deletion was 34 minutes (Geiger, 2013). Wikipedia's quality control processes are particularly designed to be efficient ways to minimize damage, but as my colleagues and I have argued, the bespoke code supporting such processes often casts good-faith mistakes and bad-faith malicious contributions in the same category: damage to encyclopedia articles, which ought to be removed as soon as possible. (Halfaker, Geiger, & Terveen, 2014)

I also draw on the concept of articulation work to inoculate the rest of my analysis in this chapter against an overly deterministic reading casting such bots and other bespoke code as inherently ruthless and inflexible agents that have been unleashed upon Wikipedia. Rather, in fitting with the broader algorithms-in-the-making approach in this dissertation, I emphasize how bot-articulated work in Wikipedia is deeply bound up in how Wikipedians actively produce and maintain particular understandings of encyclopedicness. Bruno Latour famously defined reality as “that which resists” (Latour & Woolgar, 1979), referencing the more and less formalized “trials of strength” in which scientists present and defend claims about what is and is not true. The dozens of processes like speedy deletion, the Arbitration Committee, or Administrator Intervention against Vandalism are similarly trials of strength in which Wikipedians identify what Wikipedia is not, removing such content and contributors from the encyclopedia project. Many Wikipedians specialize in particular trials of strength, but part of becoming a Wikipedian is also understanding the multitude of trials that one can encounter in contributing.

## **2.2 Specialization and socialization**

One core aspect of bureaucratization is an increasing specialization and division of labor (Weber, 1968). Wikipedia researchers interested in issues of membership and socialization have written extensively on the specialization and division of labor that takes place in Wikipedia as newcomers gain experience in the project (Bruckman & Forte, 2008; Burke & Kraut, 2008; Forte et al., 2012; Kriplean et al., 2008; Rossi, Gaio, den Besten, & Dalle, 2010; Welser et al., 2011). In this literature, Lave and Wenger's (1991) framework of “legitimate peripheral participation” is a dominant approach among researchers. This understanding of “becoming Wikipedian,” (Bryant et al., 2005) is often expressed through the framework of “reader to leader” (Preece & Shneiderman, 2009), in which newcomers to the project initially engage in smaller, simpler, and lower-risk modes of participation working alongside those who are engaging in larger, more complex, and higher-risk modes of participation. Part of the socialization process involves specialization in particular areas of expertise, which is made possible through the wide range of roles and tasks available. In addition to specialization in more general article topics (such as articles on science fiction or military history), these studies have documented increasing specialization in a variety of project-wide practices, such as article

writing, copyediting, requests from non-Wikipedians, quality control (or “vandal fighting”), policy formation, policy enforcement, and dispute resolution.

### **2.2.1. The diversity of processes: XfD**

In line with this research, I have found that it is common for veterans to specialize in both a particular kind of practice as well as particular specialized venues relevant to such practices. I have found that many Wikipedians express strong opinions about their favorite (and least favorite) centralized venues and administrative processes, choosing to spend more and less time in certain spaces. Furthermore, even within a specific kind of practice such as dispute resolution, quality control, or requests from non-Wikipedians, there exist different specific administrative processes in which veteran Wikipedians often further specialize. For example, I have personally spent a substantial amount of time in the processes dedicated to reverting and blocking spammers and vandals<sup>38</sup>, as well as the processes for deciding which encyclopedia articles ought to be deleted from Wikipedia – which is how I know the speedy deletion process mentioned in the previous vignette so well.<sup>39</sup> Yet for all my years of experience within the processes for deciding about article deletions, until recently, I had not spent hardly any time in the other processes that make up what Wikipedians call the “XfD” family, of which article deletions are the largest process. As I detail in depth in the next chapter, cases pertaining to the existential fate of all pages across all namespaces were once decided in the same process, but sub-processes were spun off this process to separately deal with different kinds of pages. As of 2015, there are also processes for making decisions about files and images (FfD), categories (CfD), templates (TfD), redirects (RfD), and all other non-article wiki pages like talk pages (MfD).

At a first glance, these processes appear to operate in similar ways, which makes sense given their shared 10 year lineage out of a single process: someone believes a certain action needs to be taken about a page based on existing policy (typically deletion, but also renaming and merging are proposed); they nominate the page under the relevant process; a separate space is created for Wikipedians to discuss the nomination; contextual information and relevant links are generated at the top of this discussion; the discussion is listed in various on-wiki spaces so that relevant parties are aware of it; people discuss the case; an administrator “closes” the case by pronouncing what the “consensus” was; the discussion is archived; and if a decision is to be implemented, it is typically done by the closing administrator. Yet there is a diversity within these processes as well, which speaks both the different ways they are supported using bespoke code and the different kinds of Wikipedians who participate in them. For example, the “Files for deletion” (FfD) process is specialized to the particular kinds of decisions that Wikipedians believe are most relevant to these kinds of cases, such as deciding if a copyrighted image qualifies as “fair use,” which is a different kind of decision to make than whether a topic is notable enough to have an article about it. Such spaces are supported by different bespoke code

---

<sup>38</sup> Reverting malicious contributions and blocking malicious contributors are two linked but distinct processes, involving different skills, roles, discourses, tools, and venues.

<sup>39</sup> Although I prefer the Articles for Deletion process far more than the fast-paced speedy deletion CSD process. I run a monthly podcast/playlist called “KAfD: The Wiki,” where I feature songs from bands whose Wikipedia articles have been recently nominated for deletion in the Articles for Deletion process.

as well, from the bots that notify relevant parties that a case has been opened to the templates that generate different kinds of links upon a new case. For example, the Articles for Deletion nomination templates (Figure 21), contains automatically-generated links to search for sources, which the Files for Deletion (Figure 22) and Redirects for Discussion templates (Figure 23) do not.

In another example, I was once talking to a Wikipedian about the Articles for Deletion process, and he told me that I should spend some time in a process he particularly enjoys that was spun off from AfD to deal with how page redirects should operate. The “Redirects for Discussion” (RfD) process is one that involves a much smaller group of regulars who discuss something that, he joked, nobody else will ever see. Unlike some of the other processes for dealing with articles and files, RfD is named “for Discussion” rather than “for Deletion,” which speaks to the different kinds of decisions they make. After spending some time in that process, I experienced this difference between RfD and AfD, but for some reasons, RfD never ‘clicked’ with me in the way that AfD did. This is not out of the ordinary, as I have found that Wikipedians often speak of the need for newcomers to try out different spaces to find which ones they prefer to participate.

**Heavy Mojo** [\[edit source\]](#)

[Heavy Mojo](#) ([edit](#) | [talk](#) | [history](#) | [links](#) | [watch](#) | [logs](#) | [views](#)) – ([View AfD](#) · [Stats](#))

(*Find sources:* ["Heavy Mojo"](#) – [news](#) · [newspapers](#) · [books](#) · [scholar](#) · [highbeam](#) · [JSTOR](#))

Not an obviously notable hip-hop group. There is an [AllMusic page](#) [↗](#) on one of their albums, a short review but nothing substantial. [Ritchie333](#) ([talk](#)) ([cont](#)) 12:39, 1 July 2015 (UTC)

Figure 20: Nomination for an Articles for Deletion case (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

**File:Al-Shabaab fighters.jpg** [\[edit source\]](#)

[File:Al-Shabaab fighters.jpg](#) ([delete](#) | [talk](#) | [history](#) | [links](#) | [logs](#)) – uploaded by [StanTheMan87](#) ([notify](#) | [contribs](#) | [uploads](#) | [upload log](#)).

Fails [WP:NFCC#8](#): not critically discussed. [Stefan2](#) ([talk](#)) 09:29, 16 July 2015 (UTC)

Figure 21: Nomination for a Files for Deletion case (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

**Neon Light** [\[edit source\]](#)

- [Neon Light](#) → [Neon Light \(Blake Shelton song\)](#) ([links](#) · [history](#) · [stats](#)) [ Closure: (@subpage) ]

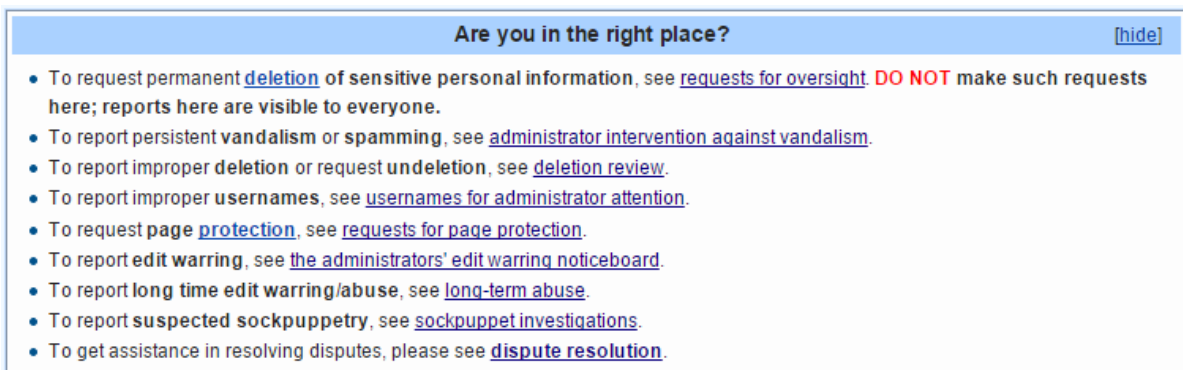
Redirects to a song. In my view, it should redirects to [Neon lighting](#). © [Tbhotch](#)<sup>TM</sup> ([en-2.5](#)). 10:02, 18 July 2015 (UTC)

Figure 22: Nomination for a Redirects for Discussion case (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

### 2.2.2. High-level administrative processes

While I consider Redirects for Discussion one of the tamer and smaller specialized venues in the project, others involve highly consequential project-wide decision-making, making them highly visible and subject to intense controversy. One of the most well-known is the Arbitration Committee, often referred to as Wikipedia’s “Supreme Court” given that it rules on issues where there is a lack of existing process, policy, and precedent to resolve a project-wide issue. “ArbCom” receives much attention and visibility both inside Wikipedia and in mass and social media, largely due to the controversial nature of cases that make it to that final stage of dispute resolution. One particularly polarizing venue in the project is the “Incidents” sub-board of the Administrator’s Noticeboard (referred to as AN/I) which is a designated venue for Wikipedians to bring issues that need urgent attention from the project’s administrators. The approximately 1,600 users with administrative account privileges each have the ability to delete or undelete any page, block or unblock any user account, and protect or unprotect any page from public editing. While there are more formalized processes for requesting these administrative actions, AN/I is a catch-all for both urgent cases and cases that do not cleanly fit into existing processes – including allegations of impropriety by administrators in other processes.

Figure 24, the “Are you in the right place?” banner at the top of AN/I, directs those on the board to other processes and venues. AN/I has a relative lack of procedural formalization



**Are you in the right place?** [\[hide\]](#)

- To request permanent [deletion](#) of sensitive personal information, see [requests for oversight](#). **DO NOT** make such requests here; reports here are visible to everyone.
- To report persistent vandalism or spamming, see [administrator intervention against vandalism](#).
- To report improper deletion or request undeletion, see [deletion review](#).
- To report improper usernames, see [usernames for administrator attention](#).
- To request page [protection](#), see [requests for page protection](#).
- To report edit warring, see [the administrators' edit warring noticeboard](#).
- To report long time edit warring/abuse, see [long-term abuse](#).
- To report suspected sockpuppetry, see [sockpuppet investigations](#).
- To get assistance in resolving disputes, please see [dispute resolution](#).

Figure 23: The "Are you in the right place?" banner at the top of Administrator's Noticeboard/Incidents, directing Wikipedians to more specialized venues (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

compared to other processes due to its status as a space for exceptional issues, with administrators engaging in long discussions in an attempt to build consensus. While AN/I is fascinating site for studying governance in Wikipedia, from my standpoint as a “WikiGnome”<sup>40</sup> who generally contributes in minor ways by cleaning up articles, I consider myself lucky to have never received the dreaded automatically generated template notification on my talk page summoning me to an AN/I thread, although Wikipedians enjoy adjudicating this contentious space (Figure 25).

### Notice of discussion at the Administrators' Noticeboard [\[ edit source | edit \]](#)

---

Hello. This message is being sent to inform you that there is currently a discussion at [Wikipedia:Administrators' noticeboard/Incidents](#) regarding an issue with which you may have been involved.

Figure 24: An automated notification sent to an editor mentioned in an ANI thread (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

### 2.3. Generalization and organizational literacy

The previous section discussed the diversity of processes as a way of speaking to how and why Wikipedians specialize in particular processes in the course of becoming a Wikipedian. However, while this specialization is an important part of becoming a Wikipedian, another vital aspect is generalization across processes. A number of studies of Wikipedian socialization using the legitimate peripheral participation framework have found that successful socialization involves gaining familiarity with a diverse set of activities and spaces in the project (Antin, Cheshire, & Nov, 2012; Antin & Cheshire, 2010; Bryant et al., 2005). I have seen a corresponding phenomenon with administrative processes. This form of socialization involves gaining a baseline level of organizational literacy around these processes, such that those who focus on any given kind of activity – such as proofreading encyclopedia articles or resolving interpersonal disputes, for example – can participate in any given process that may be tangentially relevant in the course of that activity.

As a Wikipedian, my own experience with the project’s processes varies wildly. First, I know some processes by heart and participate in them on a regular basis for their own sake, such that I can often correctly predict what the outcome of a case will be (or have strong feelings when my predictions are not correct). Second, I participate in other processes more peripherally and occasionally, only as they become relevant to other tasks. With those processes, I may be able to know what kinds of decisions are made there, but have less experience in the criteria used to evaluate, for example, whether a username is deemed appropriate or offensive. Third, there are some processes in which I have never personally participated, but have ‘lurked’ due to the important and far-reaching decisions made there. Finally, there are some processes of

---

<sup>40</sup> Part of the popular “WikiFauna” taxonomy of Wikipedia contributors, which Wikipedians humorously use to self-identify different styles and affects of participation. Other fauna include WikiDragons (who make large and ‘bold’ edits), WikiFaeries (who work on images and layout), WikiCats (who are easily distracted and move from article to article), and WikiSharks (who seek out spam, vandalism, and errors to remove).

which I remain relatively ignorant. Yet even for the processes in which I have never even been a ‘lurker’, I understand the general “delegation regime” (Ribes et al., 2013) in which decision-making takes place in Wikipedia. I understand the specific kinds of pages supported by bespoke code I expect to see when I go to one of these processes, I understand the broader norms and assumptions about wiki-based encyclopedia-building that Wikipedians more generally hold, and I understand how these broader norms are generally put into practice in specific processes.

Such generalization is supported and sustained in a variety of ways, as Wikipedians have to not only learn about these issues when becoming a Wikipedian, but also keep up to date with ongoing developments. Like many Wikipedians, one of the major ways in which I keep up with ongoing debates is in reading the weekly *Wikipedia Signpost*, a newsletter written by Wikipedians that generally reports on a wide variety of issues. A typical issue (Figure 26) will include summaries of important decisions made through particular processes, such as: the election of new trustees, the deprecation of the Persondata template, and the promotion of an article about an Italian opera to Featured Article status. *Signpost* issues also keep Wikipedians up to date with broader discussions about processes themselves, like the report summarizing a discussion about whether Requests for Adminship was “a broken process.”

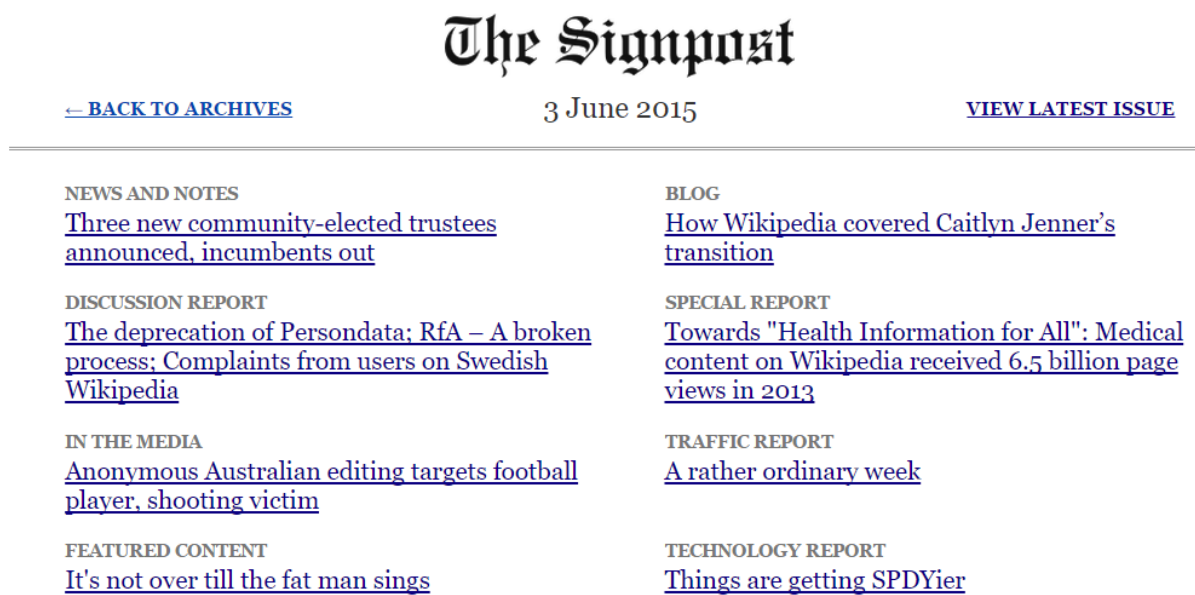


Figure 25: Front page of The Signpost, 3 June 2015 (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

An issue will also occasionally report on particularly contentious cases involving articles, which often span multiple issues and processes, like the report on how Wikipedia covered Caitlyn Jenner’s transition. That *Signpost* article details how a wide set of issues around the article played out, such as the determination that the famous photo of Jenner on the cover of *Vanity Fair* qualified under fair use. There were also two title move requests for the Caitlyn Jenner article: one approved request to rename the article from Bruce Jenner to Caitlyn Jenner,

and another rejected request to name it back. The article discussed how both move requests were processed after a “short, uncontentious discussion,” which was uncontroversial because Wikipedians had previously extensively debated and established guidelines for covering transgender transitions after Chelsea Manning’s transition in August 2013 – a case that ultimately was referred to the Arbitration Committee. The *Signpost* article noted that a new guideline on “identity” about transgender individuals had been developed after the Chelsea Manning ArbCom case was closed in October 2013, reporting that this “This new guideline was applied to Caitlyn Jenner's article and, in part, is the reason that her Wikipedia article has seen a comparative lack of controversy.”<sup>41</sup>

In these ways, Wikipedians are able to keep up with developments at the scale in which Wikipedia operates, maintaining a baseline level of organizational literacy. For example, when the professor asked me about editing his own article in the opening vignette of the previous chapter, I had not dealt with the conflict of interest process in a while, so I had to look up how it was supposed to work. It had been at least a year since I had personally dealt with conflict of interest issues, and I had remembered reading on the *Signpost* that there had been some major changes to the conflict of interest policies on issues around paid editing. Yet because of my general experience with process in Wikipedia, I knew a few things: I knew that there was some kind of centralized edit request process, even though I didn’t know where it was centralized or how to participate in it. I knew that the process would probably look similar to others I had more experience participating in across Wikipedia. Finally, I knew where I should look in order to find more information about participating in the process, or at least people I could ask if I couldn’t find it after a bit of searching. After I spent about three minutes searching and found more information about it (specifically, the page on the `{{request edit}}` template), I immediately recognized the process as having a kind of family resemblance to the other processes which I know far better. In my conversations and interactions with Wikipedians, this kind of partial understanding of the project’s processes is common among even the most highly active editors and administrators.

### **3: Discussion: The encyclopedia that *anyone* can edit?**

#### **3.1 Overview**

As Wikipedia’s slogan is “the free encyclopedia that anyone can edit,” there has been a longstanding concern over the extent to which anyone can ‘actually’ edit Wikipedia in such a way that their contributions have any lasting impact. Issues around participation and socialization in the project has long been a topic of ongoing conversations among Wikipedians and Wikipedia researchers, which intersect with the “systemic bias”<sup>42</sup> that results from the highly unrepresentative demographic composition of Wikipedia’s largely young, male, and Western editors. While I describe these highly-automated administrative processes as core modes of participation in the project and important aspects of membership in the Wikipedian community, this is not to say that such processes are universally accepted by Wikipedians.

---

<sup>41</sup> [https://en.wikipedia.org/wiki/Wikipedia:Wikipedia\\_Signpost/2015-06-03/Blog](https://en.wikipedia.org/wiki/Wikipedia:Wikipedia_Signpost/2015-06-03/Blog)

<sup>42</sup> There has been a “WikiProject” on Countering Systemic Bias (<http://enwp.org/WP:CSB>) since 2004, where a select group of Wikipedians have long called attention to and discussed responses to these issues.



There are many Wikipedians who defend the necessity and configuration of particular processes, as well as many Wikipedians who have long been opposed to the increasing bureaucratization of the project.<sup>43</sup> Those who speak out against bureaucratization often argue that it makes participation incredibly difficult for newcomers (in addition to simply being very “un-wiki”). In talking about the difficulties newcomers face with a number of long-time veteran Wikipedians, I have heard a similar belief repeatedly expressed: that had they entered the project as it exists today and edited as they did when they joined in 2001 to 2006, they would have their edits reverted, articles deleted, and possibly even banned from editing.

### **3.2 Wikipedia and its discontents**

A series of largely quantitative studies by university and WMF researchers – including myself – showed that the number of active editors to the English-language Wikipedia peaked in 2007-8 and has steadily declined since then (A. Halfaker et al., 2013; Suh, Convertino, Chi, & Pirolli, 2009; van Liere & Fung, 2011). According to these studies, veterans are leaving the project at the same rates as they always have, as there is an assumed natural attrition and turnover in any group. Yet starting in 2007, the rate at which newcomers were continuing to contribute to Wikipedia after their first edits substantially declined. These quantitative studies indicate that newcomers are registering and making initial edits at the same rates, but they are not staying around as long as they used to, and not nearly enough to replace the veterans who were leaving. This research on newcomer retention has taken place alongside a simultaneous set of research on the substantial gaps in participation among Wikipedia’s core contributor base, which is disproportionately made up of white men with college educations from the US and UK (Glorr, Schmidt, & Ghosh, 2010; van Liere & Fung, 2011). Subsequent research found that these issues with newcomers were also highly gendered and racialized, with women and non-Western editors experiencing substantially higher rates of attrition (Lam et al. 2011; Antin et al. 2011; Collier and Bear 2012).

### **3.3 Organizational literacy**

Wikipedia’s administrative processes are one of the many different, overlapping factors at play in these issues of newcomer socialization and demographic gaps. Like the procedures found in any organization or institution, these modes of participation certainly do not make sense to many newcomers in the project, and can even be invisible to those who do not know how to look for them – as Heather Ford and I (Ford & Geiger, 2012) have found in our studies of the “organizational literacy” (Darville, 1989, 2009) required to participate in these highly-structured processes. This study was based around cases involving newcomers<sup>44</sup> to Wikipedia, ranging from tenured university professors in the United States to students in Kenya, who all encountered trouble when contributing to encyclopedia articles. These people authored and edited articles as they saw fit in line with the project’s “anyone can edit” motto, seeking to add

---

<sup>43</sup> Such beliefs are also not mutually exclusive

<sup>44</sup> Referring to these individuals as “newcomers” rather than “outsiders” or “non-Wikipedians” is more of a political choice that has significance for broader debates among Wikipedians about membership and socialization. My personal stance is that such individuals should always be seen as potential members of the community who are to be gently and productively socialized into the project, rather than individuals who do not belong.

their knowledge to this collective whole, but found their contributions reversed and articles deleted. These newcomers expressed desires to defend their contributions, but were generally not able to articulate their views in ways that were procedurally legible and intelligible to Wikipedians. In fact, one of the interviewees who encountered the same speedy deletion process I detailed in the introductory vignette (but failed to properly express his objections within this administrative process) expressed uncertainty as to whether the article was deleted by a human or a bot.

As two veteran Wikipedians, Ford and I were able to locate many moments where these newcomers could have been more successful had they been able to author and decode the traces that constitute participation in these highly automated administrative processes. Such organizational literacy is not the kind of generalizable, easily transferrable skill that is often imagined in discussions on Internet literacy. Similarly, knowing how to interact in an academic peer review process to get a journal article published involves a roughly equivalent level of organizational literacy, but it is of a dramatically different configuration than knowing how to interact in Wikipedia to prevent an encyclopedia article from being deleted. Those who enter the project and engage as more peripheral participants must persevere as their contributions are removed from articles until they learn how to “become Wikipedia literate” (Ford and Geiger, 2012: 1), which many do not. Furthermore, in a different study looking at the first message sent to newcomers to Wikipedia, my collaborators and I found that over 75% of these messages were pre-written templates, overwhelmingly sent via an automated software agent or semi-automated bespoke tool (Geiger et al., 2012).

#### **4. The (Tea)house that bots built**

##### **4.1. The Teahouse**

Due to these many of these issues, the Wikimedia Foundation launched a series of funded initiatives in 2012-2014 to support Wikipedians who had concrete proposals to improve the retention of new editors, particularly those from underrepresented groups. These initiatives included rewriting the tutorial and help pages, reforming the dispute resolution process, redesigns of the user interface, classroom education programs, community support for ‘editathons’, outreach to cultural institutions like libraries and museums, and an on-wiki program called “the Teahouse.”

The main purpose of the Teahouse was to create a single, friendly, safe space for new editors to learn how to be Wikipedians, learning the many norms and practices which are necessary to fully participate in the project. The Teahouse’s designers sought to make the space as user-friendly as possible, which was a challenge considering how difficult it is to use the MediaWiki platform upon which Wikipedia runs. By default, all interaction in Wikipedia takes place via the same model Ward Cunningham devised for WikiWikiWeb in 1995: almost every activity on the site is based around editing flat text files<sup>45</sup>, written in a markup language that is

---

<sup>45</sup> The actions not based around flat text files are largely limited to administrative actions. For example, blocking/unblocking a user is performed in a special page that does not look or operate like a wiki, instead using a simple HTML form. However, these pages are so rare that they are literally called “special pages.” Special

functionally equivalent to HTML, but different enough to require additional training. In fact, any of Wikipedia's bots that support administrative processes arose from efforts to use automation to extend this page-based system. For example, as discussed in the chapter on HagermanBot, to reply to a message in a threaded discussion, users must edit the entire discussion page, which is a wiki page like any article. This means that anyone can technically remove or modify anyone else's comments, maliciously or accidentally. Once users are editing the discussion page, they have to then find the message you want to reply to, create a new line, add a colon for each level of indentation, type the message, and then add four tildes (~~~~) to append a signature consisting of your username and timestamp. Even sending messages to other editors takes place in this manner, as there is no private messaging functionality built into MediaWiki.<sup>46</sup> Instead, each user has a dedicated wiki page which anyone can edit that is designated as their "user talk page" – and users will be notified with a banner if someone else edit's their user talk page.

In interviews with those leading the Teahouse project, they expressed their desire to create a more user friendly space where editors could come and get support for learning about how to participate in Wikipedia. They noted that the way in which editors have to generally interact with each other in Wikipedia is dramatically different than most contemporary websites, particularly social media sites. However, they also noted that there was a strong resistance from many of the most active and veteran Wikipedians in making Wikipedia "social" (that is, into a Facebook-like site) – something I have independently seen in a variety of on-wiki discussions and debates. One common sentiment in favor of keeping the outdated interface and interaction experiences is that the barrier to entry is necessary to keep non-Wikipedians out of the project. Those who perform gatekeeping and quality control work have expressed how much work they have to do "cleaning up" after such newcomers, who do not know the project's norms and processes.

The Teahouse team disagreed with such beliefs, and they are part of a broader group of Wikipedians who are in favor of incorporating more WYSIWYG ("What You See Is What You Get") interfaces. The Teahouse is based on the idea that while clicking an "edit" button and modifying a document may make sense for editing encyclopedia articles, it certainly does not for interpersonal interaction – at least when getting newcomers to ask questions. In particular, the Teahouse team wanted newcomers with questions to be able to click a single button, type their question in an overlaid popup box, and have it automatically appear in a queue. They also wanted members of the Teahouse to have lightweight profiles, feeding into a gallery of newcomers and veterans in order to make the space feel populated by people and facilitate mentoring relationships. The Teahouse team also wanted profiles so users could have images that would appear next to their comments when they interacted – a state of the art feature in bulletin boards of the early 2000s. Finally, in what may be assumed to be simplest redesign (but

---

pages are supported by separate PHP-based extensions which make possible page titles that always begin with "Special:", such as <https://en.wikipedia.org/wiki/Special:Block> or <https://en.wikipedia.org/wiki/Special:CreateAccount>

<sup>46</sup> Users can specify an e-mail address in their profiles and there is a feature built into MediaWiki where users can send messages to those who have listed an e-mail address.

ultimately was not), they wanted the most recent questions and discussions to appear at the top of the list, rather than the bottom.

Yet as a set of wiki pages within Wikipedia, the Teahouse had to operate within the wiki document model structuring almost all interaction on the site. Some of the design requirements were satisfied by scaffolding a large set of client-side Javascript and CSS code on top of the wiki pages, but others appeared to require more fundamental changes in how the MediaWiki platform itself operated. Even though the project was funded by the Wikimedia Foundation that legally owns and operates the servers hosting the MediaWiki-powered site, extending this server-side code was deemed both too difficult for the team to do alone – both as a matter of software development and in getting such featured implemented server-side. And so, as Wikipedians do, the Teahouse team built a bot that would use some of the wiki pages more as database tables than as documents. For example, a CSS-styled Javascript element could launch a popup window for asking questions, which would actually have the user’s web browser edit a wiki page based on what they typed in the box. To get these questions to be entered into a reverse-chronologically ordered queue, or to notify veterans that a new question had been asked, they needed a bot that was constantly scanning the Teahouse pages and performing algorithmically-defined actions based on these actions. The bot, named HostBot, was also used to identify potential newcomers who might be in need of help, then send them an automated message inviting them to the Teahouse and linking them with a specific mentor. One of the core problems with newcomers is that they do not know how to ask for help, or even that there are spaces in which they can ask for help. Through HostBot, newcomers are directed towards this space and shown how they can get help if they need it.

## **4.2 Discussion**

The Teahouse raises a number of issues about how bespoke code intersects with values in software design and development. The Teahouse itself is a somewhat polarizing space, with vocal supporters and detractors – which is not surprising due to the ongoing debate within the Wikipedia community about whether the editor decline, new editor retention, and issues of the gender and ethnicity gaps among Wikipedians are “really” problems. The Teahouse has also been heavily critiqued for contributing to the “Facebookification” of Wikipedia, opposed by Wikipedians who also argue against user interface improvements implemented in the core MediaWiki software, most notably the visual (or WYSIWYG) editor and native support for threaded discussions. However, such opposition is varied and multi-faceted, often bound up in stated concerns about membership, participation, and community. For example, according to the Teahouse team, there was one group of Wikipedians who believed in the idea of the Teahouse – a safe space for newcomers where they can ask questions without fear of reprisal – but thought that newcomers should learn how to interact with others by directly editing wiki pages. Ultimately the Teahouse continued with its pop-up window, but added a feature to help newcomers learn about one important aspect of contributing to discussions on wiki pages: the submit button on the pop-up window would not become active until the user signed their comment with the four tilde signature that the MediaWiki parser turns into a signature and timestamp.

Further complicating this case is that the long history of Wikipedians using bots and other bespoke code (especially template scripts) to extend the functionality of the wiki is that such efforts have reached a point where they actively inhibit the development of server-side code built directly into the MediaWiki – especially the efforts to create a visual editor for articles and natively implement threaded discussions. With the threaded discussion feature, there have been many projects that have more and less successfully implemented this into MediaWiki, including a 2006 Google Summer of Code fellowship and three separate WMF-funded projects launched in 2008, 2011, and 2014. However, such functionality is still not present in the version of MediaWiki that Wikipedia runs, and there is strong opposition among veteran Wikipedians to activating these features due to their inability to support the administrative processes which are scaffolded on top of ordinary wiki pages.

According to a WMF staff developer working on the most recent project to incorporate threaded discussions server-side (called Flow), a major reason for the failures of previous UI extensions (and what they seek to avoid) is a “lack of flexibility with regards to workflows and collaboration techniques beyond simple discussion” – specifically, the idiosyncratic, bot-and-template based processes used by various specialized venues and spaces across the Wikipedia projects. A few Wikipedians and WMF staff developers have proposed simply starting over from scratch in the course of redesigning the server-side platform to be more in line with contemporary websites – an unacceptable solution for many of the project’s most veteran and active participants, who have threatened to resign if such a situation were to occur.

## 5. Conclusion

Socialization and membership in Wikipedia is deeply bound up in the bots and other bespoke code that operate in the encyclopedia project. These issues show how, like many information technologies in organizations, the bots that support Wikipedia’s administrative processes are best seen not as “automated plumbing” that simply makes Wikipedia more efficient, but something that is deeply woven into the “fabric” of the project’s organizational culture (Orlikowski & Scott, 2008; Zammuto et al., 2007). Focusing on the role of bots in Wikipedia’s organizational culture is another way I further apply my algorithms-in-the-making approach. For veterans, such bespoke code is a core part of what it means to be a Wikipedian, supporting many of the “trials of strength” (Latour & Woolgar, 1979) in which Wikipedians make decisions about what Wikipedia is not. There is both specialization and generalization across these administrative processes, as they are important ways in which Wikipedians come to interact with each other about regularly-occurring issues.

On the one hand, such administrative processes make it increasingly difficult for newcomers to participate in the project – even veterans must keep up to date with developments, using media like the weekly *Wikipedia Signpost* newspaper. However, as the case of the Teahouse shows, the use of bespoke code to create specialized spaces in Wikipedia is not inherently detrimental to newcomers, as such a mode of software development can help make the wiki more user-friendly, rather than less. This tension shows how an algorithms-in-the-making approach to bespoke code is one of the ways in which Wikipedians work out the many different understandings they have about what the project is and ought to be. Those concerned with quality control have developed certain kinds of bespoke code to scaffold and support their

priorities, while those concerned with newcomer socialization have developed others. This mode of software development often is invisible to both newcomers (who do not know of it) and veterans (who background it), but it is central to understanding the ongoing issues about participation and gatekeeping in Wikipedia.

## Chapter 6. An algorithmic history of Articles for Deletion

### 1. Chapter Summary

In the first section of the dissertation (chapters 2 & 3), I expanded the concept of the bot beyond the software agent and the developer who programs it, arguing that bots were projects of collective sensemaking in which Wikipedians express and negotiate high-level issues about what Wikipedia is as an encyclopedia and a community. In the previous chapter, I illustrated various elements of Wikipedia's specialized administrative spaces. These spaces are where many decisions about the encyclopedia are made, which is a core mode of participation for many of the project's most active contributors. I introduced Wikipedia's specialized administrative spaces using the concept of articulation work, which helps specify how bots are able to provoke and resolve high-level debates. Such bots do not autonomously decide which articles ought to be deleted or which sources are considered reliable; they instead "clerk" for specialized venues in which the more Wikipedians make such commonly reoccurring decisions.

In this chapter, I apply this lens of bot-articulated work as collective sensemaking in studying the history of a single specialized administrative process: deciding which articles ought to be deleted from the encyclopedia. I ask question I raised towards the end of the previous chapter: how did these highly structured, centralized processes develop into what they are now, especially given the early rhetorical position of Wikipedia as an anti-institutional mode of cultural production? Did such processes initially come on the scene with the kind of highly automated algorithmic support I detailed in Wikipedia's contemporary operation? This second question is easier to answer than the first, but I discuss both in analyzing the history of deletion processes in Wikipedia. I conducted archival research relying on several sets of sources: Wikipedia's revision histories of "on wiki" discussions, archives of official Wikipedia mailing lists, personal accounts by early Wikipedians (some written at the time and others written in retrospect), and existing academic scholarship on the history of Wikipedia. I focused on the "Articles for Deletion" (or AfD) process, which was initially called "Votes for Deletion" (or VfD) when it was created in 2002 – the year after Wikipedia was founded.

I chose to focus on deletions in Wikipedia for many reasons. First, deletion processes were the earliest and most longstanding specialized administrative process in Wikipedia. The first version emerged in 2002, just a year after Wikipedia was founded. While the deletion processes have changed substantially, there has always been some kind of specialized venue for deciding what the policy ought to be for deleting articles, as well as a centralized space for evaluating particular articles based on the principles elaborated in the deletion policy.

The second reason for focusing on deletions is that this kind of gatekeeping work provokes rich reflections among Wikipedians about what they think their project is and ought to be. This is similar to how Finn Brunton discusses his broader history of spam on the Internet as "the negative shape of the history of people gathering on computer networks ... It is defined in opposition to the equally shifting and vague value of 'community'" (Brunton, 2013, p. xvi). Wikipedians worked out what they were in part through deciding what they were not, something long discussed by Wikipedia researchers (Bruckman & Forte, 2008; Konieczny, 2010; Reagle, 2010; Tkacz, 2015). In fact, as such researchers all note, the policy on "What Wikipedia is Not"

(WP:NOT) has been and remains a core site and document through which Wikipedians come to represent, reflect, and negotiate what they think they are involved in when they call themselves “the free encyclopedia that anyone can edit.” Reagle argues that deciding what Wikipedia is not is a core way in which Wikipedians work out the project’s foundational, productive tension: deciding what it means to be both a “wiki” (where anyone can edit) and a “-pedia” (a high-quality reference work).

In a related but distinct vein, I continually found that deletion processes are important to study because the process has continually been a bellwether for broader issues and trends around governance and decision-making in the project. When a more systemic, project-wide issue arises in Wikipedia, it often is manifested first in issues around deletion – possibly because of the central role that deletions play in defining what Wikipedia is and is not as an encyclopedia. For example, as I show in this chapter, the first of many major ‘constitutional crises’ between Wikipedia’s volunteer contributors and the “benevolent dictators” who owned and operated Wikipedia’s servers (initially Jimmy Wales and Larry Sanger, now the Wikimedia Foundation) was over Sanger’s unilateral deletion of wiki pages. The article deletion process was the project’s first and most longstanding specialized process, created in response to this controversy. It was the first process to take place on the wiki instead of the mailing list, which was the start of a shift in media use for meta-level issues, in addition to the formalization of process.

Next, when Wikipedia experienced sustained exponential growth in terms of the number of editors and edits made in 2004-2005, article deletion processes were some of the first areas to become overwhelmed by this influx of newcomers and contributions. A broad shift by Wikipedians to focus on quality rather than quantity – which Jimmy Wales famously advocated in his keynote speech at the first annual Wikimania conference in 2005, responding to a number of high-profile controversies over inaccuracies in Wikipedia – played out heavily in article deletion processes. Such processes were a key site in which Wikipedia’s first opposing ideological factions of “inclusionists” and “deletionists” formed and engaged in heated debates. As administrators had the capacity in MediaWiki to delete or undelete any pages, issues around deletion played out in the election of administrators, as well as how to deal with “wheel wars” between administrators. Finally, as I discuss in depth, article deletion processes were the first to be specifically supported through bots, which first took place in January 2005, around the time that these issues were starting to emerge.

In one sense, the emergence of bots and other bespoke code was based in a common problem where automation is posed as a solution: efficiency, particularly at scale. The sustained exponential growth in contributions and contributors to Wikipedia during 2004-2006 led to a corresponding increase in the number of deletion decisions to make, which strained the capacity of the wiki-based process Wikipedians used to make such decisions. Even after Votes for Deletion was split into Articles for Deletion and a set of other similar processes for other kinds of deletion-related wiki pages, hundreds of articles were being nominated for deletion every day. Yet I warn against unreflectively using the concept of efficiency to discuss the emergence of bots that helped “clerk” this process, which would be the kind of “automated plumbing” trope that scholars of IT in organizations caution against (Orlikowski & Scott, 2008; Zammuto et al. 2007). Instead, in line with this literature on organizations and sociomateriality, I discuss



both successful and failed attempts to improve the efficiency of the deletion process as they were situated within the broader context of Wikipedia’s organizational culture. I focus on two major internal factions of “inclusionists” and “deletionists,” who held opposing views about Wikipedia’s standards for inclusion, and correspondingly, how easy or difficult it ought to be to decide to delete a Wikipedia article.

While automation of decision-making was generally frowned upon by inclusionists and deletionists alike, the “clerk” bots that were designed, developed, and deployed were part of a broader project-wide shift in the increasing formalization of policy and process. This formalization was itself was part of a shift in priorities as a growing number of Wikipedians sought to focus on improving the quality of existing articles, particularly in response to high-profile controversies over the accuracy of Wikipedia (or lack thereof). Previous Wikipedia researchers have made similar arguments about the history of Wikipedia, but have largely limited their methods and cases to the more immediately ‘human’ aspects of Wikipedia. My contribution with this historical analysis is to make such an argument by showing how an algorithms-in-the-making approach illustrates how code can be a medium in which people work to imagine, articulate, negotiate, enact, and contest ideas about what a site like Wikipedia is and ought to be. My analysis specifies particular aspects of how bots serve as projects of collective sensemaking, focusing on how these issues played out when bots were delegated the articulation work needed to support decisions about deletions.

## **2. A History of Deletion**

### **2.1: The problem of server-sovereigns: Wikipedia’s early years (2001-2002)**

While these specialized process pages are dominant in the English-language Wikipedia in 2015, the project has not always had such formalized procedures for decision-making, as I found in archival research I conducted into Wikipedia’s early and formative first years. Wikipedia was created as a side project to the failing Nupedia, which was a volunteer-edited, freely licensed encyclopedia project run by Wikipedia’s co-founders Jimmy Wales and Larry Sanger. Nupedia was based on a highly formalized, expert-based editorial model, which Wikipedia was explicitly created to eschew – as discourses of anti-institutionalism indeed portray. By the end of Wikipedia’s first six months, a convention had emerged in which pages on the wiki were to be used for encyclopedia articles and article-specific discussions (with talk pages designated like “Creationism/talk”), while the mailing list was to be used for governance and meta-issues. There were certainly many disputes as former Nupedians and newcomers worked out how what it meant to write a wiki-based encyclopedia. Yet for the most part, these were handled by the then small and tight-knit group of contributors discussing content disputes on the wiki or higher-order disputes on the mailing list. However, in November 2001, Wikipedia’s co-founder Larry Sanger became involved in a bitter conflict with a user known as The Cunctator, the two disagreeing on the subject of vandalism.

The Cunctator had created an archive of (in their opinion) humorous and historically significant vandalism on an article titled “Wikipedia vandalism,” which Sanger permanently deleted from Wikipedia using his privileged technical access to the server. The Cunctator then

recreated the vandalism archive on a sub-page<sup>47</sup> of their user page (“User:The Cunctator/Wikipedia vandalism”). At the time, user pages were generally considered to be under the purview of each contributor; the main user page was where contributors would describe themselves, while subpages were used to write personal essays about Wikipedia as well as serve as a draft space for articles that were not yet ready for the main encyclopedia. In deleting The Cunctator’s vandalism archive, a number of contributors alleged that Sanger not only violated the unwritten assumption about the independence of user pages; they also alleged he went far beyond the recently formed practice for removing pages that were merely unsuitable for inclusion in an encyclopedia. At the time, there was a strong consensus on the mailing list that Wales and Sanger had not only the right but the obligation to use their access to the server to permanently delete legally problematic content (like copyright violations or libel) – Wales and Sanger would typically send a notification to the list when this occurred. However, all other pages that were merely unencyclopedic were to be ‘deleted’ by editing the page to make it blank. In this way, the text would not be immediately accessible to readers or indexed in search engines, but the history of a page’s revisions would be intact such that any editor could reverse it (or simply browse the content themselves). Instead of this convention, Sanger used his technical access to the server to permanently delete any record of the page’s existence. At this time, Wikipedia was still running on the older UseModWiki software system, which did not have specialized user accounts for administrators or built-in affordances for deleting articles. Instead, administrative actions were taken by directly modifying the server-side database, and Wales and Sanger gave out the administrative password to a small few who they trusted.

In messages to the wikipedia-l mailing list, a number of Wikipedians accused Sanger of abusing his administrative powers, starting a long threaded discussion.<sup>48</sup> Not only had Sanger gone beyond the emerging conventions around deletion, he also had not discussed this issue with anyone, even The Cunctator, nor did he give notice that he deleted the articles. A flurry of e-mail messages followed the initial accusation, including one by Wales in which he said he was “investigating the page deletion ‘allegations’”<sup>49</sup> as well as implying that he agreed with the deletion. Soon after, Sanger responded in a forceful e-mail claiming that he and Wales could rightly delete any page that worked to undermine their authority. Furthermore, Sanger wrote that the nature of his authority actually demanded that he not justify his decisions to the community, because that to do so would undermine his authority and subsequently hurt the project. Instead, they had to simply trust the benevolent dictators that held the keys to the server. In his reply to the mailing list, he declared:

I do have more authority on this project than do others. So does Jimbo [Wales], of course. On issue of more importance, when a controversial or important decision must be made, my role in this project is to make it (or delegate it) and, if necessary, to defend it (or to justify it in advance).

---

<sup>47</sup> Sub-pages are created by adding a backslash after an article’s name. Originally, sub-pages were just a naming convention, treated as independent and distinct pages by the wiki software, but later versions of MediaWiki recognized subpages as tree structures and incorporated browsing them in the user interface.

<sup>48</sup> <http://www.webcitation.org/5d4psV9Sr>

<sup>49</sup> <http://www.webcitation.org/5aJpVzGzR>

[...] I do reserve the right to permanently delete things—particularly when they have little merit and when they are posted by people whose main motive is evidently to undermine my authority and therefore, as far as I’m concerned, damage the project.

[...] there will be other somewhat similar situations, in which people's pages are deleted and the injured parties will demand justice in a public forum. Then I will, of course, be accused of acting like an autocrat ... these accusations will be raised by teenagers and college students with too much time on their hands, and by intelligent people with mental problems whether moderate or serious.

[...] In such situations, I’m going to have to trust that you will trust that I am acting in the best interests of Wikipedia, and indeed not abusing my authority.<sup>50</sup>

## 2.2: New process, new software (2002-2003)

Chaos ensued on the mailing list as many Wikipedians were furious at Sanger, bringing in many to the thread who initially did not have much sympathy for The Cunctator and their vandalism archive. Jimmy Wales tried to defuse the situation by taking the blame: “It is possible that in testing the page deletion command, I had something to do with all of this. ... Let’s assume that I did it, by sheer accident ... and let’s all move on. I will help to restore the pages.”<sup>51</sup> However, this was unsatisfactory for many Wikipedians, who for some reason did not believe Wales. Eventually, Sanger replied, issuing a vaguely worded “blanket apology” and proposing that all interested parties would “compose a (publicly-editable, of course) page about what basic policies we will follow in deleting pages permanently.”<sup>52</sup> This first became realized in an article titled “Wikipedia policy on permanent deletion of pages”<sup>53</sup> that later was condensed to “Wikipedia:Deletion policy.” This original policy was based on a software-based limitation that deletions in UseModWiki were irreversible (“Hence, the decision to permanently delete an article is not to be taken lightly,” an early version of the policy read). However, this changed a few months later, with the release of a completely rewritten wiki software package by Wikipedian Magnus Manske, which would later be named MediaWiki. (Originally, it was called “the PHP script”).

The new wiki software deployed in January 2002 supported special user accounts for administrators, whose actions were to operate in a similar spirit to the “anyone can edit” model of article writing. Any admin could delete any page (removing it from public view), but any admin could also undelete any page. New administrators could also be given this ability without giving them full access to Wikipedia’s servers. The deletion policy page shifted from a kind of Magna Carta-esque document seeking to constrain the powers of server-sovereigns to one bound up in a corresponding process page called “Votes for Deletion.”<sup>54</sup> VfD, as it was called, was a specialized wiki page in which Wikipedians would manually list potentially unencyclopedic articles in bullet points, then discuss in manually-indented replies below

---

<sup>50</sup> <http://www.webcitation.org/5d4q2Izo3>

<sup>51</sup> <http://www.webcitation.org/5d4q6jWnm>

<sup>52</sup> <http://www.webcitation.org/5d4qPfY3U>

<sup>53</sup> <http://www.webcitation.org/5d4rOqHkx>

<sup>54</sup> [http://web.archive.org/web/20030228094730/http://www.wikipedia.org/wiki/Wikipedia:Votes\\_for\\_deletion](http://web.archive.org/web/20030228094730/http://www.wikipedia.org/wiki/Wikipedia:Votes_for_deletion)

whether each article nominated for deletion fit these new deletion rules. A page was only to be deleted if an uninvolved administrator deemed that there was a consensus in favor of deletion, based on the discussion that took place in that space. As the project's administrative corps grew, the decision to promote users to administrators (once simply the prerogative of Wales and Sanger) was devolved into a process called Requests for Adminship.<sup>55</sup> RfA became another specialized space for the same kind of decision-making process, dedicated to appointing new administrators. Sanger, increasingly unsatisfied with the lack of respect for expert contributors and his own administrative authority, resigned from Wikipedia in late 2002, leaving Wales as the project's sole "benevolent dictator" – a role that was, as Wales occasionally reflected in messages to the mailing list, more about giving "royal assent" to decisions that "the community" had already agreed to.

After the creation of Votes for Deletion and Requests for Adminship, other process pages were created for blocking/unblocking user accounts and protecting/unprotecting pages from public editing<sup>56</sup> – the other tasks that the new class of administrative user accounts could unilaterally perform. With these later versions of the wiki software, any administrator had the ability to delete any page, block any user or IP, and protect any page from public editing, but any other administrator could reverse their action. Just as Wikipedians discouraged each other from entering into "edit wars" over article content, administrators were also to not enter into "wheel wars" over administrative actions – although there are a number of these cases in Wikipedia's early years. To prevent "wheel warring," a convention was developed in these spaces where each case would be "closed" by an administrator who was not involved in the discussion. The closing admin would archive the debate (which took many different forms), announce what the "consensus" was, then take the appropriate action using their administrative account privileges.

By 2003, these process pages became well-developed specialized venues, known among veterans as the default place for these decisions to be made. Links to these venues could be found in various related meta-level spaces directing those with issues to the right venues, and Wikipedians also began leaving notices on the user talk pages of those who they thought should know about a particular case going through a particular process. Wikipedians who were active in particular processes began to develop more and more sophisticated standards they wrote in policy documents, which were typically described as not being prescriptive, but "intended for the most part to be descriptive of existing community norms,"<sup>57</sup> synthesized from the outcomes of previous discussions. At the time, these pages were not extended using the highly-automated assemblage of bespoke code present in Wikipedia process pages today. Such process pages were manually

---

<sup>55</sup>

[http://web.archive.org/web/20030801083124/http://www.wikipedia.org/wiki/Wikipedia:Requests\\_for\\_adminship](http://web.archive.org/web/20030801083124/http://www.wikipedia.org/wiki/Wikipedia:Requests_for_adminship)

<sup>56</sup> [https://en.wikipedia.org/w/index.php?title=Wikipedia:Requests\\_for\\_page\\_protection&oldid=1855623](https://en.wikipedia.org/w/index.php?title=Wikipedia:Requests_for_page_protection&oldid=1855623)

<sup>57</sup> [https://en.wikipedia.org/w/index.php?title=Wikipedia:Policies\\_and\\_guidelines&oldid=575028](https://en.wikipedia.org/w/index.php?title=Wikipedia:Policies_and_guidelines&oldid=575028)

- [Deeper Than Schmitz](#) and all related articles about band members and albums etc... "Deeper Than Schmitz" - produces no hits on google[1]. Even if they exist they are not worthy of an article here. [Mintguy](#)
- [cargo cult business](#), [cargo cult investment](#), [musical chair finance](#) and [founder's fever](#)
  - can find no reference to these terms on Google. [Mintguy](#) 03:06 Feb 17, 2003 (UTC)
  - all four appear to be bogus terms made up by the anonymous contributor [Mintguy](#)
  - I'm not sure that I've heard any of these terms before either (though founder's fever rings a faint bell), but these terms are very apt for the business concepts they describe. If these concepts don't have a name, these are good ones. [Stephen C. Carlson](#)
  - Wikipedia is not a place to coin new terms for existing concepts however apt they may be, if no one actually uses them they are non-encyclopaedic. [Mintguy](#) 01:12 Feb 22, 2003 (UTC)
- [IRC International Water and Sanitation Centre](#)
  - copied from <http://www.irc.nl/aboutirc/> [The Anome](#) 15:20 Feb 17, 2003 (UTC)
- [Johannes Mueller](#)
  - Now an empty page (the redirect was broken) [User:Loxix](#) Mon Feb 17 19:48:26 UTC 2003

Figure 26: Votes for Deletion, Feb 2003 (from archive.org) (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

See [Wikipedia:Protected page](#) for details on the purpose of protecting pages and the guidelines concerning page protection. If you would like to request a page be protected; please list it (and the date) below, with the reason that it needs protecting

Please remove pages once they have been protected; or once the requestee no longer wishes for them to have protected status.

- [2003 U.S.-Canada Blackout](#)
  - Wik continues to violate the naming conventions
- [Alternative Medicine](#)
  - The original version of this page is fundamentally anti-alternative medicine, is biased and is factually incorrect. My version of this page is fundamentally pro-alternative medicine and is factually correct. This encyclopedia exists to convey factual information rather the deep seated hatred of the opponents of alternative medicine.
- [Alan Dershowitz](#)
  - User 195.92.168.172 reverts anything not on his agenda and calls everything else propoganda.

See also: [vandalism in progress](#), [Edit wars in progress](#)

Figure 27: Requests for Page Protection, Dec 2003 (from Wikipedia's revision histories) (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA)

curated with new cases indicated by a top-level indent or bullet point and replies threaded below (as in other wiki talk pages). The closing administrator would generally remove all the replies about a case upon making a decision, and if someone disagreed with the close or wanted to discuss the issue further, they could simply add a new thread to the page and start the discussion again.

### 2.3: The bots enter the process (2004-5)

By the end of 2004, Wikipedia was gaining significant popularity beyond its once tight-knit group of contributors, beginning several years of sustained exponential growth in terms of contributors and articles (Halfaker et al., 2013; Suh et al., 2009). The Votes for Deletion process, dedicated to removing articles that anyone could create, had become one of the most active and fast-paced spaces in Wikipedia – and one of the most controversial, as factions had broken out between “inclusionists” and “deletionists” over what the project’s policies were to be (this broader conflict between these self-identified groups is a core tension in the project and has been extensively analyzed in Reagle 2010; Stvilia et al. 2008; Konieczny 2009; Kennedy 2010; Kostakis 2010). At this time, between 50 and 200 articles were being nominated for deletion every day, and deletion debates were to stay open for seven days before being closed except for extreme cases of patent nonsense, copyright violations, and spam. VfD grew dramatically in late 2004: according to statistics I collected from the Wikimedia Foundation’s analytics servers, these deletion process pages were edited by non-bot accounts approximately 3,000 times in the month of July 2004, but jumped to over 30,000 edits in the month of July 2005 (Figure 29). Originally, every debate about potentially unencyclopedic articles took place on the same wiki page, which was just a flat text file anyone could edit, like all discussion pages in the project. Participating in the process was harrowing, as anyone who wanted to list a new article for deletion or participate in a deletion debate had to edit a wiki page that was multiple megabytes in size – one Wikipedian posted that even on DSL, the page took over 10 seconds just to load. There were also frequent calls to disband VfD by inclusionists, who were not only opposed to deleting most articles, but also referred to the increasingly-formalized process as a very “un-wiki” development. At one point, an inclusionist administrator simply deleted the VfD page, with another administrator almost immediately undeleting it – which reportedly locked

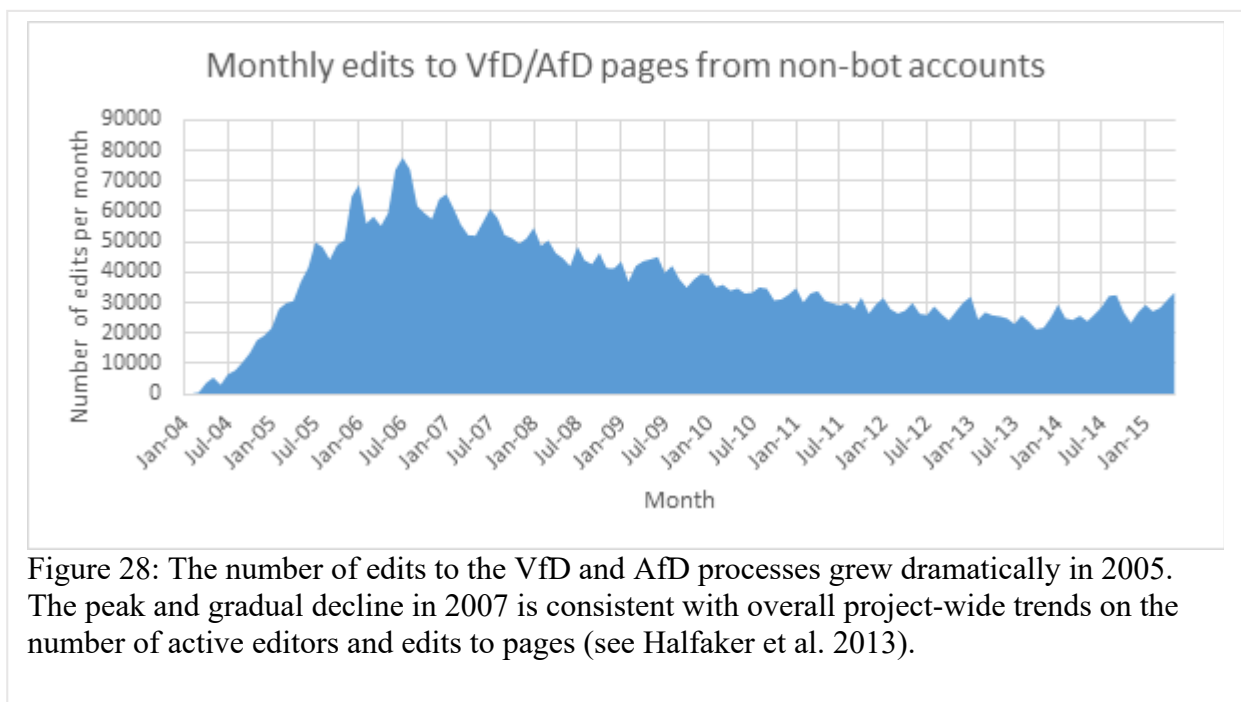


Figure 28: The number of edits to the VfD and AfD processes grew dramatically in 2005. The peak and gradual decline in 2007 is consistent with overall project-wide trends on the number of active editors and edits to pages (see Halfaker et al. 2013).

up all of Wikipedia from editing as the servers struggled with processing these conflicting commands for a page with tens of thousands of revisions.

The first mention of a proposed bot to operate in VfD I have been able to locate was made by an inclusionist in a long, broader debate about the future of deletions and process on Wikipedia. This Wikipedian stated that because they objected to the deletion of any article on principle, they would like to have their “keep” vote automatically added to every VfD debate. This was countered with strong opposition by deletionists, including a proposal to have a bot automatically remove votes made by those who expressed this kind of far-inclusionist position. No consensus was reached about this bot (with many labeling the original proposer a troll), but some participants said it raised an important issue about what a vote in VfD meant for closing admins. Were votes to be counted in a majoritarian fashion like in an election, or were they simply to advise and guide admins in reaching a more informed decision? The “auto-keep bot” discussion died out after a few dozen replies, with the highly-active participants in VfD going back to the longstanding debate over whether VfD should be made into a more efficient space for quick decision-making (as deletionists desired) or whether gaining a consensus to delete an article ought to be something that was difficult. As more deletionist-aligned VfD regulars sketched out possible redesigns for the process, they accused inclusionists who opposed such efforts of being obstructionists, not wanting to see VfD be an efficient space.

In January 2005, those who supported more formalized processes ultimately won out and restructured Votes for Deletion. Each article would be discussed on its own separate wiki page, which would be titled “Wikipedia:Votes for deletion/Article name.” A bot developer active in these meta-level discussions about the future of VfD named AllyUnion used this restructuring as an opportunity to better support the process with automation. At this point, bots had generally only been used to automatically author or edit encyclopedia articles, of which there were dozens by January 2005. With separate pages for each discussion, VfD debates were more easily parsed using automated software agents, and there were many aspects of the VfD workflow that could then be automated. Many VfD regulars celebrated AllyUnion’s work in trying to create usable spaces, discussing how the process could be improved further. As the wiki page supporting VfD turned from a massive threaded discussion board to a directory linking to each individual deletion discussion, AllyUnion’s VFDBot took on more and more coordination work that Wikipedians were having to do manually. For example, VFDBot was delegated the role of helping close VfD debates after the designated 7 day period, which was something originally done manually by an administrator. Once a page for a VfD debate had existed for 7 days, VFDBot would remove it from the list of active discussions and copy it in a temporary archive. Closing admins were to scan through this archive and look for pages that had not been “closed and boxed”: when pronouncing an outcome for the debate, admins were to also archive it by adding special templates at the top and bottom of the debate, which declared the text in between to be an archive and changed the text background color to light blue (Figure 30).

This page is an archive of the proposed deletion of the article below. Further comments should be made on the appropriate discussion page (such as the article's talk page or on a Votes for Undeletion nomination). No further edits should be made to this page.

The result of the debate was **delete**. [Woohookitty](#) 09:27, 27 August 2005 (UTC)

**Michel Delon** [\[edit source\]](#)

**Delete** nn bio ad. Made by a blank IP whose only other edits were of the company this guy owns. [Karmafis](#) 21:11, 20 August 2005 (UTC)

- **Comment** Ad and worthy of deletion in its current form. The only link is from [Detroit Fashion Week](#) so it may be worth rewriting if he is significant enough in fashion. However, a quick Google showed that there is a French professor at the [University of Paris](#) who has published several works on the Enlightenment see [\[28\]](#) [↗](#). Perhaps we should create a disambiguation for this page. C 00:50, 21 August 2005 (UTC)
- Very clear **delete**.—[Encephalon](#) | [↗](#) 01:01:36, 2005-08-21 (UTC)
- **Delete** [Dottore So](#) 05:50, 21 August 2005 (UTC)
- **Delete**. [Alf](#) 07:59, 21 August 2005 (UTC)
- **Disambiguate** per above. Wiping all spam ofcourse. [Zanaq](#) 00:56, 22 August 2005 (UTC)
- **Delete**. Non-notable ad copy. (Oops! Forgot to sign my name yesterday, sorry!) [Friejose](#) 15:18, 26 August 2005 (UTC)

*The above discussion is preserved as an archive of the debate. **Please do not modify it.** Subsequent comments should be made on the appropriate discussion page (such as the article's talk page or in an [undeletion request](#)). No further edits should be made to this page.*

Figure 29: A typical Votes for Deletion discussion in August 2005 (from Wikipedia's revision history). This discussion has been closed and archived by an administrator, with this work assisted by various bots. (material © Wikipedia contributors, freely licensed under Creative Commons BY-SA license)

#### 2.4: The expansion of bot work in deletion-related processes, 2005-2015

As Votes for Deletion continued to develop in both size and specialization, more and more bots were deployed to help assist the increasing formalization at work in the process. Sub-processes for dealing with non-article pages were spun off from VfD, creating the “XfD” family of processes: Categories for Discussion, Files for Deletion, Redirects for Discussion, Templates for Discussion. VfD was renamed “Articles for Deletion” in September 2005 without much change to the process itself. The renaming was explicitly described as a signal that participating in a deletion discussion was not a majoritarian process of voting, but more of a consensus-based process advising administrators about whether articles fit the project's notability standards. Around this same time, the “speedy deletion” process (known as CSD based on the new “criteria for speedy deletion”) was formalized into policy and given “royal assent” by Jimmy Wales, authorizing administrators to delete pages without discussion if they were patent nonsense/gibberish, entirely copyright violations (copied and pasted from another source), or if the original creator of the page requested it. A long discussion was held about expanding the



criteria for speedy deletions, and out of 24 proposed criteria, 6 were adopted, including those authorizing speedy deletion of attack pages or articles about “a real person that does not assert that person's importance or significance.”<sup>58</sup> With all the changes to the deletion-related processes, it was not easy for even the veterans in the project to not know how they were supposed to participate, as a number of Wikipedians opposed to the increasing formalization of process in the project argued. As AfD was separated from CSD and the XfD processes and gained further specification as to how exactly Wikipedians were to participate in them, a wide array of bots were developed and deployed to support the process in various ways.

I examined all the bot accounts that edited VfD/AfD from January 2005 to May 2015 (authorized and unauthorized), finding a wide array of bots performing various tasks (Table 3). Notably, no single bot has continuously operated in VfD/AfD during this decade of the project’s history, although the closest is Mathbot, which has performed various deletion-related tasks starting in August 2005, taking over AllyUnion’s original VFD Bot. Mathbot still operates in AfD in 2015, performing tasks related to archiving old discussions and curating daily logs listing all articles nominated for deletion that day. Other clerk bots were created to fix mistakes that Wikipedians would inevitably make when participating in the process, whether that was nominating a new article for deletion, commenting in an AfD as a discussant, or closing and archiving an AfD as an administrator. One of the longer-running AfD bots was DumbBOT, which among other tasks, would look for new pages titled “Wikipedia:Articles for deletion/[article name]”, which is the page Wikipedians were to create when nominating an article for deletion. Nominators were also supposed to “list” the article for deletion by adding it to the main list of ongoing AfDs and the daily log of new AfDs, but many did not, so DumbBOT did this for them. DumbBOT operated from 2005 to late 2011, until the bot’s source code was handed over to another bot developer to take it over; this new developer created a new account for it named Snotbot, then expand the code to add new tasks and functionalities.

In all, 39 bots have made at least 500 edits to VfD/AfD pages between 2005 and 2015, and dozens more have made fewer that to this process. Of the major bots operating in VfD/AfD, 27 are bots that were specifically developed to perform a task related to this process, while the remaining 12 were bots that operated across the project. The presence of project-wide bots in this specific process are interesting as they indicate not only algorithmic specification of a particular process, but also how there are common forms of administrative work which go beyond each of these individual administrative processes. For example, counter-vandalism bots like AntiVandalBot are active in these spaces, as these debates are sometimes vandalized by those who have their articles nominated for deletion. One of the most active bots in AfD over the past ten years was HagermanBot and its replacement SineBot, which automatically sign and date comments left without a signature and timestamp – something that frequently happens in these spaces. Bots have also appeared in AfD when performing project-wide find and replace tasks, such as when the shortlinks used to refer to other specialized spaces are renamed. This also indicates how issues around automation, decision-making, and norms are frequently interrelated, forming more of a patchwork of linked sites and software.

---

58

[https://en.wikipedia.org/w/index.php?title=Wikipedia:Criteria\\_for\\_speedy\\_deletion/Proposal&oldid=19360855](https://en.wikipedia.org/w/index.php?title=Wikipedia:Criteria_for_speedy_deletion/Proposal&oldid=19360855)

**Table 3: Bots that have made over 500 edits to pages in the Votes for Deletion or Articles for deletion processes**

Bot name	Work automated in VfD/AfD	Graph of edits per month to AfD/VfD, from Jan 2005 to May 2015 (top of y-axis is 1,000 edits)
AfD Bot	archiving, create daily log	
AfDStatBot	statistics and reports	
AnomieBOT	archiving, fixing redirects	
AntiVandalBot	counter-vandalism	
BetacommandBot	unauthorized bot (now banned)	
Crypticbot	add unlisted/orphaned VfDs to main page	
Cyberbot I	categorization, cleanup after closing	
Cyberbot II	adding notice when page is protected from	
Cydebot	various tasks around the mid-2006 AfD	
DumbBOT	adding unlisted AfDs to the main AfD page	
Erwin85Bot	categorization of AfDs	
Fluxbot	template substitution, cleanup after closing	
HagermanBot	comment autosigning	
Harej bot	categorization of AfDs	
Kakashi Bot	template standardization,	
LDBot	archiving, create daily log	
Legobot	archiving, categorization of AfDs	
Lowercase sigmabot	adding notice when page is protected from	
MartinBot	counter-vandalism	

**Table 3 continued:**

Bot name	Roles in VfD/AfD	Graph of edits per month to AfD/VfD, Jan 2005 to May 2015 (top of y-axis is 1,000 edits)
MartinBotIII	one-time find and replace for interlanguage	
Mathbot	daily logs, archiving, curating the main page	
MetsBot	template substitution, cleanup after closing	
One bot	archiving, relisting, categorization of AfDs	
RFC bot	categorization of AfDs	
RoryBot	fixing links to policy pages	
RussBot	fixing double-redirects	
SD5bot	one-time template replacement / rename	
SineBot	comment autosigning	
SmackBot	mass migration of old VfDs to AfD	
Snotbot	add unlisted AfDs to main page, statistics,	
SoxBot	archiving, cleanup after closing	
SporkBot	template substitution	
TBot	template substitution	
Tawkerbot4	counter-vandalism	
The wubbot	cleanup after closing and post-archiving tasks	
Uncle Ed's major work 'bot	unauthorized bot built to undo Uncle G's bots	
Uncle G's 'bot	tasks related to the 2005 VfD->AfD renaming	
Uncle G's major work 'bot	tasks related to the 2005 VfD->AfD renaming	
VfD Bot	archiving, curating the daily log and main page	

Comparing Articles for Deletion (AfD) and speedy deletion (CSD) illustrate quite different practices at work in the same fundamental task of deciding whether encyclopedia articles ought to be kept or deleted, which is made visible in the bespoke code developed to support this kind of work. Both processes are based around a similar approach to collective decision-making in which administrative actions like deleting an article are not to be made unilaterally. Both routinize the task of deciding whether an article ought to be deleted, but the bespoke code that operates in and around the two processes differently articulate work involved in this task to administrators and non-administrators. In AfD, this is position as a week-long discussion, with bots and other bespoke code working to create and maintain orderly, distinct spaces dedicated to a conversation about each article’s encyclopedic-ness. Bots operating in AfD help dedicated “AfDers” keep track of all the cases going through the process as well as locate and fix mistakes made by those less familiar with the process. In contrast, CSD is a fast-paced process explicitly designed to take the pressure off AfD, a space for articles that are not deemed worthy of a week-long discussion. As seen in my experience with the saveMLAK article, this process can be incredibly fast: it only took a few minutes for two Wikipedians to find my article and agree that it was unsuitable for inclusion. Yet even if there were a bot deployed to force a longer waiting period, such that it would be improper to CSD tag an article in the first fifteen minutes of its existence (perennially proposed and rejected), the speed is only an indicator of a more fundamental difference in process. Unlike AfD, where a “consensus” must be built in favor of deleting or keeping an article, CSD operates as a way get “four eyes” to look at an article and decide what to do about it. Yet in both spaces, there are similar assumptions implemented in bespoke code, such as in the automatic notification to original creators of articles when their articles are tagged or nominated for deletion.

### **3. Conclusion**

In this chapter, I argue that the emergence of these processes in Wikipedia was not an inevitable consequence, but rather result from specific decisions made with certain goals in mind as Wikipedians worked out what they wanted the project be. Such bots have not “taken over” Wikipedia in the way that many discourses of algorithmic agency portray, but instead show how the highly-active, veteran Wikipedians have largely made this project theirs. Furthermore, while I argue against bot-determinist narratives, the automation involved in such administrative processes is far from incidental; the development of an automated software agent that “clerks” for a process is an important moment in which decisions get made about how a particular administrative task ought to be performed. If the more far-inclusionist Wikipedians been allowed to operate a bot that would add their opposition to the deletion of any article, both the Articles for Deletion process and the content of Wikipedia would look quite different today. Similarly, if the more far-deletionist Wikipedians had expanded the speedy deletion criteria such that it effectively replace the slower, more discussion-based Articles for Deletion process, Wikipedia would also look differently.

As Wikipedians today gain an increasing awareness of the issues of newcomer socialization (particularly an issue given the dramatic inequalities in participation for women and persons of color), these administrative processes Wikipedians have developed for themselves is increasingly called into question – from both inside and outside the Wikipedian

community. Projects like the Teahouse have recently emerged as specialized spaces supported by bots and other bespoke code in ways that are specifically designed to make it easier for newcomers to get help with the rest of the site. I am not willing to predict whether the project will continue its decade-long trend of increasing formalization, or whether reformist efforts will ultimately succeed. But in any case, these broader issues will ultimately involve the development and re-development of bots and other bespoke code – or their inevitably controversial removal from the project, replaced by code more directly incorporated into the “stock” server-side codebase.

## Section 3: Blockbots

(chapters 7 & 8)

### 1. Lessons learned from Wikipedia

In the previous sections, I advocated taking a more situated and long-term algorithms-in-the-making approach in studying Wikipedia, which expands the analytic frame of ‘algorithms’ far beyond a focus on source code. Instead of seeing bots as software agents that ruthlessly enforce the politics and power relations built into their source code, I found that bots were often sites of negotiation and contestation for a wide set of normative issues in Wikipedia. The deeper I began to look at bots, the more humans I found in the mix – contrary to dominant discourses that position automation as a zero-sum game, taking agency away from humans and delegating it to machines. It is important to focus on the role that a bot’s operator plays in maintaining a software agent’s technological infrastructure, as well as the role the bot’s operator plays in relation to other people who have their own ideas about what the bot should be doing and how it ought to do it (if at all). In my analytical frame, bots started off as source code powering automated software agents, but they expanded to become projects of collective sensemaking about what Wikipedia is and ought to be. The delegation of a task to a software agent is a key moment that brings together people who have particular ideas about the task in question. In that moment, people work out not just whether the task ought to be performed by a bot in a given way (like automatically signing and timestamping comments) but also broader ideals about how the space they collectively inhabit ought to operate (like norms of deliberative democracy versus editorial autonomy).

To understand how bots are able to serve such a role within sites of collective sensemaking, I focused on the articulation work bots were delegated, finding that they were rarely fully delegated all aspects of a particular task or decision. Rather, bots were delegated the work required to keep humans organized and on task within a particular administrative apparatus, serving more as ‘clerks’ than judges or juries. This shifts discussions on governance and software code dramatically, moving away from Lessig’s focus on how ‘server sovereigns’ write code to delegate their dictatorial authority to algorithmic police officers. While there certainly are algorithmic police officers in Wikipedia (which patrol for spam and vandalism), my focus on the articulation work performed by bots as clerks reveals how bots are more subtly woven into the fabric of Wikipedia’s organizational culture. These kinds of clerk bots make possible increasingly complex forms of collective action in Wikipedia, making it possible for Wikipedians to govern their project in particular ways. Far from bots coming on the scene to take agency and decision-making capacity away from humans, the delegation of articulation work enables veteran Wikipedians to efficiently participate in a broad range of administrative decisions. Furthermore, in using ethnographic and historical methods to study how bots have supported articulation work over time, I showed how high-level issues – like Wikipedia’s factions of inclusionists vs. deletionists – play out in part through decisions made about how bots ought to support articulation work.

## **2. Twitter’s blockbots as a new case to apply the algorithms-in-the-making approach**

### **2.1 Introducing harassment in Twitter and various responses**

In the chapters in the next section, I ask how this algorithms-in-the-making approach applies to other systems in which automated software agents are performing governance work. I introduce bot-based collective blocklists (or “blockbots”) on Twitter, which have generally been created by anti-harassment advocates and activists, who argue that harassment has particularly proliferated on Twitter. The affordances provided by the social networking site easily facilitate unsolicited interactions, which can range from the serendipitous to the abusive. By default, any Twitter user can have notifications sent to any other user, and hashtag streams created around particular events or issues are also easily ‘hijacked.’ Twitter does have a built-in blocking feature that lets users hide specific accounts from their experience of the site, but individually blocking such accounts can be a Sisyphean task. In response, blockbots implement collective blocklists, in which an individual, group, or algorithmic procedure (or a combination) curates a list of blockworthy Twitter accounts, based on various criteria and processes. Then, using an automated software agent (or a “bot”), any Twitter user can subscribe to the blockbot and have every account on blocklist automatically be made invisible to them on Twitter. Blockbots are bespoke algorithmic systems, in that they run alongside a core platform and extend the features and functionality, but have been designed, developed, and deployed by volunteers who are relatively independent from staff at Twitter, Inc. In studying blockbots in Twitter, I find that there are many similar issues that can be explored when taking bots to be projects of collective sensemaking around the delegation of articulation work. There are also many differences between Wikipedia and Twitter (both as sites in general and in how bots operate in those sites), which means that some issues around bots play out quite differently.

As blockbots have come into Twitter around the issue of harassment, they have provoked substantial discussion and debate about the moderation of privately-owned public spaces by counterpublic groups, as well as the role of algorithmic systems in the governance of centrally-hosted digitally-mediated environments. In other words, blockbots are important to study not only because of the consequences of the specific tasks they are delegated; they are also important to study because of the broader role they play in how people are collectively making sense of what Twitter as a social networking site is and ought to be. Blockbots have passionate advocates and critics, who both frequently call on Twitter, Inc to act about this issue: anti-harassment activists argue that the corporation should give people better and easier ways to moderate their own experiences on the site, while opponents of blockbots call for the bots to be banned and have their API keys revoked. Twitter’s recent partnership with Women, Action, and the Media! (WAM!) to provide an independent, secondary process for reviewing harassing tweets (Matias et al., 2015) speaks to a growing recognition by the company that harassment is an issue, and not one that Twitter, Inc. as a server sovereign will be able to tackle on its own. Yet blockbots are different than the Twitter/WAM! partnership, in that they are not officially partnered with the company nor are blockbots integrated into the existing processes for reviewing and removing content that violates the site’s rules. Instead, blockbots are typically used to block accounts that have not been found to violate Twitter’s rules, but still are considered to engage in harassment by those who must encounter those accounts in their everyday use of Twitter.

As Marwick & Miller (2014) note in their extensive review of the literature on online harassment, much of the scholarship and public commentary on online harassment has focused on the roles of dominant, formal institutions social institutions, such as policymakers, law enforcement, schools, or companies that own and operate social media sites. However, Marwick & Miller argue that online harassment demands an examination of both these formal institutions as well as the local practices and social structures in which harassment takes place. Blockbots on Twitter show how the complicated and multifaceted question about what ought to be done about harassment in a social networking site like Twitter is not at all limited to the decisions and designed made by staff at Twitter, Inc. Furthermore, blockbots show that algorithmically-supported filtering and moderation of content does not need to operate on the frequently criticized model exemplified by Facebook's filtered news feed. Through the development of bespoke algorithmic systems that extend the functionality and affordances of Twitter as a networked public space (boyd, 2010), these counterpublic groups (Fraser, 1990) are imagining, implementing, and iterating on their own understandings about what kind of space Twitter ought to be and how it ought to be governed. The development of these blockbots involve developing not just software code, but also ideas about what Twitter is as a public space. These ideas are expressed through code and a wide set of other media and discourse, as people build their own social and technical systems for identifying harassment. This involves reconfigurations of a bot's software code, as people work out how they ought to collectively moderate their own experiences on the site.

## **2.2 Contrasting blockbots with malicious bots**

From a software development standpoint, most bots in Twitter (including but not limited to blockbots) are quite similar to the malicious and productive bots that operate in other major sites, which I have previously discussed in this dissertation. Bots are typically powered through shorter scripts that rely on larger software libraries, specifically developed to query a site's Application Programming Interface (API). An API is specifically designed to efficiently support third-party applications (including but not limited to bots). This means a bot developer must not only write the code of the bot, but also get access to query the API and find a server where their automated software agent can run. This server can be anything from a developer's personal computer to a sophisticated cloud computing infrastructure like Amazon EC2. In the standard use case of bots, a software program is linked to a given user account with the user's authorization, which is conducted in the same way that users authorize mobile apps to post to the site, for example. Once authorized, the program can query the API to request that the site perform an action on behalf of that account. In sites that have well-supported APIs (which include Twitter, Wikipedia, and reddit), almost any action that can be done manually by human users using the web or mobile interface. In contrast, some sites heavily restrict what actions can be taken by a program through an API (such as Facebook), and others do not have an API at all. APIs are controlled by the people who own and operate a site, meaning that third-party applications operate with a kind of tacit approval. Staff at Twitter, Inc. do revoke API access for bots and apps that violate their understanding of what Twitter ought to be, as can be seen in the recent revocation of API access for sites like Politwoops, which monitored politicians' Twitter accounts for deleted tweets (Hern, 2015).



### 2.3 Contrasting blockbots with other automated software agents in Twitter

Many Twitter bots automatically detect some kind of alleged normative violation and send a reply to the sender, which I discuss to distinguish from blockbots. Such norm enforcing auto-reply bots range from the whimsical to the sincere, including: BuzzFeedPSA, which tells those who reply to Donald Trump to not engage or encourage him; RedScareBot, which sends McCarthy-ist replies to tweets mentioning “commies” or “communism”; and she\_not\_he, created to tell those who referred to Caitlyn Jenner as “he” (she came out as a trans woman in June 2015) that “she” is the correct pronoun. Twitter’s automation policy generally prohibits bots that auto-reply based on keyword searches, which is seen as a kind of spam. Some of these more norm-enforcing auto-reply bots have been suspended, while others continue to operate. Twitter’s automation policy is vague, and according to bot developers who have knowledge and/or experience with the process, it appears to be enforced on an ad-hoc basis: staff at Twitter, Inc. generally only review bots that are reported as abusive or as spam by users.

Auto-reply bots like RedScareBot and she\_not\_he do perform automated normative moderation, but bot-based collective blocklists (or blockbots) are quite different in how they use automation to support the moderation of a privately-owned public space. Blockbots were initially created around the broader issue of online harassment, which has received substantially more attention specifically in Twitter in 2014 and 2015, due to a number of more and less organized harassment campaigns that unfolded on Twitter – including the sustained anti-feminist GamerGate movement and multiple shorter cases in which celebrities have been harassed (Chess & Shaw, 2015; Heron, Belford, & Goker, 2014; Matias et al., 2015). I investigated the historical development of these algorithmic systems, and the earliest bot-based collective blocklists that operated in Twitter were created specifically and explicitly in the context of a perceived governance gap around harassment. In 2012, some targets of harassment (and their allies) were arguing that Twitter, Inc. was not doing nearly enough to respond, particularly against those who were using the social networking site to launch large-scale, coordinated campaigns against particular individuals. Such anti-harassment activism and organizing took place in smaller online communities that did not make as many headlines in mainstream or social media as they have in 2014 and 2015, particularly playing out in communities that had fractured on the issues of feminism and social justice.

Many people who were facing coordinated harassment campaigns for their feminist political stances called for changes to the site’s policies, enforcement mechanisms, and user interface features in order to minimize the impact of harassment and take action against identified harassers. In addition to this more discursive anti-harassment work petitioning Twitter, Inc. as a server sovereign, other advocates and activists turned to software-based approaches to help their fellow community members moderate their own experiences on Twitter. Such software tools were based on automated software agents that extended the built-in affordances of Twitter by making creative use of features developed largely to support third-party clients. Blockbots made it possible for people to collectively curate blocklists of those they identified as harassers, then synchronize these blocklists with subscribers. With a few clicks, a Twitter user could subscribe to a blocklist and automatically no longer see any tweets or notifications from anyone added to that list. Some of these blocklists are curated by single individuals, others by tight-knit groups, and a final set are algorithmically generated based on

various methods of data collection and analysis. In fact, the modularity of the blockbot approach means that a bot-based collective blacklist can theoretically be created and curated by any given socio-technical system (e.g. an open wiki in which anyone can edit or comment, a vote-based process restricted to a tight-knit group, a ‘team of one’ using Twitter’s default blocking interface, an automated agent running predictive models, etc.), then the list can be automatically synchronized to subscribers using the same computational infrastructure. Blockbots as a computational entity encapsulate an organization that collectively curates a list of blockworthy accounts; yet in doing so, the blockbot-as-organization also implements a particular understanding of what harassment is, as well as what Twitter is and ought to be as a public space.

### **3. Overview of the two chapters**

In the following two chapters on Twitter blockbots, I apply the algorithms-in-the-making approach I developed in cases of bots in Wikipedia. In the previous two sections, I made two analytical expansions of what a bot was. In the first section, I showed how bots were far more than their source code. They were accomplishments that had to be continuously operated and maintained by their developers, who are responsible for ensuring that the software agent could continue to perform the programmed task. This includes the individual effort of maintaining computational infrastructure like servers, power, and Internet access, but also includes the broader efforts of representing and negotiating on behalf of the bot. In these negotiations, bots are projects of collective sensemaking, in which developers and non-developers seek to build a consensus for a given task. Furthermore, such projects of collective sensemaking are not just about whether the task in question is a good or bad idea; bots can also provoke and resolve high-level normative issues about how Wikipedia as a community is governed. Then, in the second section, I used the concept of articulation work to focus on the kinds of work that bots perform in Wikipedia. Contrary to common discourses that critique automation for taking away agency and decision-making responsibility from humans, I found that bots were predominantly delegated the work of supporting humans who had to make commonly reoccurring decisions. In other words, bots in Wikipedia are not autonomously and independently deciding what articles ought to be deleted or which users ought to be blocked. They are instead structuring and supporting a particular way in which human Wikipedians can more efficiently make these kinds of decisions, particularly at scale.

In the first chapter of this section, I apply the framework and findings from the first section, in which I move from seeing bots are purely abstract, algorithmic agents constituted through source code to bots as projects that involve developers and non-developers. I make this argument by relating a “tutorial ethnography” – a kind of ethnographic composite account in which I give detailed, step-by-step instructions about how to develop and deploy a blockbot. In relating this account, I viscerally demonstrate how the work of building a blockbot requires a substantial amount of infrastructural work by the developer/operator. The step-by-step process is a quite literal way of taking the reader through an algorithms-in-the-making approach. In giving this account, I show not only that there is a substantial amount of infrastructural work performed by a blockbot operator which goes beyond programming a software agent; I also show how this particular kind of work matters to what a blockbot ultimately becomes. I first

take the reader through an individual process of the work involved in sharing their own blocklist with others, which is how many bot-based collective blockbots initially began. After I have explained how to build a system that will let others automatically block any account that you block using your personal account, I then detail the work involved in building a system in which a group can collectively curate a shared blocklist. While the tutorial is a fictional account, it does faithfully detail how to build a blockbot in the general computational framework that almost all major blockbots use.

Like with administrative clerkbots in Wikipedia, this tutorial-ethnography shows how the work that blockbots are delegated is articulation work – it is the work of keeping more human agents organized and on-task, so that they can efficiently complete the project at hand. In the case of blockbots, the project at hand is curating a shared list of blockworthy accounts, which is then automatically blocked by everyone who subscribes to the blocklist. Yet there are many different ways in which a group of people can collectively curate a list of blockworthy accounts, just as there are many different ways in which a group of people can collectively decide what articles ought to be kept or deleted from Wikipedia. Just as clerkbots in Wikipedia provoke and resolve high-level debates about what Wikipedia is and ought to be through the articulation work they are delegated, so do clerkbots perform a similar function. To explore this, in the second chapter of this section, I present several specific empirical cases of blockbots – some of which have substantially changed their processes for curating a collective blocklist over time.

## Chapter 7: The blockbot tutorial-ethnography: a composite account

### 1. Introduction

#### 1.1 A composite account

In this chapter, I present my own ‘unpacking of the black box’ of blockbots as algorithmic systems through a kind of composite ethnographic account, in which I identify various elements that are common across many blockbots. Composites are one common convention among ethnographers (Murchison, 2010), where an individual or event will be described in the abstract out of specific cases observed by the ethnographer. A composite is a kind of generalized account; although composites are not intended to be statistically representative, they are often crafted to emphasize what is particular to the setting under investigation compared to other settings. Many are quasi-fictional accounts assembled out of the empirical observations made by the ethnographer. As such composites have a contested status both among ethnographers and between ethnographers and non-ethnographers.<sup>59</sup> Often composites are formed to protect the anonymity of the people studied, but as some critics have argued, this anonymization also can be a form of unacknowledged theory-building, presenting some kind of ideal type (Marcus & Cushman, 1982). As Marcus and Cushman critique, this practice is one of many tropes of “ethnographic realism” – a descriptivist epistemology that portrays the ethnographer as a kind of neutral camera, convincing the reader “you are there, because I was there.”

For decades, anthropological theorizing about ethnographies after the “postmodern turn” – often marked in anthropology by the publication of the edited volume *Writing Culture* (Clifford & Marcus, 1986) – tends to emphasize the constructed nature of all ethnographic texts as form of literature that mediates between the ethnographer, the people studied, and the reader. Such a text does not purely represent the people studied in their context, but represents an encounter that the ethnographer has with those people, in which the ethnographer and the people studied bring their own understandings, concerns, literacies, skills, contexts, and assumptions. For example, speaking about the then-recent controversy over whether Margaret Mead’s depictions of Samoan life were accurate, Clifford and Marcus write: “Was Margaret Mead simply wrong about Samoa as has recently been claimed? Or was her image of an exotic land a partial truth reflecting the concerns of her time and a complex encounter with Samoans?” (Clifford & Marcus, 1986, p. i) Above all, such scholarship emphasizes the need for the ethnographer to actively reflect on the triad relationship between the people studied, the ethnographer, and the reader, rather than put forth a description that contextualizes a people without contextualizing the person who wrote it.

#### 1.2 The genre of the tutorial

With these critiques in mind, I do not want to unintentionally fall into the tropes of “ethnographic realism” in giving a composite account of what it is like to be a blockbot developer. As I have far less experience in the area of blockbots as I have with bots in

---

<sup>59</sup> For example, see recent controversies over Alice Goffman’s *On the Run: Fugitive Life in an American City*

Wikipedia, I feel even less comfortable with such an account being taken by the reader as a way to fully immerse themselves in the lived experiences of the blockbot developers I have studied. Instead, I intentionally subvert the tropes of ethnographic realism by presenting my composite in the genre of the software development tutorial, giving step-by-step instructions about how someone would go about building a blockbot themselves in June 2015. This tutorial-ethnography is intended to introduce various elements and activities in and around the development of blockbots, ostensibly for readers who have little to no experience with how such systems operate. While this tutorial gives important details about how blockbots operate (which will be useful in the later cases presented in the next chapter), it also speaks to the broader issues in this dissertation about algorithmic transparency/opacity, specifically about what it means to “open the black box.” First (as previously discussed), this tutorial shows how such unpacking involves far more than the encoded procedures of algorithms, situating these algorithms in broader computational infrastructures in which they operate. Second, my use of this specific genre of the tutorial gives the reader a chance to reflect on how their own experience in software development (or lack thereof) relates to their ability to follow along as these computational procedures are detailed in a second-person declarative narrative. Finally, this tutorial establishes a background for the kinds of cases which I will examine in the second chapter in this section, which involve specific details about how blockbots operate that will be more legible after reading this tutorial.

The tutorial is a common genre among software developers, sometimes referred to as ‘how-tos’ or ‘getting started’ guides. A tutorial generally involves a text that weaves second-person declarative descriptions (e.g. “before you can query the API, you need to import the library”) with excerpts of software code and step-by-step instructions needed to carry out these tasks. I find these second-person declaratives compelling for an ethnographic composite, because this pushes the implicit claim of ethnographic realism to its limits. This genre provokes the reader to think about to what extent they are experiencing (or capable of experiencing) the kinds of activities that the tutorial tells them they are conducting in a second-person point of view. In many cases, readers can even attempt follow along and attempt to build their own blockbot, but this ability will vary wildly based on the reader’s existing familiarity with both software development in general and the specific kinds of languages, libraries, protocols, and interfaces that are assembled in this explication of blockbot development. I also subvert the genre of the tutorial by inserting sections that use these same literary conventions to describe exceptional events that can happen in the course of blockbot development, which are composite events from multiple blockbot developers with details changed and removed to protect anonymity.

Like ethnographic texts, the construction of a programming tutorial involves creating a narrative; for the tutorial, it is often a narrative about how various aspects of a software program fit together around an increasingly-complex problem. Some will simply go line by line through a program they have already written, but I find those less compelling and instructive. In contrast, my favorite tutorials usually begin with a conceptually simple core task (which can typically be performed in a few lines of code) and then gradually expand this task. Many tutorials also make heavy use of diagrams and visualizations as ways to make each step visible, sometimes beginning with a single diagram and then expanding it with each step. Because of this instructional focus, many guides include a significant amount of expository text that discusses

why each particular element or step is needed and how it connects to the broader task at hand. Often there is more of this background text than is needed for some readers with more experience in software development, who can skip a descriptive section and move on to the code that implements it, if they choose. There is also the inevitable possibility that the tutorial will be incomplete for other readers, who may not understand why a particular line of code or protocol specification operates in the way it is presented. I rely on these aspects of the genre of the tutorial to discuss aspects of bot development that can be cast as “purely technical” as well as those that illustrate the complex socio-technical entanglement involved in blockbot development in Twitter.

## 2. The tutorial-ethnography

### 2.0: blocking on Twitter’s interface without a bot

If you have an account on Twitter, you can block other Twitter accounts. Any user account can block any other user account, which for most people involves clicking on a series of buttons in a web or mobile interface. Blocking in Twitter is quite similar to the ‘killfiles’ of USENET mailing lists, although blocking is a feature supported and implemented server-side by Twitter, rather than bespoke code. If you (as someone with an account on Twitter) block someone, then their account generally becomes invisible to your account on Twitter. They cannot send you direct messages, you do not get notifications if they mention you in their tweets, you do not see their tweets if you search for a keyword or hashtag, and you will not see their tweets in your timeline (even if someone you follow retweets one of their tweets). When you block (or unblock) someone, there is a change made to a database table on Twitter’s servers, containing a list of the users who your account has blocked. This table is routinely queried when serving content via the web and mobile interface to implement these partial visibilities. One consequence of this is that all Twitter clients which operate by querying the API (which includes the web interface) first retrieve a list of tweets in a user’s timeline or a list of notifications, and then filter those tweets or notifications based on the client’s blocklist.

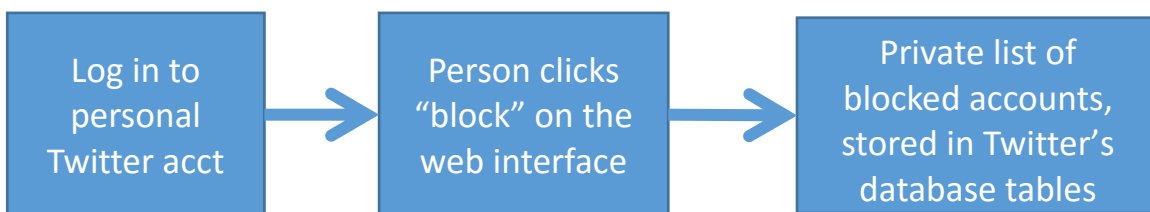


Figure 30: Diagram of a person blocking a user

### 2.1: the API, queried by a software program

The typical way for someone to block a user on Twitter is through Twitter’s own web or mobile interface, and this affordance has been built into Twitter’s interfaces for years. In addition, the developers at Twitter have made an API (or Application Programming Interface) available to anyone who requests access (approval is automatic and given to all Twitter accounts by default within seconds, but can be revoked for those that violate terms of service). An API

gives software developers a direct connection to Twitter, so that a computer program can act on behalf of a user account in a secure, standardized, and efficient way. In a simple example, if you're writing a program that integrates with Twitter – such as a camera app that automatically posts photos to social media sites – you can call the API to do this work behind the scenes. Most major programming languages have sophisticated libraries for querying the APIs behind Twitter, Facebook, Wikipedia, reddit, and other sites, so implementing this in code is as simple as:

```
# Import the library to query the Twitter API
from twython import Twython

# Login to Twitter with a user's account
twitter = Twython(/* LOGIN DETAILS for @subscriber */)

# Post a tweet with the photo attached
twitter.updateStatusWithMedia('photo.jpg', status='I just took this photo!')
```

Of course, you (with the username “@subscriber”) need to log in with your password to authorize the program to post on your behalf, but once you authenticate, it can keep acting on your behalf without your intervention. A program with an API can typically do anything that a person can with a web interface, and sometimes even more. On Twitter, an API-authorized program can follow, unfollow, block, or unblock users. It can get a list of all of your followers – or anyone else’s followers, for that matter. It can then store this information in a structured list and do lots of interesting and complex things with them; things that would be very hard and/or tedious for a human to do manually. If you found someone who follows a bunch of interesting accounts on Twitter (we will refer to them with the username “@blocker”), a program can use the API to get a list of all of their followers. Then that program, logged in as @subscriber, can have the API tell Twitter to start following all the accounts @blocker follows. It takes about three lines of code – or five, if we haven’t already imported the library and authenticated an account.

```
# Create a list and populate it with the accounts of everyone who @blocker follows
blocker_following = twitter.get_friends_ids(screen_name="blocker")

# Go through that list of everyone @blocker follows, one user id at a time...
for following_id in blocker_following['ids']:

    # And for each user, tell Twitter to have the logged in user follow them
    twitter.create_friendship(user_id=following_id)
```



Figure 31: Getting a list of accounts @blocker follows. Information flows from list of accounts on Twitter's databases to the software program

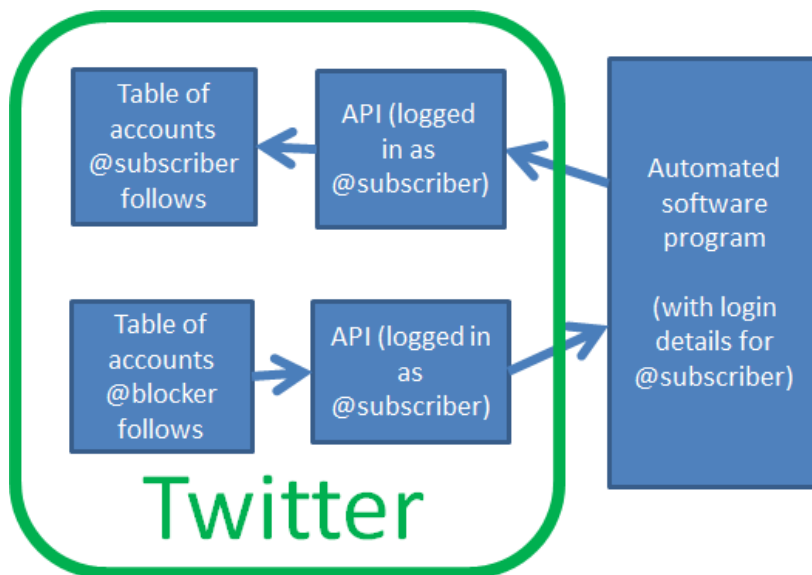


Figure 32: Having @subscriber follow everyone @blocker follows using the API. Information flows from Twitter's servers to the software program then back to Twitter's servers. The database tables and the APIs are owned, operated, developed, and hosted by Twitter. The automated software program is not.

## 2.2: Synchronizing blocklists

You could also do the inverse, getting a list of everyone a user account follows and then blocking every single person. In that case, instead of calling the function `twitter.create_friendship` in the last line, you'd call `twitter.create_block`. The more likely scenario is that you found someone who you trust who actively maintains a blocklist of abusive accounts. Wouldn't it be great if you could get a list of everyone who that person blocks, and then have your account block them too? That, you might imagine, could also be three lines of code:



```
# Create a list and populate it with everyone who @blocker blocks
blocker_blocking = twitter.list_block_ids(screen_name="blocker")

# Go through that list of everyone @blocker blocks, one user id at a time...
for blocking_id in blocker_blocking['ids']:

    # And for each user, tell Twitter to have the logged in user block them
    twitter.create_block(user_id=blocking_id)
```

The barrier to this is that development staff at Twitter have deemed blocklists to be private information<sup>60</sup>, and implemented this understanding of privacy directly into the API. While the list of someone's followers is considered public and can be queried by anyone, you are the only account authorized to query your account's own blocklist. So even for those who know computer programming and are able to get one of these API-based scripts online, there needs to be another layer for a collective blocklist to operate. In the most basic version that we have at this point, you have to convince your friend with a good blocklist to give you their Twitter password. Your software agent will log in as your friend (@blocker), get a list of who this person blocks, log in as you (@subscriber), and then update your blocklist accordingly.

```
# Import the library to query the Twitter API
from twython import Twython

# Login to Twitter with each user's account
Twitter_subscriber = Twython(/* LOGIN DETAILS for @subscriber */)
Twitter_blocker = Twython(/* LOGIN DETAILS for @blocker */)

# While logged into @blocker's account, get a list of everyone @blocker has blocked
blocker_blocking = twitter_blocker.list_block_ids(screen_name="blocker")

# Go through that list of everyone @blocker blocks, one user id at a time...
for blocking_id in blocker_blocking['ids']:

    # And logged into @subscriber's account, tell Twitter to block them
    Twitter_subscriber.create_block(user_id=blocking_id)
```

---

<sup>60</sup> As of July 2015, a user account cannot get a list of all accounts that are blocking them and users are not sent notifications when they are blocked. However, a user can tell if a specific user is blocking them by visiting that account's profile.

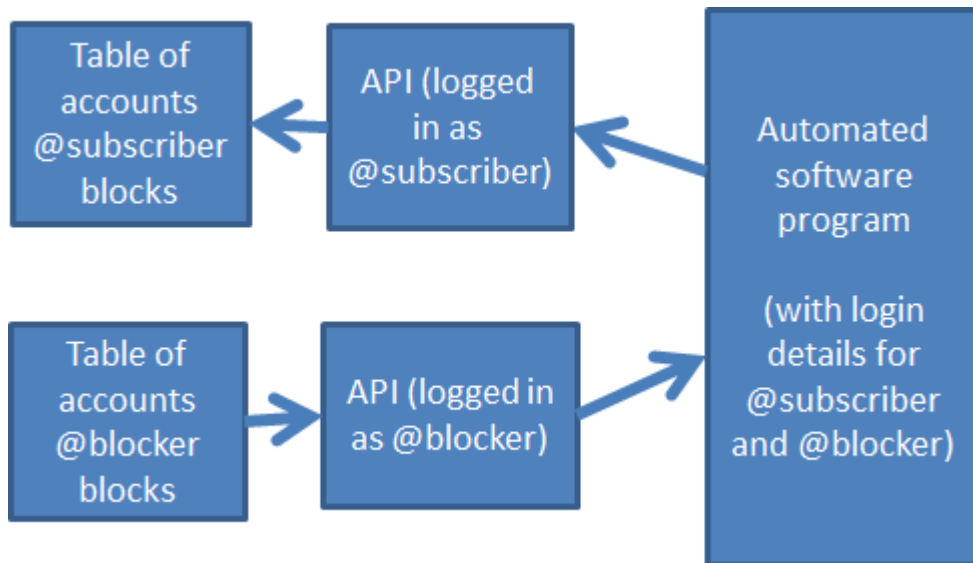


Figure 33: Having @subscriber block everyone @blocker blocks using the API. Information flows from Twitter's servers to the software program then back to Twitter's servers. The software program needs passwords from both accounts.

Then, in order to have this program run on a regular basis on a computer running Linux, you can add a single line can be added to the operating system's list of scheduled programs (called *cron*). For example, to have the script (saved as `updateblocks.py`) run every day at 6:30pm local time, you would add the following line would be added to the computer's *crontab* file:

```
30 18 * * * python /home/blocker/updateblocks.py
```

This version will be somewhat inefficient, since it will always retrieve every account @blocker blocks and send a request to Twitter's servers for @subscriber to block those accounts – many of which will already be blocked, if it is routinely updating. However, Twitter's API will gracefully decline the request to block an account that @subscriber is already blocking, and your program can continue to the next request without any problem. You could add code that would check for duplicates, but this is not required – although it is a good idea, especially for large blocklists.

### 2.3: oAuth

In the six lines of python code and the one line cron directive written above, you can implement the fundamentals of a continuously-updating bot-based shared blocklist. However, in order for this to work, you as the developer of the bot needs access to the passwords of both @subscriber and @blocker. This is a very risky proposition, even if those people trust you and think you have their best interests at heart. You could also lose access to update their accounts anytime they change their passwords, needing them to send you their login details for this to work. Thankfully, you are not the only person to have such a problem – this is something any developer of a third-party client or linked application has to deal with. In response to the

growing demand for third-party clients and other similar programs that query using APIs, there has been a widespread standard for selectively delegated access called OAuth.

With OAuth, a user can direct a platform like Twitter to generate a unique, cryptographically-secure ‘token’ (or a long string of characters) which grants specific access to that user’s account through an API. With such a token, your application can log in and act on behalf of an account as if they were the user who operates that account. These tokens are analogous to secondary passwords, and they can be restricted in terms of use: for example, in some configurations, you can request that a user only grant your bot access to their blocklists, without granting it the ability to post tweets. These tokens remain valid even if the user changes their primary password, and they can be individually revoked at any time without requiring that the user change their primary passwords. This standard was spearheaded in part by a Twitter employee in 2006 is now an IETF standard (RFC 5849; Hammer-Lahav, 2010) that is widely-used by many major web companies for selectively delegated account authentication. The most recent version of the OpenID authentication layer – in which you can log into third-party sites using your Google, Facebook, or Twitter account, for example – uses the OAuth standard.

Without OAuth (or a similar selectively delegated access standard), Twitter users would be required to give their passwords to potentially unknown individuals in order to have a bot update their blocklists on their behalf. With OAuth, a user can quickly, easily, and securely grant blockbots access – the exchange of tokens is even done behind the scenes. This involves setting up a web application in Twitter’s developer center, which registers your program with Twitter’s OAuth authorization system. For someone to sign up with your bot and grant it access, you ask Twitter for a long, dynamically-generated URL that the subscriber has to visit while logged in to Twitter. They will see a screen like in Figure 6, and if they sign in and authorize it, Twitter will give your application their specific OAuth key. You store their key and use this to log in as their account, instead of needing their password. With OAuth, we expand the diagram in Figure 5 with two boxes on the right side: the blocker and the subscriber have to both authorize the application, which stores the OAuth tokens – hopefully securely, given what kinds of access they provide.

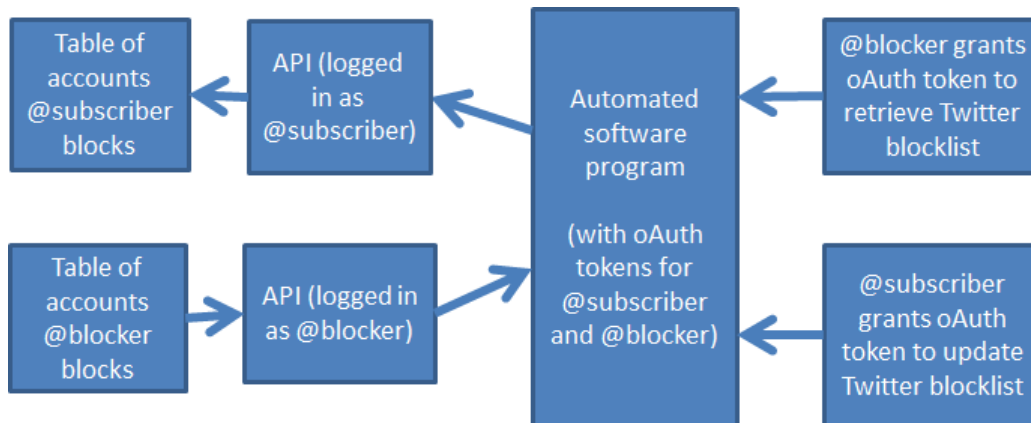


Figure 34: Synchronized blocklist using OAuth. Authorization information flows from both users to the program.

This diagram depicts this process of granting an oAuth token for @blocker and @subscriber.

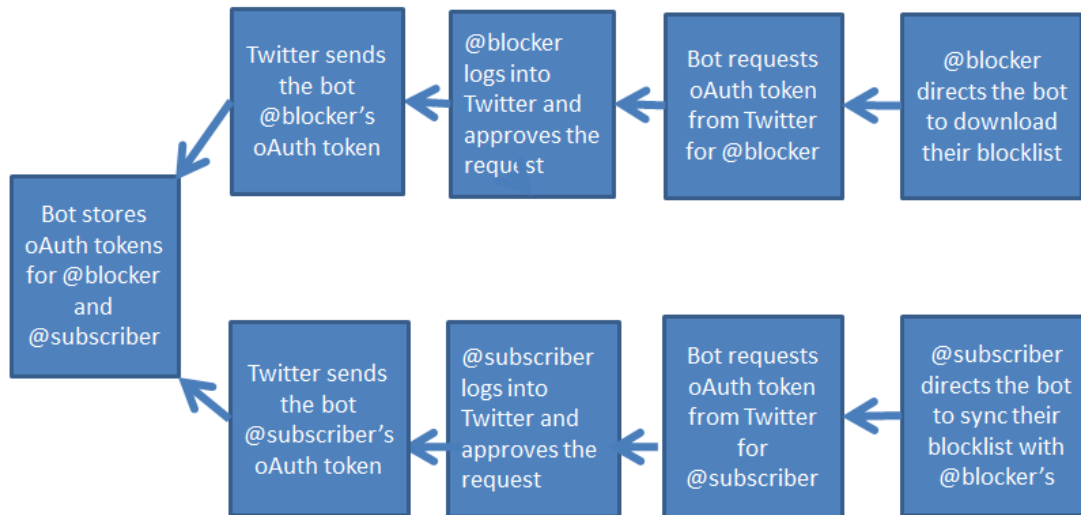


Figure 35: Expanded diagram of the oAuth process, which when complete, lets the bot operate as detailed in Figure 35

## 2.4: coordinating servers: blocktogether.org

At this point in the tutorial, you might be wondering how you are supposed to set up such a service where your subscribers can grant your blockbot access to their blocklists using oAuth. This would take far, far more than six lines of code. While some of the first blockbots had to develop this code on their own, there is now a general purpose service for Twitter blockbots called [blocktogether.org](http://blocktogether.org) that has standardized this process and made it incredibly easy to run a blockbot. We won't go into detail about how to build your own version, but the source code for [blocktogether.org](http://blocktogether.org) is open source and up at GitHub (it is over 5,000 lines of code, not including the frameworks and libraries it uses). The way this service works is that you (as a potential subscriber to a blocklist, someone who wants to share their blocklist with others, or both) authenticate your Twitter account with [blocktogether.org](http://blocktogether.org). This uses oAuth to give its automated software programs API access to download your account's existing blocklist and/or update your account's blocklist on your behalf.

Once you authenticate, a parallel user account is also created on [blocktogether.org](http://blocktogether.org)'s servers for you, keeping track of which blocklists you subscribe to and what users you have blocked. Once this is all set up, you can share your own blocklist with others if you choose (which generates a unique, practically impossible-to-guess URL for you to share). You can also subscribe to someone else's blocklist by going to that URL and clicking the "block all and subscribe" button. [Blocktogether.org](http://blocktogether.org) also has two default blocklists that are dynamically

generated and prominently displayed on the sign-up page: accounts that mention you which were created less than a week ago, and accounts that mention you which have less than 15 followers. In all, this means that with [blocktogether.org](http://blocktogether.org), the task that we had initially set out to perform with a bot we wrote ourselves – you want to automatically block everyone who your friend blocks – can be entirely performed by this general purpose bot-based service. Your friend can share their blocklists with as many people as they want, and you can subscribe to as many blocklists as you like, and there only needs to be one system that is authorized to retrieve and update everyone’s blocklists.

Hosting such a site might seem like a big responsibility, given that it is a centralized repository for oAuth tokens and potentially private blocklists – and it is. Yet [blocktogether.org](http://blocktogether.org) is run by Jacob Hoffman-Andrews, who as of mid-2015 is a senior staff technologist at the Electronic Frontier Foundation and was a former security engineer at Twitter, Inc. While at Twitter, Hoffman-Andrews pushed for improved security and encryption in the social networking site long before Snowden leaks about the NSA’s extensive surveillance programs. Since such revelations, he has been a strong advocate for implementing security measures that will make it increasingly difficult for even an agency with the resources of the NSA to surveil on people. With [blocktogether.org](http://blocktogether.org), your friend doesn’t need to trust that you’ve got the kind of background and resources to run a secure blocklist sharing system. Furthermore, you should know that while the code in the previous sections of this tutorial function to support blocklists, they are quite vulnerable to a wide variety of attacks – especially when you are storing the passwords to accounts in the source code of a program.

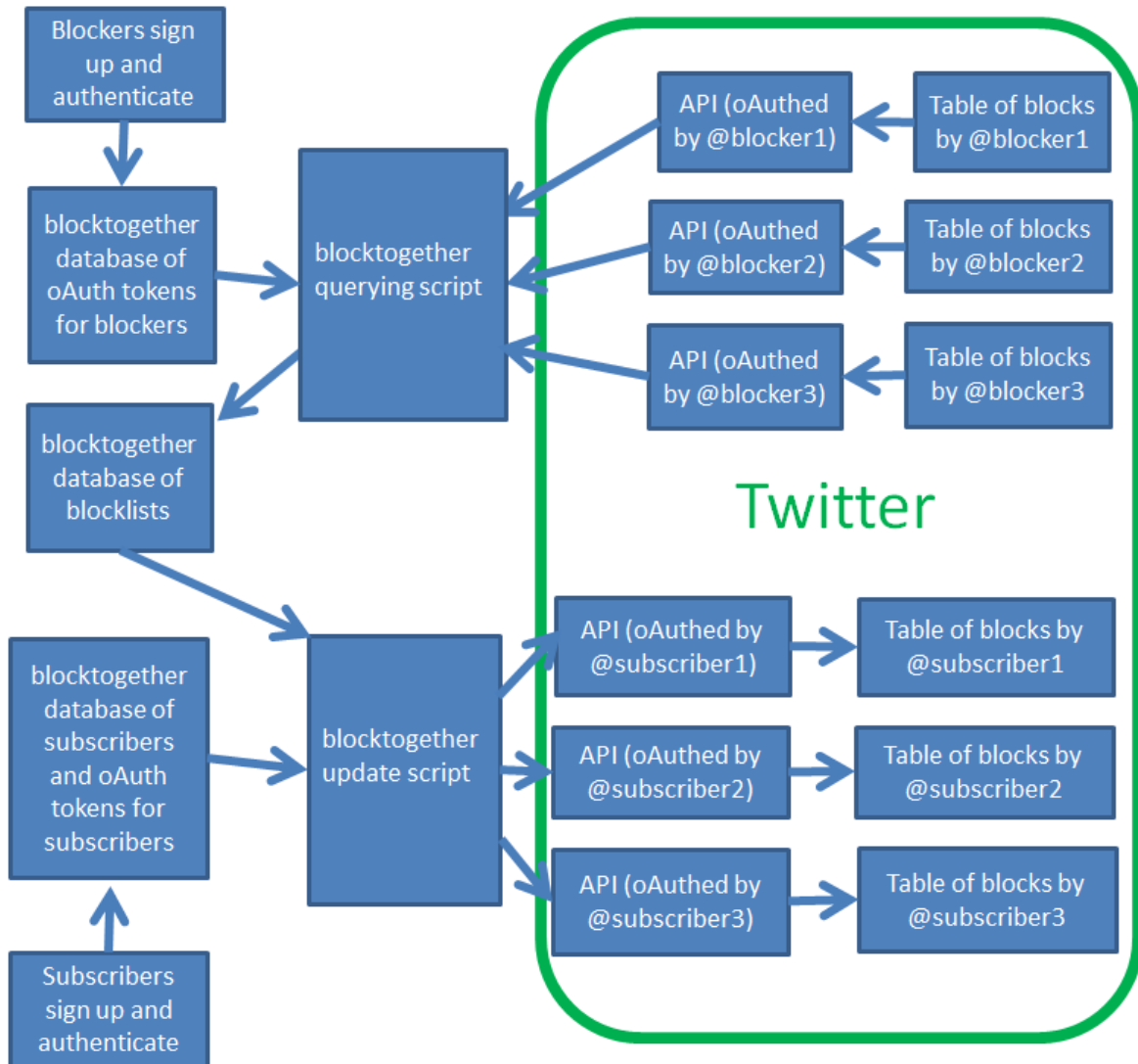


Figure 36: Diagram of blocktogether.org standard use cases

## 2.5: Deployment

By now, you've authenticated with [blocktogether.org](http://blocktogether.org), subscribed to your friend's blocklist, and had an experience of using Twitter where anyone who your friend blocks is automatically no longer visible to you. However, what if you find that your friend doesn't exactly have the same approach to blocking as you? Perhaps they block a lot of people you wouldn't block or they don't block a lot of people who you would block. You unsubscribe from your friend's blocklist and search for a few others, but find nothing that fits for you. You've been thinking about blocking on Twitter for some time now, and you've got some ideas about who you think deserves to be on a blocklist and who doesn't. You write some thoughts about blocklists up and publish them to your blog, which gets a few positive comments from friends and strangers. You opt to share your own blocklist on [blocktogether.org](http://blocktogether.org) and share the link broadly, posting it on Twitter, your blog, and some forums, message boards, mailing lists, and subreddits you frequent where you think people have the same kinds of ideas about these things as you do. Before you know it, your list takes off. You were right that there wasn't anything else exactly like what you wanted, and it turns out that lots of other people felt the same way you do about who they want to see and not see in their Twitter feeds. You name the blocklist something catchy, you create a website for it, and you even register a domain name to host it. When registering the domain name, you're asked to give your address and phone number for the ICANN registration, and you do, without much of a second thought.

It turns out that people have very strong feelings about the blocklist you created – especially the people who you've added to this list. You open up Twitter one morning and find hundreds of notifications, split pretty evenly between people who call you a genius and people who call you a Nazi. You add a few dozen more names to your blocklist. When you were first creating your blocklist, it turns out that one of the people you've blocked is a celebrity – somebody who you think is completely worthy of being blocked, but somebody who also has hundreds of thousands of followers on Twitter who would support him (and it is a he) in the end. He is incredibly offended that you have done something to make his opinions invisible for the thousands of people who have now subscribed to your blocklist. This celebrity also lives in a country that has stronger libel and defamation laws than most countries, and he sends you an e-mail threatening to sue you if you don't take him off the list. Your reply is short and direct; it may or may not contain profanity.

You get a message from a friend who monitors some of the anonymous message boards where trolls hang out, who tells you that you've been "doxxed" – somebody has found out every detail they can about you and created a document containing your home address, your employer, your phone number, the names of your parents and their addresses, photos of you, accounts you have on various social networking sites, and more. You follow the link to that thread on that anonymous messageboard and see that people are gleefully talking about getting you "SWATed" – calling your local police department and reporting a violent crime-in-progress to your address, in hopes of trying to get a SWAT team to conduct a no-knock raid. You go to your local police department and report this. While it takes a while to get to someone who can actually help, you are eventually put in contact with an officer who helps you file the right kind of police report. If your police department has a good procedure for dealing with these kinds of issues, your report will be incorporated into their response systems, such that if someone does

call in a crime-in-progress to your address, your report will be taken into account. This makes it less likely for the police to conduct a no-knock raid if someone calls in such a false report.

## **2.6: single-purpose accounts**

At this point, you might decide that you need some help in curating the blocklist. Figuring out which accounts to block is a lot of work, especially after the blocklist became more visible. You have had many like-minded people ask you to add accounts to your blocklist – while some people don't align with your views as well, there might be some people who you feel you can trust with the authority to add or remove accounts from the blocklist. How do you give them such access? In the current system you have set up via [blocktogether.org](http://blocktogether.org) at this point, you are the only one able to curate your own account's blocklist, which is then synchronized with all your subscribers via [blocktogether.org](http://blocktogether.org). In order to block accounts that another person has identified as blockworthy, they must send you a list of accounts to be blocked (typically in the form of links to their profiles) over a medium like e-mail or Twitter's direct messages. You visit those links which take you to Twitter's web interface, click on the "block" button, confirm the block, and then move to the next link.

While [blocktogether.org](http://blocktogether.org) supports collective blocklists, it does not support collaboration on blocklists, such that people can collectively discuss who ought to be added or removed – at least in the core functionality of the site. Yet one of the fascinating consequences of Twitter's lack of a 'one person, one account' policy (as in Facebook and some other social networking sites) is that you can register as many user accounts as you like, and you can share these among multiple people. Corporate and institutional accounts are the best example of this, but people share access to accounts for a number of other purposes too. The relative lack of restrictions on creating new Twitter accounts means that you can create a 'single-purpose account', which you can use to extend the practice of blocking even further than you can with the standard use case of [blocktogether.org](http://blocktogether.org).

In the standard use case for [blocktogether.org](http://blocktogether.org), you have your own personal account on Twitter which is what you use as a primary social networking site account – you follow interesting accounts, you post updates in the form of tweets, you block users who you think are worthy of being blocked, and so on. In this case, [blocktogether.org](http://blocktogether.org) synchronizes your personal account's blocklist with anyone who subscribes to it. However, if you want to share the responsibility of curating the blocklist (which has now spread across Twitter and has thousands of subscribers), the current infrastructure you have in place only lets this happen if you give someone else access to your Twitter account, so they can add or remove people to your blocklist in the same way you do. However, sharing passwords or login credentials to your personal Twitter account is a pretty bad idea, even if you trust them implicitly and even if your blocklist hadn't made you a target of the people who you have blocked.

To solve this, you create a new Twitter account, which is not intended for anyone to use in the way most people use Twitter accounts. Rather, this will be one that you share with people who you feel you can trust. Anyone with the password to the account will have the ability to block or unblock any account. Instead of automatically uploading your personal blocklist to [blocktogether.org](http://blocktogether.org), you have this shared single-purpose account be the source of the blocklist.



In the simplest version, you don't even need to write or run a single line of bespoke code yourself, as blocktogether.org's servers do all of the work involved in this process. You and your trusted group of blockers can log in to this account in a standard web or mobile Twitter interfaces (or any third-party clients like Hootsuite) and block or unblock accounts in the same way that personal blocks operate. Whenever blocktogether.org's servers routinely query the single purpose account's blocklist, the current version of the blocklist that everyone with access to the account has collectively curated will be synchronized with all the blocklist's subscribers. There is a problem in shifting from using your account to the single-purpose account shared among your fellow group of blockers. In blocktogether.org, blocklists are linked to specific accounts, and there are now thousands of people who have subscribed to your original hand-curated blocklist linked to your personal Twitter account. There is no way to automatically have everyone who subscribes to your personal account's blocklist switch to subscribe to the new blocklist of the single-purpose account in blocktogether.org, as subscriptions are tied to specific accounts. You will have to broadcast this change widely, telling your subscribers to subscribe to the new blocklist (and there is no way for you as a blocklist maintainer in blocktogether.org to get a list of all of your subscribers, for privacy protection).

## **2.7: encapsulation of the organization**

At this point, you have a new single-purpose Twitter account, with the log-in details shared among a trusted group of fellow blockers. The list of accounts blocked by this single-purpose account is stored on a server at Twitter, Inc., and a software agent at blocktogether.org routinely logs in as your single-purpose account and downloads the account's blocklist. Any member of your group can add someone to the blocklist of the thousands of subscribers by opening up the Twitter app or website, navigating to a user's profile, and clicking "block." Anyone with access to the account also has the ability to unblock any account that has been blocked. From the perspective of blocktogether.org's software agents (which run on servers that are independent from Twitter, Inc.), it is unable to tell that your original blocklist was one you curated yourself, while the second one is one you curate with a group. All that is visible from blocktogether.org's point of view is that there is a Twitter account which has blocked a set of accounts. Those with access to Twitter's servers almost certainly have the capability to internally collect and store records associated with each blocking or unblocking of a user, such as the IP address of the device used to take such action, meaning that it is quite possible that if someone at Twitter wanted to find out if an account was being used as a single-purpose account shared among multiple people, they could. However, even if Twitter did internally keep these records, blocktogether.org interacts with Twitter's servers through the API in such a way that when it comes to blocks, it must treat the account as a black box.

This collaboratively-curated blocklist, which is made possible through the shared single-purpose account, is quite limited in functionality at this point. Authorized blockers log in and block or unblock accounts whenever they see fit, and with what we have right now, there is no way to coordinate this work. There are no comment threads, voting mechanisms, or editable documents present in this assemblage of software and systems. However, your collaborative blocklist could be maintained using a more sophisticated platform, all while still relying on blocktogether.org's bot-based service. It only means you have to do this work in yet another system, which is where the encapsulation supported through single-purpose accounts

comes in. Such a model means that you can use the collaboration platform of your choice – a wiki page, a Google Docs spreadsheet, or a custom-built platform. All that is required is that there be some consistent way to curate a structured list of accounts to be blocked, which can be queried by an automated software agent. The agent will serve as the single authorized intermediary between your new collaboration platform and the single-purpose Twitter account that is serving as the source list for blocktogether.org. Unlike the previous version, a major strength of this approach is that you don't have to give out login credentials to the single-purpose Twitter account. You instead give out login credentials to your wiki, spreadsheet, voting system, or other custom-built collaboration platform. Your group curates a blocklist on that third-party site, and your automated software agent regularly logs into the single-purpose Twitter account and updates the account's blocklist, bringing it in line with the current state of the off-site blocklist. Then, when blocktogether.org's servers query the single-purpose Twitter account's blocklist, it will effectively receive the blocklist that you curated off-site. (For those of you wondering why you have to go through a single-purpose account, this is because blocktogether.org's software agent is only programmed to pull from Twitter account blocklists, not from any other source. This is potentially a big area for the software to be expanded, and the software is open source.)

### **2.7.1 Using a wiki**

For example, you could set up your own private wiki using MediaWiki, and have a page that lists all the accounts that are to be blocked, one per line. In this approach to collaborative blocklist curation, the list of accounts on the wiki page is the source list for blocktogether.org, which uses the single-purpose Twitter account as an intermediate storage system. With MediaWiki's permissions system, you could make this wiki page this private, so that only people who have an authorized account can view the list or edit it. You could also make the list public to view, but only let authorized accounts edit the list. Or you could open it up to editing from anyone. In any of these cases, you would have to delegate the work of updating the blocklist. You could have a designated member of your group regularly monitor the list for changes and use the web or mobile interface to bring the single purpose account's blocklist in line with the list on the wiki. Or you could write a bot to do this, which would be far more efficient. You need to make sure that new accounts added to the list on the wiki are blocked, as well as unblocking accounts that are removed from the list on the wiki. The following code implements such a bot:

```

import mwclient, Twython

# Connect to Twitter and log in as the single-purpose account
Twitter_blocker = Twython(/* LOGIN DETAILS for @ourblockbot */)

# Connect to our private wiki and log in with the wiki account that has access
ourwiki = mwclient.Site("https://ourblockbotwiki.org/w/api.php")
ourwiki.login("username","password")

# Get the list of accounts on the wiki page
wiki_blocklist = ourwiki.Pages["Blocklist"].get_expanded().splitlines()

# Get the list of accounts currently blocked by the single-purpose account
twitter_blocklist = Twitter_blocker.list_blocks()

# List all accounts currently blocked on Twitter, but not on the wiki blocklist
to_unblock = set(twitter_blocklist) - set(wiki_blocklist)

# List all accounts on the wiki blocklist, but not blocked on Twitter
to_block = set(wiki_blocklist) - set(twitter_blocklist)

#iterate through the list of accounts to be unblocked, one per row
for unblocked_account in to_unblock:

    # And logged into @ourblockbot's account, tell Twitter to unblock them
    Twitter_blocker.destroy_block(unblocked_account=blocking_id)

#iterate through the list of accounts, one per row
for blocked_account in to_block:

    # And logged into @ourblockbot's account, tell Twitter to block them
    Twitter_blocker.create_block(blocked_account=blocking_id)

```

### 2.7.2 Using a subreddit

With this kind of encapsulation, blocktogether.org can be used to support an automatically-updating blocklist that is generated from any given socio-technical system, so long as an automated software agent can generate a list of usernames that are to be blocked and update a single-purpose account. For example, you could even use the discussion site reddit to vote on what a blocklist should be. Reddit is a site that supports relatively independent “subreddits” on tens of thousands of different topics. Subreddits can be either public (where any registered user can view, post, comment, and upvote/downvote) or be made private (accessible only to invited accounts). People submit posts to a specific subreddit that contains a title, a description, and optionally a link. For every post, a threaded discussion is created about that post, where people can comment. Both posts and comments in posts can then be upvoted or downvoted by other people to increase or decrease their visibility in subreddits and threads respectively. Each user account can only upvote or downvote a post or comment once, although they can change their vote at any time. By default on the site’s web or mobile interface, a subreddit displays the 25 “hottest” posts using a formula based on the ratio of upvotes to downvotes and the amount of time since the post was made (newer content is preferred over

older content, and votes closer to the time of submission are weighted more). Subreddits can also be sorted by the raw difference of upvotes to downvotes received over a given interval of time, including all time.

Given reddit's existing affordances and the fully-featured API that is just as richly supported through software libraries as APIs for Twitter and MediaWiki are, you could create a private or public subreddit for curating a blocklist. The easiest way would be for people to submit posts that have no link (called "self" posts), with the title of the post being nothing but the username of the Twitter account to be blocked. Submitters could include descriptions about why they thought the account needed to be blocked (which can include links) and the commenting system could be used to discuss the case. The question then is how you would want to use a subreddit to generate a list of accounts to be blocked, which would be synchronized to the blocklist's subscribers via a single-purpose account. In one case, you could have the blocklist contain the 25 top user accounts of all time (and only those accounts), with a bot regularly querying reddit's API to select the posts which have the highest number of upvotes minus the number of downvotes. Then it would unblock accounts that have fallen out of the top 25 and block accounts that have entered the top 25.

```
# Import libraries for connecting to reddit and Twitter APIs
import praw, Twython

# Connect to Twitter and log in as the single-purpose account
Twitter_blocker = Twython(/* LOGIN DETAILS for @ourblockbot */)

# Connect to reddit and identify with a user agent
reddit = praw.Reddit(user_agent='Blockbot query bot 0.1')

# Create an empty list for the reddit blocklist
reddit_blocklist = []

# Get the 500 newest posts in the redditblockbot subreddit
for post in reddit.get_subreddit('redditblockbot').get_new(limit=500):

    # If the post has received more than 2 upvotes
    if post.ups > 2:

        # Add the title of each post to the list for the reddit blocklist
        reddit_blocklist.append(post.title.encode('utf-8'))

# Get the list of accounts currently blocked by the single-purpose account
twitter_blocklist = Twitter_blocker.list_blocks()

# List all accounts on the reddit blocklist, but not blocked on Twitter
to_block = set(reddit_blocklist) - set(twitter_blocklist)

#iterate through the list of accounts, one per row
for blocked_account in to_block:

    # And logged into @ourblockbot's account, tell Twitter to block them
    Twitter_blocker.create_block(blocked_account=blocking_id)
```

Alternatively, a subreddit could be used to support a quite different way of curating a blocklist, such as a process in which people submit posts containing the usernames of those they wish to add to the blocklist as a kind of nomination, and then any post that receives more than a certain number of upvotes from others would be added to the blocklist. The bot would operate in such a way that even if someone with access to the single purpose account's login credentials unblocked an account manually in the web or mobile interface, the next time this script ran, the bot would see that the post naming that account had more than 2 upvotes and would block it again.

```
# Import libraries for connecting to reddit and Twitter APIs
import praw, Twython

# Connect to Twitter and log in as the single-purpose account
Twitter_blocker = Twython(/* LOGIN DETAILS for @ourblockbot */)

# Connect to reddit and identify with a user agent
reddit = praw.Reddit(user_agent='Blockbot query bot 0.1')

# Create an empty list for the reddit blocklist
reddit_blocklist = []

# Get the top 25 posts of all time in the redditblockbot subreddit,
# according to the number of upvotes minus downvotes
for post in reddit.get_subreddit('redditblockbot').get_top_from_all(limit=25):

    # Add the title of each post to the list for the reddit blocklist
    reddit_blocklist.append(post.title.encode('utf-8'))

# Get the list of accounts currently blocked by the single-purpose account
twitter_blocklist = Twitter_blocker.list_blocks()

# List all accounts currently blocked on Twitter, but not on the wiki blocklist
to_unblock = set(twitter_blocklist) - set(reddit_blocklist)

# List all accounts on the wiki blocklist, but not blocked on Twitter
to_block = set(reddit_blocklist) - set(twitter_blocklist)

#iterate through the list of accounts to be unblocked, one per row
for unblocked_account in to_unblock:

    # And logged into @ourblockbot's account, tell Twitter to unblock them
    Twitter_blocker.destroy_block(unblocked_account=blocking_id)

#iterate through the list of accounts, one per row
for blocked_account in to_block:

    # And logged into @ourblockbot's account, tell Twitter to block them
    Twitter_blocker.create_block(blocked_account=blocking_id)
```

This encapsulation means that the key aspects structuring the process of curating a blocklist behind your blockbot can change quite dramatically, while core parts of the infrastructure supporting it remains the same. You could switch between multiple versions of the reddit-based blockbot system without needing to reconfigure anything besides this particular script that updates the single-purpose account's blocklist – or even switch between

one curated on a wiki and one curated on reddit. The configuration at blocktogether.org remains the same, and subscribers do not have to re-authorize or re-subscribe. However, switching between these different software configurations is the easy part – the hard part is deciding which kind of configuration is best for curating a shared blocklist.

### **3. Conclusion**

If you are going to develop a bot-based collective blocklist, you ultimately have to do far more work than just software development – or even systems administration. You have to make a decision about what kind of sociotechnical system you want to have for the group of people who will be curating a shared list of blockworthy accounts. You have to think about what it means for your group to collectively curate a blocklist and what kinds of values you want to uphold in a bespoke system. You probably have to talk to the people who you’ll be relying on to help you curate this blocklist, asking them for ideas and sharing prototypes of your ideas. You won’t get it right the first time – or the second, or the third, or the twentieth. Deciding what kind of system you want to support the work of curating a blocklist is going to be an ongoing process, one that will change as you and your group come to better understand what it means to be engaged in this kind of project. As you start to do more work in this space, you’ll come to understand certain aspects better than you have before. Your group will also change – it might rapidly grow, it might merge with other like-minded groups, and it may come under serious attack from opposing groups. The way that you and your group engage in blocklist curation will change as you and your group change. Or at least, it better change – if it doesn’t, then something is probably wrong.

If you want more help in thinking about these decisions, read the next chapter. In chapter 8, I give specific historical accounts of how two different blockbots have been developed and redeveloped over time. Both of the projects I studied began as individual projects by a single developer, but expanded well beyond that, as the lead developers and those who supported their work came to new understandings about what they wanted bot-based collective blocklists to be. They faced challenges and criticisms, both from within the group and from outside of the group, which spurred debates about what it meant to curate a collective blocklist. Sometimes code was rewritten, sometimes new norms were established, and sometimes they struggled hard to keep everything the same. But in all, in coming to these understandings about how to operate blockbots, they also came to understandings about what they wanted Twitter as a social networking site to be.

## Chapter 8: Blockbots as projects of collective sensemaking and reconfiguration

### 1. Introduction

The previous chapter gave a detailed description of the kinds of work that are involved in designing, developing, and deploying a bot-based collective blocklist. The tutorial-ethnography ended with an open question: once you have a way for people to subscribe to a blocklist, what kind of sociotechnical system ought you build to collectively curate that list of blockworthy accounts? The modularity of the blockbot model can be confusing and even a bit Rube Goldberg-esque, but the benefit is that it supports both multiple blocklist-curating organizations (who might share quite different ideas about what constitutes blockworthiness) and multiple ways of doing the work of curating a blocklist. Even people who share the same ideas about what constitutes blockworthiness might have different ideas about how they want to go about curating a list. Some people prefer to err on the side of false negatives, while others prefer to err on the side of false positives. Some people are willing to put in substantially more work to maintain a higher quality blocklist, while others prefer a process that may be lower quality, but takes less effort to curate. Some believe more in a consensus-based model, while others favor voting. Some believe in having firm standards and established policies, while others prefer more open-ended processes. Some believe in a more egalitarian model of decision-making, while others prefer to have leaders with established roles. In all, there is no one way of curating a collective blocklist, nor should there be.

Like in Wikipedia, one of the defining features of blockbots is that they are delegated articulation work in a broader task. Bots are performing work that keeps members of a team organized and on-track, aggregating and consolidating their actions in a way that helps produce a common product.

#### 1.1 Classification and its consequences: beyond “What *really* is harassment?”

Blockbots are information infrastructures that are bound up in a problem of classification, a subject extensively studied in informatics and science and technology studies (Bowker et al., 2010; Bowker & Star, 1999; Star, 1999). The question “what *really* is harassment?” has no single universal answer, just like questions about identifying hate speech, spam, or pornography. Like all questions involving classification, the question about what is harassment is answered in reference to some external set of norms, practices, standards, and infrastructures that define harassment and delimit it from non-harassment. There are multiple ways of making such judgments, and many of the ways that governments, societies, organizations, and individuals have for making this classification produce contradictory outcomes and use many different concepts and frameworks (Marwick & Miller, 2014). For example, legal definitions of harassment in various jurisdictions can be quite different from definitions of harassment instituted in privately-owned spaces within that jurisdiction. In other words, an action can be deemed to not violate a law against harassment in a court of law, but can still be determined to violate a norm or policy against harassment in a workplace, social club, place of worship, online forum, professional association, social networking site, or a group of friends. Even isolating harassment solely as a question of law in the United States differs depending on the jurisdiction. A particular action can be legally determined to be harassment

in a court of law in the same jurisdiction according to workplace safety and equal employment statutes, but not determined to be harassment according to criminal statutes.

As Bowker and Star (1999) emphasize in *Sorting Things Out: Classification and Its Consequences*, these kinds of differences in classification systems are not just based on the different formal definitions for a term like harassment; their differences extend to the way in which a socio-technical system for classifying harassment evaluates cases according to a definition. A court of law, a workplace, and a group of friends can have the exact same definition of harassment, but have radically different ways of deciding whether a case fits that definition. Such processes show how classification systems “grow out of and are maintained by social institutions” (p. 61). They argue in favor of a highly-situated approach to studying how these definitions are put into practice, advocating:

fine-grained analyses of the nature of information infrastructures such as classification systems ... to demonstrate how they simultaneously represent the world “out there,” the organizational context of their application ... and the political and social roots of that context (61).

In their analysis of several classification systems – including those for classifying diseases, race, and types of work – Bowker and Star argue that classification systems are ways in which people work out different (and sometimes incompatible) understandings of what they think the world is and ought to be. People develop classification systems in the course of developing such broader understandings – and vice-versa. They call the mutual co-constitution of information infrastructures and social worlds “convergence,” arguing that these two domains are inextricably linked:

Convergence ... is the double process by which information artifacts and social worlds are fitted to each other and come together ... On the one hand, a given information artifact (a classification system, a database, an interface, and so forth) is partially constitutive of some social world. ... On the other hand, any given social world itself generates many interlinked information artifacts. ... information artifacts undergird social worlds and social worlds undergird these same information resources. (82)

## **1.2 Blockbots are embedded in and emerge from counterpublic communities**

As an algorithmic system for supporting the work of classifying harassment, a blockbot’s software code implements a particular understanding of what harassment is and how it ought to be identified. Blockbots are compelling cases for showing how algorithmically-supported classification systems are situated within particular contexts, extending far beyond their source code. I have seen how such computational systems be continually re-developed as people come to understand what it even means to ask and answer questions like “Who is and is not a harasser?” and “What ought to be done about harassment?” The answers to such questions do not simply require building the right technical infrastructure or even the right socio-technical system. Such answers are situated in and emerge from the lived experiences of many different kinds of people, which is why the centuries-old thorny question of how to operate a universal public sphere is so complicated (Fraser, 1990; Habermas, 1989). In both my archival research



and interviews, I initially began focusing on this specific kind of automated software agent, but was continually drawn to the broader projects, communities, and institutions in which those blockbots had meaning and significance. The overwhelming majority of blockbots I encountered were not one-off software development products, but were instead developed out of or into broader projects seeking to formulate responses to online and/or gender-based harassment. This context is important in understanding how different bot-assisted projects built around curating a collective blocklist operate as counterpublic modes of filtering and gatekeeping – which is not how most bots in Wikipedia operate, for reasons I will discuss later.

As many blockbots have grown their subscriber base beyond an initial tight-knit core, I have found that their developers and authorized curators (sometimes called ‘blockers’) revise the code and procedures for curating a bot-supported shared blocklist. These revisions are compelling cases of socio-technical reconfigurations (Suchman, 2007), as they simultaneously involve changes in more abstract, normative understandings about harassment as well as concrete alterations to a blockbot’s source code. Specific reconfigurations I have observed include: creating an appeals board with a formalized process to review accounts that were allegedly wrongly added to a blocklist; providing support for blockers to document why they added an account to a blocklist; requiring that a second authorized blocker review and approve a new addition to a blocklist, when previously, any authorized blocker could independently add an account to the blocklist; and splitting a single blocklist into a set of multiple lists, based on different understandings of what constituted blockworthiness. These modifications and extensions illustrate how the people who operate such blockbots are actively and continually reflecting on how to best design a social-computational system for moderating their own experiences of Twitter.

## **2. Literature review: harassment, technological determinism, and networked publics**

### **2.1 Harassment as a civil rights issue about participation in networked publics**

This chapter’s contributions are at the intersection of the online harassment literature and the literature on the governance and moderation of networked publics. It is important to focus on harassment as an issue about participation in a public, which Citron does in her recent book *Hate Crimes in Cyberspace* (2014). Citron defines cyber harassment as “intentional infliction of substantial emotional distress accomplished by online speech that is persistent enough to amount to a ‘course of conduct’ rather than an isolated incident,” which is typically carried out through “threats of violence, privacy invasions, reputation-harming lies, calls for strangers to physically harm victims, and technological attacks.” (p. 3). Citron’s most powerful argument is in her responses to claims of “free speech,” in which she argues that harassment is a civil rights issue, given how harassment works to inhibit participation. This extends a long line of scholarship and public policy that distinguishes negative and positive freedoms (Berlin, 1969). Instead of just seeking to minimize restrictions on speech by an established authority, this view also prioritizes the ability for all people to freely express themselves in public without fear or coercion, regardless of the source. This means it is crucial to focus on the chilling effects in which people are silenced and excluded from public discourse by other private individuals, whose acts of harassment are often seen as protected in more traditional frameworks. If

harassment is considered protected under principles of free speech, then this trades off with the free speech rights of those targeted, and Citron reviews many cases and studies about how harassment works to silence targets into withdrawing from public spaces – in fact, this is frequently the goal of harassers. Such a situation is not unique to online media. Nancy Fraser extensively discusses how this played out in the coffeehouses and other public forums of early-modern Europe in her critiques of Habermas’s influential account of the public sphere (Fraser, 1990).

## **2.2 The Californian Ideology, individual/collective ethics, and technological solutionism**

The current state of harassment online is in part a product of a particular technologically determinist, libertarian mindset that is prominent in Silicon Valley, which Barbrook and Cameron identified as “the Californian Ideology” (1996). The modern Internet as we know it is the product of both countercultural and libertarian activists, who were both concerned with censorship from traditional governments and corporations (Turner, 2006). Activists like John Perry Barlow of the Electronic Frontier Foundation wrote tracts like “A Declaration of the Independence of Cyberspace” (1996), celebrating the technological principles that made online interactions “immune” to more traditional forms of regulation. Texts expressing this ideology even frequently included pro-inclusion and diversity statements, imagining that the Internet’s inherent resistance to state censorship would lead to “a world where anyone, anywhere may express his or her beliefs, no matter how singular, without fear of being coerced into silence or conformity,” as Barlow wrote. Yet this understanding of a mediated public is based on an ideal of the autonomous individual, whose classical liberal rights to freedom of expression are to be supported by technology (rights that are allegedly under attack from more traditional authorities, as Barlow and others claim). As Adam critiques in her work on computer ethics (Adam, 2005), when ethics does come on the scene among technologists, it often is discussed in a way that assumes “individual, rationalistic, rule-based ethical models” (p. 38) like utilitarianism, which align with technologically determinist principles. In contrast, Adam argues for feminist ethics of collectivity and care that focus on the structural inequalities of marginalized groups. She argues that “despite holding a rhetoric of equality and participation,” the standard utilitarian, individualistic, allegedly meritocratic ethics common among technologists “often make no challenge to the structures that are causing that inequality in the first place” (Adam, 2000, p. 2).

Harassment, abuse, bullying, and incivility are longstanding problems in any communication medium and public space, and it is no surprise that these issues have played out online almost as soon as the Internet entered the households, workplaces, and dorms. Nakamura, writing against the utopian imaginations of those like Barlow, argued that existing social inequalities and hierarchies have been reinforced online, rather than diminished (Nakamura, 2002). Prioritizing the expressive rights of individuals against “censorship” is problematic, because the owner/operators of platforms can be unwilling to take actions against harassers. This issue also intersects with another mindset prevalent in Silicon Valley, which sees these problems as technological ones requiring technological solutions – contrary to scholars of online harassment like Citron who argue that harassment is ultimately a social, cultural, and legal problem. Evgeny Morozov has recently termed this mindset “technological

solutionism” (Morozov, 2013), and the search for technical solutions to societal problems is also an aspect of the Californian ideology Barbrook & Cameron identified. Barbrook & Cameron, Morozov, and Adam – writing at different times about the same dominant mindset – concur that one of the core problems with such solutions is that they are often based in an autonomous individualist mindset. Technology is often used to shift the burden of solving these problems to the individual, frequently assuming that having such a responsibility is empowering. In the issue of harassment, this can be seen in the rise of muting, blocking, flagging, and reporting features, which Crawford and Gillespie critique on these grounds. They advocate “a more social mechanism for a more social problem,” looking to the open backstage model of Wikipedia, where people discuss and debate cases in public. Ultimately, they conclude that individualistic mechanisms like “flags may be structurally insufficient to serve the platforms’ obligations to public discourse, failing to contend with the deeper tensions at work.”

One kind of response that takes a more communitarian, discursive, and civic-focused approach to platform governance is the recent partnership between Twitter, Inc. and the non-profit Women, Action, and the Media (or WAM), analyzed and described by (Matias et al., 2015). In that three-week pilot collaboration, WAM was granted an “authorized reporter” status, where Twitter users could report harassment to WAM, instead of the default flagging mechanism that sends reports to Twitter, Inc.’s internal processes. WAM volunteers were an intermediary between reporters of harassment and Twitter, Inc.’s somewhat black-boxed team of humans and algorithms that enforce the site’s rules. WAM volunteers would review and discuss these reports, sometimes working with the reporter to further specify or document harassment in cases of incomplete or ambiguous reports. If volunteers decided that a report was appropriate to escalate to Twitter, Inc. staff, they would submit it to a specialized ticketing system. The WAM team would interact with Twitter, Inc. staff as necessary on the reporter’s behalf, answering questions and advocating if necessary. In the three-week pilot period, WAM received 811 reports and decided to escalate 161 of them. Of those 161 reports, Twitter, Inc. took action on 55%, suspending 70 accounts, issuing 18 warnings, and deleting one account. WAM claimed that the project was a success in piloting an alternative to the often opaque processes and policies around harassment in major web platforms. They argued that it was important to build a more communal way of responding to harassment, finding that many targets of harassment needed different kinds of support in making sense of harassment, particularly given how harassers can use sophisticated techniques to overwhelm their targets and mask their own identity. WAM also noted that only 43% of reports came directly from those targeted by harassment, with the majority coming from a target’s authorized delegate or a bystander who observed the harassment.

### **2.3 Theorizing publics and counterpublics**

WAM’s partnership with Twitter raises the issue about how non-dominant groups assemble in response to hegemonic venues, where they are often excluded from fully participating. In the next section, I turn to Fraser’s feminist critique of Habermas’s account of the public sphere, in which she theorizes how non-dominant groups form “counterpublics.” This theoretical foundation is crucial to understanding the implications of blockbots for Twitter as a networked public, which I discuss with specific cases in sections 3 and 4.

Scholars and commentators have long been concerned with the prospect of the Internet as a public sphere, drawing from Jurgen Habermas's influential writings on the emergence of public forums and media in early-modern Europe. Nancy Fraser's concept of counterpublics provides a compelling framework for theorizing blockbots in a privately-owned public space like Twitter. Fraser critiqued Habermas's influential account of the bourgeois public sphere, which celebrated the coffeehouses, newspapers, and other forums where 'members of the public' could rationally debate socio-political issues and build consensus. Habermas's ideal concept of the public sphere imagines spaces where anyone can enter, bracketing their own social status to engage with others as equal peers in deliberation. Fraser notes that these were highly exclusionary spaces, particularly for women, persons of color, and members of the working classes. Yet even when marginalized individuals were not officially restricted from participating in these spaces, the fiction that the space was a neutral one and that all participants were equal often served to make subtler forms of exclusion less visible: "such bracketing usually works to the advantage of dominant groups in society ... the result is the development of powerful informal pressures that marginalize the contributions of members of subordinated groups." (Fraser, 1990, p. 64) Against the idea of the universal public sphere where as a neutral space where all are ostensibly equal, Fraser contrasts the "counterpublics" in early-modern Europe that existed alongside the spaces typically reserved for wealthy white men. Such counterpublics "contested the exclusionary norms of the bourgeois public, elaborating alternative styles of political behaviour and alternative norms of public speech."

Crucially, Fraser's central critique of Habermas is **not** the frequently made argument that the coffeehouses and newspapers of the bourgeois public sphere are illegitimate if they are not accessible for all to participate (which Habermas also strongly advocates). In addition, Fraser critiques the hegemonic way in which certain public venues for socio-political debate and discussion came to be known as "the public," while other kinds of venues and activities that were populated by women, minorities, and working class individuals were excluded from this concept of the public. Subordinated groups had to enter hostile spaces in order to have their discourse be considered part of the public, and Fraser extensively reviews the literature about how dominant groups engage in various practices to silence, intimidate, and chill participation by non-dominant groups. Fraser's feminist critique recasts the bourgeois public sphere as but one of many public spheres – albeit one that was exclusively assumed to represent the population as a whole, particularly to the state, gaining a kind of cultural hegemony (Gramsci, 1995).

Traditionally, counterpublics have been understood as "parallel discursive arenas" (Fraser, 1990, p. 67), separate spaces where members of a subordinated group are able to participate in their own kind of collective sensemaking, opinion formation, and consensus building. Fraser references the extensive set of media in late 20<sup>th</sup> century U.S. feminism as an example of a vibrant counterpublic, with independent publishers, bookstores, conferences, festivals, advocacy organizations, and other spaces for face-to-face and mediated interaction that ran parallel to more dominant equivalent institutions. As counterpublics are characterized by their lack of a hegemonic claim to represent the population, members must employ alternative tactics to make their concerns and activities visible to 'the public,' while maintaining

a safe space to discuss and understand issues relevant to them. With the rise of computer-mediated communication in the 1990s, many scholars discussed the potential for digitally-mediated environments to be counterpublics (Fernback, 1997; Papacharissi, 2002; Poster, 2001). In self-organized spaces, marginalized groups could assemble free from the modes of domination that existed in ostensibly ‘neutral’ spaces. Members of counterpublic spaces could potentially discuss and debate issues according to their own discursive norms, come to common understandings about issues, then engage with more hegemonic media that claims to represent ‘the public.’ However, as Fernback and Papacharissi note, there are many ways in which such separate online spaces can become disrupted, derailed, and delegitimized by more dominant and hegemonic forces, just as in the counterpublics of early modern Europe.

As blockbots effectively create multiple versions of the same centrally-hosted social networking site, they are a different version of Fraser’s “parallel discursive arenas” than the separate digitally-mediated environments that are to run alongside more dominant and hegemonic spaces. Rather than creating a separate, alternative discursive space, blockbots are a way in which counterpublic groups exercise agency over their own experiences within a hegemonic discursive environment. In the rest of this article, I discuss two different but linked aspects about how blockbots support counterpublic groups. In section 3, I focus on how this form of moderation has been a powerful way for targets of harassment to moderate their own experiences on the site. This technology does raise some political and ethical issues about algorithmic gatekeeping and “filter bubbles” (Pariser, 2012), which I argue must be understood in their counterpublic context. After discussing these issues around the impacts of blockbots in section 3, I move to issues around the internal design and development of such systems in section 4. I show how blockbots are not just important for their consequences; in addition, they are one of many ways in which counterpublic groups work to understand and articulate what kind of world they want to live in. To this end, I discuss several key moments of reconfiguration in the history of two different blockbots, where counterpublic groups worked out the messy details about how exactly they ought to algorithmically support the classification of harassment.

### **3. How blockbots help counterpublic groups moderate their own experiences**

#### **3.1 The impacts and effects of blockbots**

In this section, I discuss the impacts and effects of blockbots, showing how they help counterpublic groups participate in a hegemonic networked public on their own terms. Blockbots were initially created around the broader issue of online harassment in Twitter, which has received substantially more attention in 2014 and 2015 due to a number of more and less organized harassment campaigns that unfolded on Twitter – including the sustained anti-feminist GamerGate movement and multiple shorter cases in which celebrities have been harassed (Chess & Shaw, 2015; Heron et al., 2014; Matias et al., 2015). I investigated the historical development of these algorithmic systems, and the earliest bot-based collective blocklists that operated in Twitter were created years earlier, but also were developed specifically and explicitly around a perceived governance gap about harassment. In 2012, some targets of harassment and their allies were arguing that Twitter, Inc. was not doing nearly enough to respond, particularly against those who were using the social networking site to

launch large-scale, coordinated campaigns against particular individuals. Such anti-harassment activism and organizing took place in smaller online communities that did not make as many headlines in mainstream or social media as they have in 2014 and 2015, particularly playing out in communities that had fractured on the issues of feminism and social justice.

Many people facing coordinated harassment campaigns for their feminist political stances called for changes to the site's policies, enforcement mechanisms, and user interface features, in order to minimize the impact of harassment and take action against identified harassers. In addition to this more discursive anti-harassment work petitioning Twitter, Inc., a few advocates and activists turned to software-based approaches to help their fellow community members moderate their own experiences on Twitter. Such software tools were based on automated software agents that extended the built-in affordances of Twitter by making creative use of features developed largely to support third-party clients. These blockbots made it possible for people to collectively curate blocklists of those they identified as harassers, then synchronize these blocklists with subscribers. With a few clicks, a Twitter user could subscribe to a blocklist and automatically no longer see any tweets or notifications from anyone added to that list. Some of these blocklists are curated by single individuals, others by tight-knit groups, and a final set are algorithmically generated based on various methods of data collection and analysis. In fact, the modularity of the blockbot approach means that a bot-based collective blocklist can theoretically be created and curated by any given socio-technical system (e.g. an open wiki in which anyone can edit or comment, a vote-based process restricted to a tight-knit group, a 'team of one' using Twitter's default blocking interface, an automated agent running predictive models, etc.), then the list can be automatically synchronized to subscribers using the same computational infrastructure.

Blockbots as a computational entity encapsulate an organization that collectively articulates a list of blockworthy accounts, meaning that blockbots are both technical and social entities. In fact, blockbots are most compelling in that they help bring together a group of people who oppose a particular understanding of harassment. The history of blockbots cannot just be told in terms of software development release cycles, as they are also about the formation of counter-harassment communities. This capacity for collective action in counter-harassment work is important given the disparities of scale that are associated with online harassment. As many scholars note, a particularly problematic form of harassment takes the form of 'piling on', where a large number of people each send a small number of messages, overwhelming the target. The work of harassment can be efficiently distributed and decentralized, with anonymous imageboards serving as one of many key sites for the selection of targets. Some prominent anti-feminist individuals also use Twitter itself to direct their tens of thousands of followers to particular accounts. In such a situation, it only takes a short amount of time and energy to send a single harassing reply. In contrast, the work of responding to harassment is much more difficult to scale, as each of those messages must be dealt with by the recipient. Targets of more-and-less coordinated harassment campaigns are at a distinct disadvantage even with Twitter's built-in blocking feature. With blockbots, counter-harassment work can be efficiently distributed and decentralized across a group with common understandings about what harassment is.

### 3.2 A hashtag hijacking: selectively tuning in to an affective public

One compelling example of the use of blockbots can be seen in a case of a ‘hashtag hijacking.’ A hashtag stream is a feature supported in Twitter’s interfaces which displays an aggregated set of all tweets containing a particular hashtag (i.e. #egypt, #YesAllWomen, #blacklivesmatter). Hashtags are frequently used to bring together people around a common event or issue. Bruns and Burgess discuss hashtags as “ad-hoc publics” (Bruns & Burgess, 2011) and Papacharissi discusses as “affective publics” (Papacharissi, 2014), speaking to the importance of these spaces for the formation and activation of publics. Many studies of hashtags celebrate the collective action and opinion formation through generally convergent hashtags (like those in the Arab Spring or the Occupy movement). However, hashtag streams are quite susceptible to being used for purposes that are not intended by those who initially created and used the hashtag. Hashtag hijacking can be a way for a counterpublic group to subvert a more dominant hashtag, but as Katy Pearce has found in her study of social media use by the ruling regime of Azerbaijan, it can also be a way in which domination of more subordinate groups is conducted (Pearce, 2014).

In one prominent case, the Twitter hashtag feed for a major open source software conference was inundated with tweets from a large number of Twitter accounts in a coordinated condemnation and defamation of one of the conference’s speakers: a woman speaking about gender-based harassment in open source communities, who had developed a blockbot to help respond to this very phenomenon. This blockbot developer had targeted by anti-feminist groups, and details about her upcoming talk at the conference had been posted to various forums where anti-feminist groups congregate, with a call to flood the hashtag stream. While some of the tweets were offensive and threatening enough to be officially removed from Twitter by staff for violating the site’s policies, many more were making disparaging remarks about the speaker that did not constitute abuse according to Twitter’s policies. Some conference attendees compared the flooding of the hashtag to a distributed denial of service (DDOS) attack, while others referred to it as spam. Attendees posted that the coordinated attack made the hashtag stream “unusable” even using Twitter’s built-in blocking features, leading many (including the speaker and a conference organizer) to advocate subscribing to the bot-based blocklist. Many of the accounts that were hijacking the hashtag had even already been added to the blocklist, as such anti-feminist groups have engaged in these kinds of efforts previously. One subscriber wrote a retrospective account of the hashtag hijacking, describing how the feed was unusable until they subscribed to the blocklist – expressing a bit of a surprise that “it works.” Those who subscribed to the blocklist discussed how they were subsequently able to use the hashtag to engage with each other about the conference on their own terms, generally resisting this hijacking attempt. The blockbot’s developer also stated that the hashtag hijacking provided an opportunity for more data collection about harassment, to better improve the blocklist.

In this way, blockbots are a novel way in which counterpublic groups are seeking to refine what Crawford calls “disciplines of listening,” in which she argues that practices of seeking and consuming content in networked publics are an “embedded part of networked engagement” (Crawford, 2009, p. 527). In this view, ‘lurking’ is as much a complex and multi-faceted mode of participation as submitting content, and blockbots help counterpublic groups participate in Twitter more on their own terms. Papacharissi discusses how selective

aggregation mechanisms in social media streams support “affective publics” in which people not only share information and opinion, but also form shared “structures of feeling”:

Publics assembled out of individuals feeling their way into a particular news stream generated via Twitter engage in practices of rebroadcasting, listening, remixing content, and creatively presenting their views—or fragments of their views—in ways that evolve beyond the conventional deliberative logic of a traditional public sphere. These practices permit people to tune into an issue or a particular problem of the times but also to affectively attune with it, that is, to develop a sense for their own place within this particular structure of feeling. (Papacharissi, 2014)

Blockbots emerged in response to a perceived governance gap, when like-minded people found that Twitter, Inc. was not removing messages and accounts that they considered to be harassment. Instead of setting their account to private (which is a common recommendation to targets of harassment), blockbots let counterpublic groups continue to participate in networked publics, but selectively tuned out of the kind of content that would otherwise potentially drive them away from the site. In this way, those who use blockbots do not have to affectively attune with harassers who hijack a hashtag – they can ignore them or take action against them by adding them to the blacklist.

There is one major caveat about the role of blockbots in hashtag streams: it isn’t always supported by Twitter’s infrastructure. It is the responsibility of Twitter clients (the programs or web pages that are used to access the platform) to retrieve tweets from the platform’s Application Programming Interface, and then filter and display those to the user. Twitter, Inc.’s clients do not always do this fully with search results, although the ‘timeline’ (or default view of Twitter based on tweets from accounts a user is following) has always been designed to filter out blocked accounts. Twitter also frequently changes various aspects of their user interfaces and policies, sometimes with little warning or notice. Currently, Twitter’s clients are not filtering blocked accounts from search results. These redesigns reshape the affordances of the site in powerful ways, which also changes how bespoke tools like blockbots operate.

Blockbots do let non-staff change the affordances of a privately owned and operated site like Twitter, but they also are constrained by the design of the systems they seek to change (and the tacit approval of Twitter, Inc. staff, which can block these bots from operating if they so choose). Yet even when Twitter’s clients are not programmed to filter out blocked accounts when viewing search results, they do still help people moderate their own experiences on Twitter by filtering out unsolicited notifications and making it more difficult for harassers to identify them. Furthermore, in line with my previous argument on blockbots as sites for counterpublic collective action and community formation, this hashtag hijacking helped grow this particular counter-harassment community. The blockbot developer who had been the key target of the hijacking stated that many new people subscribed to the blockbot, and that the harassers only worked to provide a “real-time demonstration” about why harassment is a serious issue and why tools like blockbots are important in responding to harassment.



#### **4. Blockbots are embedded in and emerge from counterpublic communities**

In this next section, I take up the computational aspect of blockbots, which has theoretical implications for this line of scholarship. I argue that as computational infrastructure for supporting the classification of harassment, blockbots are ongoing accomplishments of collective sensemaking, in which counterpublic groups work to enact ideas about what harassment is and how it ought to be dealt with. I discuss moments of reflection and reconfiguration about blockbots, which show how blockbots often involve a quite different approach than the technologically determinist ‘solutionist’ mindset.

##### **4.1 Blockbots as communities, not just technologies**

As a computational system for classifying harassment, a blockbot’s software code enacts a particular understanding of what harassment is and how it ought to be identified. Like all classification systems (and all technologies), they are not neutral, but instead reflect the ways that their designers understand the world (Bowker & Star, 1999). Blockbots are compelling cases for showing how algorithmically-supported classification systems are situated within particular contexts, which extend far beyond their source code. In studying the historical development of several different blockbots over time, I have seen how such systems are continually developed and redeveloped as people come to better understand what it even means to ask and answer questions like “Who is and is not a harasser?” and “What ought to be done about harassment?” The answers to such questions do not simply require building the right technical infrastructure – the ‘solutionist’ belief that there could be one universal system that would finally settle the issue about what is and is not appropriate content. Rather, as Bowker and Star note with classification systems about race, health, and labor, these systems are ongoing accomplishments in sensemaking. Answers to questions about what kind of behavior ought to be made visible in public spaces emerge out of the lived experiences of many different kinds of people and cannot be purely reduced to an information processing problem.

In both my archival research and interviews about blockbots, I initially began focusing on this specific kind of automated software agent, but I was continually drawn to the broader projects, communities, and institutions in which those blockbots had meaning and significance. The overwhelming majority of blockbots I encountered were not one-off software development products, but were instead developed out of or into broader projects seeking to formulate responses to online and/or gender-based harassment. This context is important in understanding how different bot-assisted projects built around curating a collective blocklist operate as counterpublic modes of filtering and gatekeeping. As many blockbots have grown their subscriber base beyond an initial tight-knit core, I have found that their developers and authorized curators (sometimes called ‘blockers’) revise the code and procedures for curating a bot-supported shared blocklist. These revisions are compelling cases of socio-technical reconfigurations (L. Suchman, 2007), as they simultaneously involve changes in more abstract, normative understandings about harassment as well as concrete alterations to a blockbot’s source code.

Specific reconfigurations I have observed include: creating an appeals board with a formalized process to review accounts that were allegedly wrongly added to a blocklist;

providing support for blockers to document why they added an account to a blocklist; requiring that a second authorized blocker review and approve a new addition to a blocklist, when previously, any authorized blocker could independently add an account to the blocklist; and splitting a single blocklist into a set of multiple lists, based on different understandings of what constituted blockworthiness. These modifications and extensions illustrate how the people who operate such blockbots are actively and continually reflecting on how to best design a social-computational system for moderating their own experiences of Twitter. Far from representing a ‘solutionist’ mindset that harassment is simply a technical problem to be solved with the right assemblage of algorithms, these cases show how the ostensibly technical work of software development can be a way in which counterpublic groups work out various ideas about what harassment is and what ought to be done about it. In the next section, I discuss a pivotal reconfiguration in the history of one blockbot, which illustrates several issues about how blockbot developers, blockers, and subscribers work to enact a particular shared understanding of how to respond to harassment.

#### **4.2 From moderating their own forum to moderating on Twitter**

One of the more long-running blockbots I studied was initially created by a lead developer who was a member of a small, tight-knit online community, and the software reflected existing practices and priorities around harassment that they had developed in their web-based forum. In previous years, this online community had become targeted by those who took issue with their advocacy around feminism and social justice. They had developed sophisticated policies and procedures on their own forum for identifying trolling, harassment, abuse, and a variety of other phenomena they sought to exclude from their own space. When members of this online community began to use Twitter, some of them received similar kinds of harassment from anti-feminists on that social networking site, including one substantial coordinated campaign against the community’s founder. Some of the other community members who were on Twitter used the built-in blocking feature to individually remove harassers from view, then began to share the usernames of those they had blocked. However, this sharing of blocked accounts was unwieldy and not easily supported by Twitter’s user interfaces, which inspired the lead developer to automate this process of identifying accounts to be blocked. This blockbot was strongly situated within that relatively tight-knit community’s existing understandings of what harassment was and how moderation ought to take place. The online community had been moderating their own web forum for some time before seeking to moderate Twitter, and they had a rich set of rules, procedures, concepts, and terms for referring to specific kinds of harassment or particular groups that engaged in harassing activity. Many of these terms are also prevalent in other online communities that are concerned with online harassment and have commitments to feminism and social justice, like “doxing” (releasing personal information) and “MRAs” (men’s rights activists).

When this community began using the bot-based collective blocklist on Twitter, the lead developer was initially confident that there was no need for additional criteria for blockworthiness. They told fellow community members that anyone who would get blocked in forum they hosted for themselves should be added to the blocklist. Yet as members began to populate the blocklist with accounts they had encountered on Twitter, a split became visible

about people who espoused ideological views that the community strongly opposed. In particular, there was a debate about the blockworthiness of other feminists who shared most of their views and goals, but were allegedly hostile towards their community's advocacy for sex workers and transgender/transsexual individuals. Members were also split between subscribers who simply did not want to see any tweets from people who espoused opposing ideological positions and those who wanted to reserve the blocklist for people who sent aggressive or repeated unsolicited replies.

After some debate, the lead developer responded to these different constituencies by making a change to the bot's software code to split the blocklist into multiple lists. After the change, there was a top level for those who were aggressive and threatening to individuals, a middle level for those who were espoused ideologies that the community opposed, and a lower level for those who simply tweeted ignorantly of the issues that mattered to this community. Such a change worked well for this community, and it should be seen not as a technical solution but a socio-technical reconfiguration. The discussion took place within a shared context that had substantially more internal coherence than the general population of people who use Twitter, and this three-pronged list reflected this community's existing governance structure for moderating their own forum, which made these similar distinctions when deciding whether someone should be indefinitely banned, temporarily banned, or simply warned. This shows how this algorithmic system is as much made of up software code as it is the shared discourses, practices, internal conflicts, standards, and political and ideological commitments of the people who participated in its design, development, and deployment. Contexts and existing social artifacts do not determine the configuration of an algorithmically-supported system, but they do work as resources that people leverage when seeking to carry out their goals in a new space (L. Suchman, 2007).

## **5. Discussion**

### **5.1 Concerns about fragmentation and automated discrimination**

Blockbots can be celebrated as a way for counterpublic groups to moderate their experiences online, but it raises two of the more common fears expressed by those who discuss the Internet and the public sphere. The first fear is the fragmentation of the public sphere into separate, polarized groups, and the second is the use of algorithmic systems as discriminating gatekeepers. Critical scholars must pay close attention to such processes of inclusion and exclusion, because they are the mechanisms in which modes of cultural domination operate (Williams, 1977, p. 125). Scholars and activists who focus on issues of gender, sexual orientation, race and national origin, class, disability, and many other axes of subordination have long critically interrogated the modes of filtering and exclusion that work to erase certain kinds of activity from 'the social', 'the political', or 'the public.' Cultural curation has long been a core mechanism in which domination is reinforced (operating more subtly than more visible acts of formal exclusion and repression), and so it is understandable for blockbots to initially appear suspect by those who are deeply concerned with these issues of social justice. Many scholars both in computer science as well as the social sciences and humanities have become concerned about how automated systems for filtering and moderating content can

function as invisible gatekeepers, operating similar to the hierarchical editors that controlled content in more traditional media like newspapers, television, and radio. Given the way recommendation and filtering systems work, there is strong potential for such systems to even influence elections, as a controversial study by staff at Facebook, Inc. suggested (Bond et al., 2012). In response, scholars have sought projects focusing on “algorithmic accountability” (Diakopoulos, 2015) and have pushed for more transparency in how such platforms filter and promote content. It makes sense to ask if blockbots raise similar kinds of concerns that algorithmic recommendation and filtering systems do.

With blockbots, such fears of around the biases of algorithmic systems must be understood in their counterpublic context. Blockbots do not operate according to a top-down model of gatekeeping, as they are built for particular communities to come together and enact a different mode of gatekeeping than is the default in Twitter. Concerns about blockbots as a potentially oppressive mode of filtering must be understood in the context of harassment, which works to exclude, repress, and silence public participation, as Frasier (1990), Herring (1999), and Citron (2014) all review. Before blockbots existed, there was already a complex social and technical system that shapes people’s experiences of Twitter as a networked public space, which in turn shapes broader understandings of what ‘the public’ believes. That system is the entire ecosystem of Twitter as it currently exists in the status quo, which includes all the people who work to silence their targets from public participation in ways that are not seen as ‘abuse’ by Twitter, Inc. staff. We must also keep in mind Frasier’s critique of ‘the public’ as a hegemonic way of elevating one of many publics above all others. Part of the privilege of dominant groups is the ability to define the terms of the public by deciding what does and does not belong, as well as defining the current state of the world as the natural default. Blockbots involve commandeering that privilege to institute a different definition of the public, even if it only has direct effects for those who choose to opt in to the counterpublic group’s redefinition. This redefinition of the public calls attention to the different understandings about what a social networking site is and ought to be – and who it ought to be for.

## **5.2 Technological solutionism**

Blockbots are certainly a technology that is deployed to help solve the problem of harassment on Twitter, but they should not be seen as the kind of top-down technical solution that can be installed to fix the problem ‘once and for all’ – which, as Barbrook & Cameron and Morozov argue, is often accomplished by shifting the burden of responsibility to individual, isolated users. Instead, blockbots have emerged as more communal, counterpublic responses to harassment. Like Kelty’s discussion of open source software infrastructure (Kelty, 2008), they help form a “recursive public” in which the ideals of the group are intentionally embedded in the design of the software that supports their activities. Blockbots are a different kind of computationally-supported mode of platform governance, which can be seen in the formation of broader anti-harassment communities around blockbots and the thoughtful reconfigurations in blockbot infrastructure. In fact, despite the utility that blockbots have in helping people shape their own experiences online, they are perhaps even more impactful in that they have provided a catalyst for the development of anti-harassment communities. These groups bring visibility to the issue and develop their own ideas about what kind of a network public Twitter ought to

be. Blockbots provide a concrete alternative to the default affordances of Twitter, showing a different version of a public where people have the agency to selectively tune out of harassment without dropping out of public participation altogether. Their existence has sparked broader conversations about what public discourse online ought to look like, as well as what kind of relationship platform owner/operators ought to have with “their” users. The kind of bottom-up, decentralized, community-driven approach exemplified by blockbots stands in opposition to the more traditional top-down, centralized, systems administration approach exemplified by Lessig’s “code is law” argument and much of the “politics of algorithms” literature. Blockbots are as much of a social solution as they are a technological one, and their strength is in their capacity to serve as multiply overlapping sites for collective sensemaking and reflective reconfiguration among counterpublic communities – rather than seeking to deploy a technological solution that seeks to fix the problem for all users, once and for all.

### **5.3 The human work of infrastructure**

Blockbots are not just automated software agents; the algorithm cannot exist in the world without the constant care and concern of dedicated human beings – a finding echoed throughout studies of agency and technology (Latour, 1992; J. Law & Mol, 1995; Susan Leigh Star & Griesemer, 1989). Software bots are ongoing accomplishments which rely on such a wide set of allies and infrastructures to exist as they do. This runs contrary to the prevalent idea of the autonomous, independent, automated “killer robot” that inhabits human form and mercilessly acts according to a pre-programmed set of directives. To say that bots are deeply human is not just to say they are developed, designed, and deployed by humans or that humans play a role in shaping the design of artifacts (like all technology; see Bijker, 1995). It also emphasizes how the average, everyday existence of bots involves a wide variety of human work – much of which often sits outside of stereotypical notions of what “technical work” involves, as a number of ethnographies of tech workers have also shown (Kunda, 1992; Orr, 1996).

These different forms of work are simultaneously social and technical, as infrastructural issues frequently are (Star, 1999). This work demands technical expertise in navigating topics like OAuth protocols, rate limits, and API keys, as well as more social expertise in topics like organizing allies around a common cause, negotiating with more dominant institutions, and establishing more stable organizational forms for ensuring the bot’s continued existence. As bots are not built into server-side software platforms but instead run on servers independent from them, getting access to reliable hosting is crucial – this involves not only systems administration expertise, but also the ability to raise funds for hosting. Opponents of blockbots have organized petitions to Twitter to have the bot’s API key revoked due to alleged terms of service violations. This means that a bot developer must be able to navigate the technical specifications encoded into a legally-binding document in the context of a fluid organization that enforces such terms of service. Even developing and modifying an algorithmically supported blockbot’s core algorithm (such as who gets put on or taken off the list) involves abstracting from both database schemas and ideals about public discourse.

We cannot just look at the code of blockbots, because they continue to be a part of Twitter because of the work that their developers must constantly do to keep it running in the manner that it does. Making sure the bot doesn’t go over Twitter’s API rate limits (which limit

the number of requests a bot can make) is one of these tasks that is a core part of what it means to be a blockbot developer. However, so is talking to police officers and lawyers about threats. GitHub, blocktogether.org, and Amazon EC2 are part of the core infrastructure that blockbots needs to continue operating in the manner that it does, but are non-profit organizations that help support both blockbots and a wide range of technical and social interventions around online harassment. When we turn our attention from ‘unpacking’ the code of an algorithmic agent and instead taking an algorithms-in-the-making approach, we are able to better identify these heterogeneous elements and the roles they play in doing far more than automating a task. These blockbots certainly make the work of responding to harassment easier for those targeted, but their significance also operates at a far broader level. As design projects, these bots are also moments in which competing assumptions about what Twitter is and ought to be as a networked public space are made explicit and contestable. To date (and to my knowledge), this bespoke code has not been integrated into Twitter’s server-side codebases,<sup>61</sup> especially not its existing highly-automated human-computational systems used internally for content moderation. Yet blockbots have achieved perhaps an even greater success: they have become a part of an overlapping discussions and commentary about how not just Twitter, but social networking sites in general ought to respond to harassment. From op-eds in the New York Times, Gawker, and Breitbart to discussions across reddit, Tumblr, and 8chan<sup>62</sup>, blockbots have become one of many touchstones that frame and contextualize this issue -- even by those who violently oppose them.

Like all bespoke code, blockbots do not stand alone; they are all part of a larger and more heterogeneous project in which people are actively imagining, implementing, evaluating, and iterating alternative understandings about how moderation around harassment ought to take place. On one level, this can be seen through an innovation studies lens as a way for new product ideas to be generated and evaluated without needing the kind of formal approval that would be necessary if Twitter, Inc. decided to beta test such a feature internally. Yet more importantly, the success of such bespoke code ought not to rest on whether such code is ultimately officially incorporated into server-side codebases. As a blockbot developer repeatedly emphasized to me in an interview, implementing the blocklist feature was certainly an important goal, but it soon became apparent that this was only one part of a broader goal around responding to online harassment – something which, she remarked, would likely not be solved simply through implementing new software features.

## 6. Conclusion

Blockbots are a way to understand how algorithmically-supported systems of knowledge production are situated within particular contexts and infrastructures. Such systems continually develop as people come to understand what it means to ask and answer a question like, “who is and is not a harasser?” For example, the development of one of the earliest blockbots certainly involved a lead programmer who realized a particular vision about how automation could help people efficiently aggregate information identifying particular harassers,

---

<sup>61</sup> Twitter does now have a way to manually export/import an account’s blocklist to/from XML files. However, this is not anywhere near the functionality of bot-based collective blocklists.

<sup>62</sup> Breitbart and 8chan are two venues that are closely associated with GamerGate.

creating and implementing a shared blocklist. However, this bot also relied heavily on the existing practices and priorities around harassment and moderation that had been developed in a particular online community of like-minded individuals, of which the lead developer was a veteran member. This algorithmic system did not spontaneously emerge as the purely rational application of a technocratic logic to an abstract information processing problem, even though it can certainly be seen through such a lens. Yet the problem with taking such an abstracted analysis of such a blockbot is that it passes over how the bot was strongly situated within that community's existing understandings of what harassment was and how moderation ought to take place. Such an algorithmic system is as much made of up software code as it is the shared discourses, practices, internal conflicts, standards, and political and ideological commitments of the people who participated in its design, development, and deployment.

If we approach algorithmic systems that produce knowledge for governance purposes in a more traditional understanding of scientific knowledge, then this argument about the inextricable humanness of such systems can only be taken as evidence of bias or a lack of objectivity. Yet the question about who is and is not a harasser has no single correct answer, as it only has meaning within broader socio-cultural systems that define and delimit acceptable and unacceptable behavior. Algorithmic systems that operate in other digitally-mediated environments similarly make judgments about questions that have no single universal answer, as there are no context-independent answers to questions about what *really* is a relevant post on Facebook, search result in Google, product recommendation in Amazon, or movie recommendation in Netflix. However, this 'relativist' stance does not exempt such systems and those responsible for them from accountability or claims of bias and discrimination. Such accounts instead show how debates about objectivity, accountability, bias, or discrimination do not exist external to some essential technical core of an algorithmic system; they are as much a part of them as source code is. In studying algorithms-in-the-making, source code becomes entangled in the lives and activities of people who are involved in the design, development, and deployment of an algorithmic system. People continually and routinely iterate on not just the algorithmic system itself, but on aspirational ideals about what a good algorithmic system ought to be. They develop one version that they think will work well based on what they know at the time, deploy it (either in testing or in production), have their assumptions challenged, and respond to such objections in a variety of ways.

Sometimes these shifts result in a change to an algorithm which processes certain kinds of data from inputs into outputs according to well-defined steps. For example, in response to criticism that the collective blocklist curated by one blockbot contained too many accounts which were inappropriately added, the developer responsible for the bot changed the threshold for adding a new account to the blocklist, such that two approved 'blockers' would need to report a user for them to be added, rather than just one. Other times, the algorithm stays the same, but the way that people understand it as a mode of knowledge production shifts, resulting in a different algorithmic system built around that same 'core' algorithm. For example, in response to a similar criticism that the blocklist automatically generated by another blockbot (which was based on a social network graph) included too many false positives, the lead developer created an appeals board. The board established a documented policy and formalized procedure for reviewing cases of people who believed they should be removed from the blocklist. Both of these ways of changing how an algorithmic system operates can involve significant shifts in

how people understand and use it. New social practices and software code was established in the creation of both an appeals board and a shift to requiring two blockers to add an account to the blocklist. In both cases, these decisions involved reflection by developers and non-developers about what kind of algorithmic system was being developed, how such a system operated in an average, everyday context, and what values were important to incorporate in such a system.



## Chapter 9. Conclusion

### 1. Overview

In this conclusion, I first give a summary of the overall argument of the dissertation, elaborating on the *algorithms-in-the-making* approach I took in studying bots in Wikipedia and Twitter. Next, I identify three different themes that have emerged in these different cases: proceduralization, the regulation of bots, and bots as speculative projects. Finally, I conclude with an essay on the ideological assumptions that are present in many of today's major user-generated content sites, including not only Wikipedia and Twitter, but sites like reddit as well. I stress that we must not forget that bot developers are volunteers who support moderation practices that benefit the platform's owner-operators by keeping the site orderly, but in ways that neither cost them money nor open them up to critiques of censorship or gatekeeping.

### 2. Summary

In this dissertation, I have worked to introduce and apply a way of studying automated software agents, which focuses on what I term *algorithms-in-the-making*. When I began working on bots more than a half-decade ago, I took a highly instrumental approach to these automated software agents. I focused on the impacts they had and thought of agency as a zero-sum game: every task or act of decision-making delegated to a bot meant that a human was not performing that task or making that decision. This shows the limitations of focusing on the delegation of tasks or acts of decision-making to 'algorithms.' This changed when I began studying algorithms from a more situated and long-term view, using ethnographic and historical methods that let me see bots as projects that were changing over time. When I looked at what kinds of work were actually being delegated to automated software agents, I found that the agent was rarely delegated the entire task or decision as a whole. Rather, I found that these bots were more often delegated the articulation work needed to accomplish the task or make the decision in question. In fact, had these bots been delegated the entirety of a task (like deciding what articles ought to be deleted from Wikipedia or who ought to be blocked on Twitter), then they perhaps may have never succeeded.

Scholars and commentators have long focused on the role of software in the governance of digitally-mediated environments, including Wikipedia. Lawrence Lessig famously argued that "code is law" (Lessig, 1999), speaking to the immense governmental responsibility that software designers and developers have in enacting "different forms of life" (Lessig, 2006, p. 88; c.f. Wittgenstein, 1958) in the sites they own and administer. In this dissertation, I focused on automated software agents, commonly referred to as 'bots,' in which individuals who do not have privileged access to modify server-side code instead automate the actions available to a user account. In this way, bots are a mode of software development in which individuals who are not 'server sovereigns' are nevertheless able to design, develop, and deploy software that has profound effects and impacts on how a digitally-mediated environment operates. While there are many different kinds of bots that operate in many different sites, I specifically focused on bots that are delegated aspects of governance work in two sites: the collaboratively-edited encyclopedia project Wikipedia and the social networking and microblogging site Twitter.

In the research I conducted for this dissertation, I used a variety of methods – ethnographic, historical, and statistical – to investigate the same kinds of issues that scholars have long studied about the governance of digitally-mediated environments through software code. My studies of bots as bespoke code have given me a different angle on Lessig’s phrase and how many scholars have used it to conceptualize the relationship between power and code, where the code is typically seen as analogous to law due to its governmental effects. While bots do have similar kinds of effects, the kinds of interactions I observed between bot developers and others in these sites (including other bot developers, non-developer users, and systems administrators) reminded me of the kind of gestalt shift Science and Technology Studies scholars often discuss between taking something as “ready-made” versus “in-the-making” (Callon, 1987; Latour, 1987; Shapin, 1992). I found that the code powering bots was not law in the sense of a rigid set of already enacted rules that everyone must follow. Rather, such code was law in the sense that it was more often a fluid medium in which some people (but certainly not all people) worked to articulate a common abstract understanding that represented their ideas and values about what kind of a world they want to live in.

While there are many different ways in which automated software agents like bots can be studied, the *algorithms-in-the-making* approach I took throughout this dissertation holds that it is important to examine bots before they become settled and deployed as ready-made artifacts. My approach focuses on how algorithmic systems – particularly those that are delegated aspects of governance work – can be a way in which developers and non-developers imagine, articulate, enact, contest, negotiate, and subvert ideas about what Wikipedia as an encyclopedia project or Twitter as a social networking site is and ought to be. I found that developers and non-developers were not just developing, negotiating, and iterating on versions of software code. They were also developing, negotiating, and iterating on ideas about how people ought to behave to each other, particularly when enforcing norms in a shared socio-technical environment. In studying controversies over bots, many of these implicit assumptions were made explicit as developers and non-developers worked to express not only what tasks a bot ought to and how it ought to do it (if at all), but also more fundamental notions about how norms and governance are to operate in such spaces.

## **2. Themes**

### **2.1 proceduralization**

The first set of issues these cases raise relate to the contrasting concerns of proceduralization and participation, which are made quite visible in exploring the delegation of governance work to automated software agents. As I discuss in chapter 4, since Wikipedia’s initial creation in 2001 as the encyclopedia that “anyone can edit,” the project has developed increasingly formalized processes and policies for regulating the content of encyclopedia articles and the conduct of volunteer contributors. As many Wikipedia researchers have discussed (Geiger & Ribes, 2010; Halfaker et al., 2013; Konieczny, 2010; Kriplean, McDonald, Beschastnikh, & Golder, 2007; Lih, 2009; Pike, Joyce, & Butler, 2008b; Reagle, 2010; Tkacz, 2015; Wattenberg et al., 2007b), these increasingly-bureaucratic venues for specialized decision-making were instituted as part of broader shifts to focus on article quality rather than raw quantity of content, which first began around 2005. As I show through examining the

history of what are today highly-structured, fast-paced processes for administrative decision-making, bots have played crucial roles in the development these specialized venues. The earliest of these processes were not initially supported through algorithmic agents; rather, bots were designed, developed, and deployed as Wikipedia's popularity skyrocketed around 2005. This popularity led to exponential growth in both the number of newcomers to Wikipedia and the number of edits made to articles (Suh et al, 2009) – and correspondingly, exponential growth in the amount of work for those who were deciding, for example, which articles ought to be deleted from Wikipedia. As bots were developed to help “clerk” for these specialized processes, administrative logics had to be made explicit if they were to be automated, and competing visions of the future of Wikipedia played out in part through the development of clerk bots. This includes bots that never saw the light of day, such as a proposed bot that would let those opposed to the deletion of any article in Wikipedia automatically voice their opposition to any article nominated for deletion in the project's “Articles for Deletion” process. Today, there are dozens of bot-supported administrative processes across Wikipedia, and I argue that part of what it means to become a Wikipedian in 2015 is bound up in learning how to participate in such spaces.

Similarly, proceduralization was a core theme in my study of blockbots on Twitter. Many of the blockbot projects began as small side projects by an individual, but then expanded dramatically as the bot became more popular. This expansion often raised issues that necessitated further specifications of what kind of tasks blockbots and blockbot groups were doing. For example, one major blockbot's lead developer initially did not see a need to even specify the criteria for adding an account to a blacklist, given how the group of authorized blockers all came from the same tight-knit online community. However, differences arose over what kind of task they were collectively engaged in, and the developer ultimately split the list into three escalating levels, each with specific defined criteria. The bot did not autonomously decide which level an offending user should be placed in; rather, it was delegated the articulation work of supporting a process where authorized blockers could engage in this more refined classification work. The decision about how to reprogram the bot to accommodate the multiple constituencies of blockers and subscribers raised and resolved a tension over what it meant to add someone to a collectively curated blacklist. Similarly, other reconfigurations of the bot's source code were made in response to perceived needs to further refine and extend the procedure for curating the blacklist, like adding a two-person process of blocker review, as well as support for documenting why an account was added to a blacklist. Finally, the case of one blockbot's appeals board shows how even seemingly ‘fully-algorithmic’ systems can be extended with new procedures – which do not have to be algorithmic. The appeals board that decides on who should be whitelisted is made up of humans who have a set of written standards for determining blockworthiness, which they use in a deliberative process of decision making.

## **2.2 The regulation of bots**

The second set of issues I discussed are around the governance and regulation of bots themselves. Automation – and the delegation of work to technological artifacts in general – can be cast as force multipliers for the already-powerful, deployed by technocrats to enact their will (or the will of those who direct the technocrats) on the world. As Bruno Latour quips in his essay on the morality of speed bumps that punish speeding ambulances and teenagers with

equal rigidity, “no human is as relentlessly moral as a machine, especially if it is (she is, he is, they are) as ‘user friendly’ as my Macintosh computer” (Latour, 1992). Yet in my studies of bots, I have found that the delegation of governance work to automated software agents frequently becomes a moment in which non-developers who disagree with a certain bot-instrumented worldview come on the scene to express their dissent. In chapter [x], I discuss the first bot to enforce a discursive norm in Wikipedia’s discussion spaces (rather than make automated edits to encyclopedia articles). This norm was that in making a comment in a discussion space, everyone should leave a signature and timestamp, which was established as a recommended Wikipedia-wide “guideline.” The bot’s developers and many veteran Wikipedians thought this norm was so universal that those breaking it by not leaving signatures or timestamps in their comments (which is not natively supported in the MediaWiki software platform hosting Wikipedia) had simply “forgotten.”

When the bot was approved by Wikipedia’s then-nascent “Bot Approvals Group” and began enforcing this norm across the entire encyclopedia project, a number of people who disagreed showed up to the bot developer and the members of the Bot Approvals Group to object. The debate that ensued was on one level about this bot-enforced norm, but soon became about the broader issues regarding how norms operated in Wikipedia and the roles of bots in enforcing order. Does the existence of a documented “guideline” give sufficient cause to authorize a bot to universally enforce that guideline? Should people who intentionally chose to ignore a guideline (which was distinct from the stricter class of norms called “policy”) be allowed to opt-out of the bot’s ‘assistance’? What responsibilities do bot developers have to those who object to the operation of their bots? As Wikipedians worked to answer these questions, the solutions to this controversy involved the production of broader standards, procedures, norms, and meta-norms that provided a strong foundation for both the roles of bots and norms in Wikipedia. However, just as participation in Wikipedia’s administrative processes requires substantial amounts of specialized expertise, so is the capacity to participate in this kind of algorithmic governance. That said, the kind of expertise required in Wikipedia is not so much in the ability to read and write source code, but rather in the ability to articulate one’s concerns in alignment with the extensive discursive and normative structures that Wikipedians have developed to resolve disputes.

### **2.3 Bots as speculative projects**

The final set of issues I discuss are around bots as kinds of speculative projects, whose purpose extends beyond their immediate impacts. As projects, they have commonalities with design fiction and critical technical practice (Agre, 1997) in that they provoke reflection on how power relations and cultural values are embedded in the design, but there are noticeable differences to these existing approaches in Human-Computer Interaction. A number of bots I have studied are based in the purposeful re-design of existing, dominant systems, where automation is used to extend the affordances of a site like Wikipedia or Twitter in particular value-driven ways. Such bots are part of a broader project to imagine, implement, and iterate alternative understandings about who the site is built for – from supporting Wikipedia’s veterans in gatekeeping contributions to supporting more newcomer-focused venues for socialization and mentoring. Furthermore, as I explored in section 3, bots have played a crucial role in issues of online harassment in Twitter, particularly around highly-coordinated gender-

based harassment campaigns associated with the GamerGate movement. For years, staff at Twitter, Inc. had taken a minimalist stance to moderation of content and suspension of user accounts, which was strongly critiqued by as harassment campaigns gained visibility in a variety of social and mass media outlets. Many of targets of harassment and their allies wrote passionate op-ed articles and engaged in other advocacy work to try and get Twitter to change its policies and platform. In addition, several groups developed a variety of independent “block bots” that used different kinds of computationally-assisted approaches such that a subscriber to a block bot would no longer see messages or mentions from suspected harassers in their browsers or apps. These blockbots certainly make the work of responding to harassment easier for those targeted, but their significance also operates at a broader level.

### **3. Infrastructures and ideologies of user-generated content platforms**

#### **3.1 The rise of user-generated content**

In the past 10-15 years, the concept of ‘user-generated content’ and ‘Web 2.0’ has come to stand in for a wide series of shifts in both information technology and popular culture. Hundreds of millions of people regularly visit sites like Facebook, Wikipedia, Twitter, and reddit in order to get information, news, and commentary about practically any topic they find interesting, making these sites a form of mass media. Yet unlike many of the dominant mass media sources which pre-dated these sites (and still have substantial readership), the content of these sites are not authored or even typically reviewed by the organizations which own and operate these sites. Instead, people (cast as ‘users’) are asked to perform the work of writing, editing, reviewing, summarizing, and discussing content, which a broader public is to consume. As many social scientists and cultural critics have argued (Gillespie, 2010; van Dijck, 2013), there is a dominant ideological assumption that these sites are neutral ‘platforms’ in which the people of the world are empowered to freely express themselves. This assumption simultaneously operates at political, social, and technological levels. As policy scholars and policymakers have long discussed, it positions the sites as ‘common carriers’ who claim to not be legally responsible for the content ‘their users’ create and curate. This assumption also shapes how developers at these organizations design their sites so that users can effectively engage in such ostensibly self-organized content creation and curation.

At the organizations which legally own and operate these user-generated content sites, employees are frequently cast as mere infrastructural workers. They are not seen as responsible for creating or curating the content that has made these sites into household names, but rather are presented as doing whatever needs to happen to empower ‘their users.’ However, we cannot take such an assumption for granted, even if the people who work for these organizations genuinely see their roles as neutral support staff ‘just keeping the lights on’ – in fact, that only makes such a critical interrogation even more important. As many scholars have argued, there is substantial power and authority in this often-overlooked infrastructural work, which is a broad lesson we have learned from centuries of cases about technology and society (Bowker & Star, 1999; Hughes, 1983; Latour, 1987; Shapin & Schaffer, 1985; Winner, 1986). Star and Ruhleder (1996) discuss the invisibility of successful infrastructures, which work so well precisely because they disappear into the background and become routine. Today, the computational systems supporting user-generated content sites like Facebook, Wikipedia,

reddit, and Twitter establish the very conditions under which user-generated content is made possible, defining and delimiting the practices which constitute this form of cultural production. This can be seen in the simple fact that Facebook, Wikipedia, reddit, and Twitter look so different, even though they are all similarly framed as attempts to empower a broadly-defined public to create and curate information resources about topics relevant to them.

It is crucially important to interrogate this widespread and far-reaching assumption that sites like Facebook, Wikipedia, Twitter, and reddit are purely neutral platforms, governed by organizations that only exist to let members of the public act out their own ideas of what it means to create and curate content about the world around them. This assumption is strikingly similar to political-economic ideologies in which the ideal state is one that takes a seemingly ‘hands off’ approach, letting the people flourish on their own. Like many governments, these new media organizations are often incredibly wary of ‘censoring’ the people who they often refer to as members of a shared ‘community’ – typically only doing so when they are legally compelled to by a government who can use force to shut down their servers if they violate that country’s laws. Furthermore, as Lawrence Lessig has long argued (1999, 2006), these ‘server sovereigns’ often have the technical capacity to conduct highly-automated, algorithmically-supported regulatory practices – the kind which would seem undeniably Orwellian if conducted by a traditional government against its people. And as Lessig argues, the people who use these platforms are also closer to customers than citizens. In one sense, they do have the freedom to leave a site for a better alternative, if one exists – albeit often with substantial cost. However, when it comes to decisions about which users are blocked from contributing or what actions are algorithmically prohibited, server sovereigns are often at best benevolent dictators. (Wikipedia and the Wikimedia Foundation are an interesting limit case of this, which I will explore further in later sections.)

### **3.2 Discourses of libertarian governance**

I bring up this ‘benevolent dictator’ trope not to critique these server sovereigns as dictators, but to contextualize their reticence to directly engage in certain kinds of governance work. I do argue that there is a widespread and ideologically-charged distinction at work in user-generated content platforms, one which works to distinguish between unaffiliated users who are empowered to create content and employees who do merely do routine infrastructural work. There is a strong set of libertarian assumptions which are deployed to legitimize and reinforce this distinction, often explicitly articulated as such. This kind of discourse can be seen in comments by Alexis Ohanian, co-founder of reddit and CEO of Reddit Inc., who stated in a 2012 Forbes interview that “maybe libertarians especially like reddit because it is a perfect marketplace of content.” This interview was focused the site’s role in the SOPA/PIPA blackout protests, where reddit staff replaced the entire website with an injunction to protest the proposed U.S. copyright law. In it, Ohanian drew heavily on this distinction situating the users as the only truly empowered agents, with the staff merely working to realize their collective intentions: “I mentioned it [blacking out the site] to the Reddit team ... But taking a site like Reddit down wouldn’t feel right coming from the top down. They didn’t start thinking about it seriously until it started bubbling up from the Reddit community.”

This set of libertarian assumptions is not just present in the discourse of the site's owner-operators; it has given theoretical foundations, particularly in Yochai Benkler's work on "peer production." He argues that collective information goods in these kinds of platforms are produced in a similar way to how a free market determines a price, and this become a dominant way for both academics and practitioners to understand how these projects operate. In Benkler's 2006 book *The Wealth of Networks* – the title an explicit homage to Adam Smith – he celebrates how Wikipedia and other similar projects are operated not through top-down control, but through the "emergence of coordinate effects, where the aggregate effect of individual action, even when it is not self-consciously cooperative, produces the coordinate effect of a new and rich information environment" (5). For Benkler, if we just provide the right conditions for people to individually share information relevant to them, we can have a "networked public sphere" that is "independent of both government control and market demands" (177). While I critique these ideologies, they have to be situated in the context of the enormous governmental authority these organizations wield over people who use their systems. This authority which is principally based in these employees having privileged access to the servers hosting these systems, which is a condition they cannot escape.

It is also important to recognize that this is only one of many possible assemblages of ideologies, infrastructures, and practices which work to constitute governance work in large-scale computational platforms. Governance in reddit is quite different than in Facebook, which as an organization seems have few qualms about creating and enforcing its own relatively-strict 'community standards', with the help of an army of algorithmic and human reviewers. As such, Facebook as an organization is frequently the subject of controversy and protest by those who think its policies are either too strict or too lenient, such as in issues around photos of breastfeeding – once prohibited because nudity was not allowed anywhere on the site, but are now specifically allowed as an exception. Facebook also has a strict real name only policy, where people who use suspicious names can have their accounts deleted if they do not verify their identity to Facebook employees, which involve sending an image of a government-issued ID or multiple documentation from bank statements, utility bills, and so on. This real name policy has been heavily critiqued by privacy advocates, most notably those who argue that it disproportionately harms LGBTQ individuals, who routinely use pseudonyms and stage names to protect their identity.

In all, it is quite reasonable for server sovereigns to look at Facebook's heavy-handed governance of their site and balk at the amount of work they may be expected to do if they decided to more directly enforce the same kinds of standards and policies. Instead, the Wikimedia Foundation, Reddit Inc., and Twitter Inc. have long limited their technical authority to unilaterally remove content to material that is blatantly illegal under U.S. law (all have servers based in the U.S.) like child pornography or copyrighted material. While this has been changing somewhat as reddit and Twitter have taken some steps towards removing hate speech and harassment on the site, I see such moves as token gestures. Furthermore, many of the actions that such companies have taken are not acts of moderation by staff who directly take actions on behalf of the company, but rather in 'empowering' users to moderate and filter content themselves.

### 3.3 Who does the work of sustaining order?

There is a large governance gap between the order that many people want to encounter when they enter many user-generated content sites and what they actually encounter in these sites. Just because this governance work is not being actively conducted by organizations like the Wikimedia Foundation, Twitter Inc., and Reddit Inc., does not mean that governance work does not take place on those sites. Other actors step in to fill this gap, which is something that is ostensibly celebrated in theories of peer production, self-organization, and emergence. According to both academic and lay theories of economic peer production, these contributors are generally equal with each other: everyone's content is made more or less visible through a decentralized jury of one's peers. This is why Benkler claims such a model of cultural production is more democratic than both traditional markets and media institutions. As Ohanian states in his Forbes interview: "Every Redditor is created equal, whether you're the highest karma Redditor or a brand-new Redditor with 10 karma points. No submissions or votes are more equal than others." Similar discourses abound around user agency in Wikipedia and Twitter, despite the fact that their platforms work to support quite different projects in which people to create and curate content for a broader public.

However, the inevitable governance work needed to enforce order in these platforms is increasingly being performed not only by what we may consider to be average, everyday individuals who are on relatively equal playing field with each other, but also by unofficial software developers. These developers are using fully- and semi-automated software agents (or bots) to conduct the kind of algorithmically-supported governance work they believe are necessary to keep a user-generated content platform functioning as it should. As such, these software agents are built with particular normative and conceptual understandings of what these sites are and ought to be. This governance work is not directly conducted by employees at these organizations, although it is infrastructurally supported by these organizations through their development of powerful and easily-accessible Application Programming Interfaces (or APIs). With the kinds of APIs made available to members of the general public to act within these sites a software developer with relatively few computational skills or resources can create bots which are capable of performing the kind of algorithmic regulation that Lessig identified in his "code is law" argument. These bots are also tacitly-authorized in that these organizations have not used their technical authority over the servers to block the bots from querying the API – as they routinely do with automated software agents that they classify as spambots or other kinds of malicious bots.

In one sense, bots are a delegation of governance work to algorithms, and a wide variety of issues are made visible in such moments. While human bureaucrats can engage in governance work as a situated practice with a certain degree of flexibility and autonomy, a computational agent must be programmed according to formalized procedures, which are often unable to capture the subtlety and nuance we rely on as agents in society. Lucy Suchman discusses this tension in her study of artificial intelligence agents (Suchman, 1987), and I have found many instances of conflict and controversy which stem from the over-coding of a certain kind of governance work. Yet in another equally important sense, bots are a different kind of delegation of governance work: one that involves a shift from assemblages of people and



technology inside the organizations which own and operate these platforms to assemblages of people and technology outside of it.

### **3.4 Where is a platform?**

The consequences for this delegation are profound, challenging our understanding of what a computational platform is and where it is to be found. A decade and a half of scholarship on the governance of online/virtual communities has generally operated under Lessig's "code is law" maxim, with its assumption that the server is a privileged site and seat of algorithmic power in these spaces. Scholars and practitioners in fields like 'user-centered design' also make such assumptions when they struggle with how developers and designers are to deal with their technically-constituted power; they often advocate that these server sovereigns willingly engage in participatory and empathetic approaches, just as Plato did with concept of the wise philosopher king. However, scholars are increasingly interrogating the modes of computational power that extend beyond code run on a single platform, such as the protocols of TCP/IP (Galloway, 2004), software mashups (Wong & Hong, 2008), browser extensions like ad blockers (Díaz, Arellano, & Iturrioz, 2008), and copy-and-paste code that extend platforms like Facebook's Like button (Gerlitz & Helmond, 2013).

If we study the code which is responsible for these sites operating in the way that we expect them to operate, a growing fraction of this code is developed and run outside of the organizations we tend to hold responsible for building and administering these sites, a phenomenon I have termed 'bespoke code' (Geiger, 2014). While each of these platforms is headquartered in San Francisco, the bots I discussed were developed by software developers living in places like Washington D.C., Berlin, Calgary, Minneapolis, Oakland, and Longparish. Unlike the massive, purpose-built server farms dedicated to hosting these sites, these bots are run on far more precarious infrastructure. They run on desktop computers in living rooms, on servers with unused capacity in university research labs, on dedicated servers funded by collectives of bot developers, and on cloud computing networks which can fully host a bot for as little as \$10 a month. These bots are also organizations unto themselves: while even a single human developer with a server and code can be said to constitute a form of organization, many bots are collectively developed and administered by groups that are more and less open to the public. Some of them even have their own policies and procedures for deciding how these automated software agents are to be developed and deployed. Sometimes this bot code is open sourced, posted for the general public to review and improve, with GitHub being a particularly important site for this kind of work.

### **3.5 A delegation and a re-distribution of governance work**

I use these cases as ways to discuss particular ideological assumptions about how user-generated content platforms are governed as socio-technical systems – assumptions which are being challenged through these new relations of algorithmic power. Researchers and practitioners must not only take into account the governance work that is delegated to algorithms developed and deployed by the employees of the organizations running some of today's largest computational platforms, but also those that exist alongside these more dominant and traditional modes of computationally-derived power. It is far easier to accept the libertarian

accounts of the minimalist server sovereign – accounts which draw strict boundaries between the user as empowered creator and the employee as maintenance maintainer – when we take for granted the work done by their delegates. An ostensible lack of governance work in one context is made possible by a vast amount of governance work performed in another context – work that is often infrastructural and invisible. In other words, organizations that own and operate sites like Wikipedia, reddit, and Twitter are seemingly-able to keep their hands clean of ‘censorship’ and algorithmic despotism because unofficial agents perform a substantial amount of work policing these sites in ways. While many of these platforms present themselves as neutral administrators who seek to delegate as much of that governmental responsibility to ‘their users’ as possible, bots complicate this in three ways. First, they show us how this user empowerment is far from equally-distributed, as these bot developers have both privileges and responsibilities which extend far beyond most other users. Second, they show us how this infrastructural work is far from neutral, as governance bots are made possible in these particular systems more than others due to specific configurations of infrastructure and practice. Finally, we must not forget of the substantial amount of work that bot operators perform as volunteers, working to support moderation practices that benefit the platform’s owner-operators by keeping the site orderly, but in ways that neither cost them money nor open them up to critiques of censorship or gatekeeping. Instead, these can be deferred to the bot operators themselves – which in the case of Twitter’s blockbots is particularly costly, given the legal threats and death threats that many blockbot operators face for this work. Before embracing bots as a novel form of innovation and user empowerment, we must take a hard look at the age-old maxim of Cassius: who benefits the most?

## 10. Works cited

- Adam, A. (2000). Gender and Computer Ethics in the Internet Age. *Computer Professionals for Social Responsibility*, 18(1).
- Adam, A. (2005). *Gender, Ethics, and Information Technology*. New York: Palgrave Macmillan.
- Agre, P. E. (1997). Toward a Critical Technical Practice: Lessons Learned in Trying to Reform AI. In G. Bowker, L. Gasser, S. L. Star, & B. Turner (Eds.), *Social Science Technical Systems and Cooperative Work: Beyond the Great Divide* (pp. 1–17). New York: Psychology Press.
- Akrich, M., & Latour, B. (1992). A summary of a convenient vocabulary for the semiotics of human and nonhuman assemblies. In W. Bijker & J. Law (Eds.), *Shaping technology/building society: Studies in sociotechnical change* (pp. 259–264). Cambridge: The MIT Press.
- Anderson, C. (2008). The End of Theory: The Data Deluge Makes the Scientific Method Obsolete. *Wired*. Retrieved from <http://www.wired.com/2008/06/pb-theory/>
- Antin, J., & Cheshire, C. (2010). Readers are Not Free-Riders : Reading as a Form of Participation on Wikipedia. In *Proc CSCW 2010* (pp. 127–130). ACM. <http://doi.org/10.1145/1718918.1718942>
- Antin, J., Cheshire, C., & Nov, O. (2012). Technology-Mediated Contributions: Editing Behaviors Among New Wikipedians. In *Proc CSCW 2012* (pp. 373–382). <http://doi.org/10.1145/2145204.2145264>
- Antin, J., Yee, R., Cheshire, C., & Nov, O. (2011). Gender differences in Wikipedia editing. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration - WikiSym '11* (p. 11). New York, New York, USA: ACM Press. <http://doi.org/10.1145/2038558.2038561>
- Ayres, P., Matthews, C., & Yates, B. (2008). *How Wikipedia Works*. San Francisco: No Starch Press. Retrieved from <http://ia700603.us.archive.org/31/items/HowWikipediaWorks/HowWikipediaWorks.pdf>
- Bakshy, E., Messing, S., & Adamic, L. (2015). Exposure to ideologically diverse news and opinion on Facebook. *Science*, 348(6239), 1130–1132. Retrieved from <http://www.sciencemag.org/content/348/6239/1130.abstract>
- Barocas, S., Hood, S., & Ziewitz, M. (2013). Governing Algorithms: A Provocation Piece. Available at SSRN 2245322, 1–12. <http://doi.org/10.2139/ssrn.2245322>

- Baym, N. (1995). The emergence of community in computer-mediated communication. In S. Jones (Ed.), *CyberSociety: Computer-mediated communication and community* (pp. 138–163). New York: SAGE. Retrieved from <http://psycnet.apa.org/psycinfo/1994-98724-001>
- Beaulieu, A. (2010). From co-location to co-presence: Shifts in the use of ethnography for the study of knowledge. *Social Studies of Science*, *40*(3), 453–470. <http://doi.org/10.1177/0306312709359219>
- Becker, H. (1982). *Art worlds*. Berkeley: University of California Press.
- Benkler, Y. (2006). *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. New Haven: Yale University Press.
- Berlin, I. (1969). *Four Essays on Liberty*. Oxford: Oxford University Press.
- Beschastnikh, I., Kriplean, T., & McDonald, D. (2008). Wikipedian Self-Governance in Action: Motivating the Policy Lens. In *Proc ICWSM 2008*. Retrieved from <http://www.cs.washington.edu/homes/ivan/papers/icwsm08.pdf>
- Bespoke, adj. (2013). In *Oxford English Dictionary Online*.
- Bijker, W. (1995). *Of Bicycles, Bakelites, and Bulbs: Toward a Theory of Sociotechnical Change*. Cambridge, Mass: The MIT Press.
- Blanchette, J.-F. (2011). A Material History of Bits. *Journal of the American Society for Information Science and Technology*, *62*(6), 1042–1057. <http://doi.org/10.1002/asi.21542>
- Blumer, H. (1954). What is Wrong with Social Theory. *American Sociological Review*, *18*, 3–10.
- Bond, R. M., Fariss, C. J., Jones, J. J., Kramer, A. D. I., Marlow, C., Settle, J. E., & Fowler, J. H. (2012). A 61-million-person experiment in social influence and political mobilization. *Nature*. <http://doi.org/10.1038/nature11421>
- Boshmaf, Y., Muslukhov, I., Beznosov, K., & Ripeanu, M. (2011a). The socialbot network. In *Proceedings of the 27th Annual Computer Security Applications Conference on - ACSAC '11* (p. 93). New York, New York, USA: ACM Press. <http://doi.org/10.1145/2076732.2076746>
- Boshmaf, Y., Muslukhov, I., Beznosov, K., & Ripeanu, M. (2011b). The socialbot network: when bots socialize for fame and money. In *when bots socialize for fame and money* (p. 93). <http://doi.org/10.1145/2076732.2076746>

- Bowker, G. C., Baker, K., Millerand, F., & Ribes, D. (2010). Toward Information Infrastructure Studies: Ways of Knowing in a Networked Environment. In *International Handbook of Internet Research* (pp. 97–117). [http://doi.org/10.1007/978-1-4020-9789-8\\_5](http://doi.org/10.1007/978-1-4020-9789-8_5)
- Bowker, G. C., & Star, S. L. (1999). *Sorting Things Out: Classification and Its Consequences*. Cambridge, Mass.: MIT Press.
- boyd, d. (2010). Social Network Sites as Networked Publics: Affordances, Dynamics, and Implications. In Z. Papacharissi (Ed.), *Networked Self: Identity, Community, and Culture on Social Network Sites* (pp. 39–58). Retrieved from <http://www.danah.org/papers/2010/SNSasNetworkedPublics.pdf>
- boyd, danah, & Crawford, K. (2011). Six Provocations for Big Data. *Computer*, 123(1), 1–17. <http://doi.org/10.2139/ssrn.1926431>
- Brubaker, J. R., & Hayes, G. R. (2011). SELECT \* FROM USER: infrastructure and socio-technical representation. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work - CSCW '11* (p. 369). New York, New York, USA: ACM Press. <http://doi.org/10.1145/1958824.1958881>
- Bruckman, A., & Forte, A. (2008). Scaling Consensus: Increasing Decentralization in Wikipedia Governance. In *Proc HICSS 2008* (p. 157).
- Bruns, A., & Burgess, J. E. (2011, August 27). The use of Twitter hashtags in the formation of ad hoc publics. *Proceedings of the 6th European Consortium for Political Research (ECPR) General Conference 2011*. Retrieved from [http://eprints.qut.edu.au/46515/1/The\\_Use\\_of\\_Twitter\\_Hashtags\\_in\\_the\\_Formation\\_of\\_Ad\\_Hoc\\_Publics\\_%28final%29.pdf](http://eprints.qut.edu.au/46515/1/The_Use_of_Twitter_Hashtags_in_the_Formation_of_Ad_Hoc_Publics_%28final%29.pdf)
- Brunton, F. (2013). *Spam : a shadow history of the Internet*. Cambridge, Mass.: The MIT Press.
- Bryant, S. L., Bruckman, A., Forte, A., Bryant, S. L., & Forte, A. (2005). Becoming Wikipedian: transformation of participation in a collaborative online encyclopedia. In *Proc GROUP 2005* (Vol. 6, pp. 1–10). Sanibel Island, Florida, USA: ACM. <http://doi.org/10.1145/1099203.1099205>
- Burke, M., & Kraut, R. (2008). Taking up the mop: identifying future wikipedia administrators. In *Proc CHI 2008* (pp. 3441–3446). Florence, Italy: ACM. <http://doi.org/10.1145/1358628.1358871>
- Burrell, J. (2009). The Fieldsite as a Network: A Strategy for Locating Ethnographic Research. *Field Methods*, 21(2), 181–199.

- Burrell, J. (2012). *Invisible Users*. Cambridge, Mass.: The MIT Press.
- Butler, B., Joyce, E., Pike, J., Joyce, E., & Butler, B. (2008). Don't look now, but we've created a bureaucracy: the nature and roles of policies and rules in wikipedia. In *Proc CHI 2008* (pp. 1101–1110). Florence, Italy: ACM.  
<http://doi.org/10.1145/1357054.1357227>
- Callon, M. (1986). Some elements of a sociology of translation: domestication of the scallops and the fishermen of StBrieuc Bay. *Power, Action and Belief: A New Sociology of Knowledge*, 32, 196–233.
- Callon, M. (1987). Society in the Making: The study of Technology as a Tool for Sociological Analysis. In *The Social Construction of Technological Systems New Directions in the Sociology and History of Technology* (pp. 83–103).
- Chasin, A. (1995). Class and its close relations: Identities among women, servants, and machines. In J. Halberstram & I. Livingston (Eds.), *Posthuman Bodies* (pp. 73–96). Bloomington: Indiana University Press.
- Chess, S., & Shaw, A. (2015). A Conspiracy of Fishes, or, How We Learned to Stop Worrying About #GamerGate and Embrace Hegemonic Masculinity. *Journal of Broadcasting & Electronic Media*, 59(March 2015), 37–41.  
<http://doi.org/10.1080/08838151.2014.999917>
- Clifford, J., & Marcus, G. (1986). *Writing Culture: The Poetics and Politics of Ethnography*. Berkeley: University of California Press.
- Collier, B., & Bear, J. (2012). Conflict, criticism, or confidence: an empirical examination of the gender gap in wikipedia contributions, 383–392.  
<http://doi.org/10.1145/2145204.2145265>
- Covino, W. (1996). Grammars of transgression: Golems, cyborgs, and mutants. *Rhetoric Review*, 14(2), 355–373.
- Cowan, R. S. (1983). *More Work for Mother: The Ironies of Household Technology from the Open Hearth to the Microwave by*. *The Business History Review*. New York: Basic Books. <http://doi.org/10.2307/3104353>
- Crawford, K. (2009). Following you: Disciplines of listening in social media. *Continuum: Journal of Media & Cultural Studies*.
- Crawford, K., & Gillespie, T. (2014). What is a flag for? Social media reporting tools and the vocabulary of complaint. *New Media & Society*.  
<http://doi.org/10.1177/1461444814543163>

- Crowston, K., & Howison, J. (2005). The social structure of free and open source software development. *First Monday*, 10(2), 1–21. <http://doi.org/10.1.1.89.3506>
- Darville, R. (1989). The Language of Experience and the Literacy of Power. In M. Taylor & J. Draper (Eds.), *Adult Literacy Perspectives*. Toronto: Culture Concepts.
- Darville, R. (2009). Literacy as practices, teaching as alignment: A message in a bottle. *Literacies*, 10. Retrieved from <http://www.literacyjournal.ca/literacies/10-2009/pdf/darville.pdf>
- Daston, L., & Galison, P. (1992). The Image of Objectivity. *Representations*. <http://doi.org/10.1525/rep.1992.40.1.99p0137h>
- Derthick, K., Tsao, P., Kriplean, T., Borning, A., Zachry, M., & McDonald, D. W. (2011). Collaborative sensemaking during admin permission granting in Wikipedia. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 6778 LNCS, pp. 100–109). [http://doi.org/10.1007/978-3-642-21796-8\\_11](http://doi.org/10.1007/978-3-642-21796-8_11)
- Diakopoulos, N. (2015). Algorithmic Accountability: Journalistic investigation of computational power structures. *Digital Journalism*, 3(3), 398–415. Retrieved from <http://www.tandfonline.com/doi/abs/10.1080/21670811.2014.976411>
- Doctorow, C. (2009, June 1). Search is too important to leave to one company – even Google. *The Guardian*. Retrieved from <http://www.theguardian.com/technology/2009/jun/01/search-public-google-privacy-rights>
- Doctorow, C. (2014, February 25). Chicago PD’s Big Data: using pseudoscience to justify racial profiling. *BoingBoing*. Retrieved from <http://boingboing.net/2014/02/25/chicago-pds-big-data-using.html>
- Elliott, M. (2006). Stigmergic Collaboration: The Evolution of Group Work. *M/C Journal*, 9(2), 8–11.
- Elmer, G. (2009). Exclusionary rules? The politics of protocols. In A. Chadwick & P. Howard (Eds.), *Routledge Handbook of Internet Politics* (pp. 376–383). London: Routledge.
- Eslami, M., Rickman, A., Vaccaro, K., Aleyasen, A., Vuong, A., Karahalios, K., ... Sandvig, C. (2015). “I always assumed that I wasn’t really that close to [her]”: Reasoning about invisible algorithms in the news feed. In *Proceedings of CHI 2015*. ACM Press. Retrieved from [http://www-personal.umich.edu/~csandvig/research/Eslami\\_Algorithms\\_CHI15.pdf](http://www-personal.umich.edu/~csandvig/research/Eslami_Algorithms_CHI15.pdf)

- Fernback, J. (1997). The individual within the collective: Virtual ideology and the realization of collective principles. In S. Jones (Ed.), *Virtual culture: Identity and communication in cybersociety* (pp. 36–55). London: Sage.
- Fiol, C. M., & O’Connor, E. J. (2005). Identification in Face-to-Face, Hybrid, and Pure Virtual Teams: Untangling the Contradictions. *Organization Science*, *16*(1), 19–32. <http://doi.org/10.1287/orsc.1040.0101>
- Ford, H. (2015). Infoboxes and cleanup tags: Artifacts of Wikipedia newsmaking. *Journalism*, *16*(1), 79–98.
- Ford, H., & Geiger, R. S. (2012). “Writing up rather than writing down”: Becoming Wikipedia Literate. In *Proc WikiSym 2012*. New York: ACM Press. Retrieved from <http://www.wikisym.org/ws2012/bin/download/Main/Program/p21wikisym2012.pdf>
- Forte, A., Kittur, N., Larco, V., Zhu, H., Bruckman, A., & Kraut, R. E. (2012). Coordination and beyond: social functions of groups in open content production. *Proceedings of the ...*, 417–426. <http://doi.org/10.1145/2145204.2145270>
- Foucault, M. (1977). *Discipline and punish: The birth of the prison*. Penguin Harmondsworth.
- Fraser, N. (1990). Rethinking the Public Sphere: A Contribution to the Critique of Actually Existing Democracy. *Social Text*, *26*, 56–80. <http://doi.org/10.2307/466240>
- Garfinkel, H. (1967a). “Good” organizational reasons for “bad” clinic records. In H. Garfinkel (Ed.), *Studies in Ethnomethodology* (pp. 186–207). Prentice-Hall.
- Garfinkel, H. (1967b). *Studies in Ethnomethodology*. Englewood Cliffs, N.J.: Prentice-Hall.
- Geho, P., & Dangelo, J. (2012). The Evolution of Social Media as a Marketing Tool for Entrepreneurs. *Entrepreneurial Executive*, *17*.
- Geiger, R. S. (2009). The social roles of bots and assisted editing programs. In *Proc WikiSym 2009*. New York: ACM Press. Retrieved from <http://portal.acm.org/citation.cfm?id=1641309.1641351>
- Geiger, R. S. (2011). The Lives of Bots. In G. Lovink & N. Tkacz (Eds.), *Wikipedia: A Critical Point of View* (pp. 78–93). Amsterdam: Institute of Network Cultures. Retrieved from <http://www.stuartgeiger.com/lives-of-bots-wikipedia-cpov.pdf>
- Geiger, R. S. (2013). *The Speed of Speedy Deletion*. Retrieved from [https://meta.wikimedia.org/wiki/Research:The\\_Speed\\_of\\_Speedy\\_Deletions](https://meta.wikimedia.org/wiki/Research:The_Speed_of_Speedy_Deletions)
- Geiger, R. S. (2014). Bots, bespoke, code and the materiality of software platforms. *Information, Communication & Society*, *17*(3), 342–356.



<http://doi.org/10.1080/1369118X.2013.873069>

- Geiger, R. S., & Ford, H. (2011). Participation in Wikipedia's article deletion processes. In *Proc WikiSym 2011*. New York: ACM Press. Retrieved from <http://www.wikisym.org/ws2012/bin/download/Main/Program/p21wikisym2012.pdf>
- Geiger, R. S., & Halfaker, A. (2013). When the Levee Breaks: Without Bots, What Happens to Wikipedia's Quality Control Processes? *Proc WikiSym 2013*. Retrieved from <http://opensym.org/wsos2013/proceedings/p0200-geiger.pdf>
- Geiger, R. S., Halfaker, A., Pinchuk, M., & Walling, S. (2012). Defense Mechanism or Socialization Tactic? Improving Wikipedia's Notifications to Rejected Contributors. In *Proc ICWSM 2012*. IEEE. Retrieved from <http://www.aaai.org/ocs/index.php/ICWSM/ICWSM12/paper/view/4657>
- Geiger, R. S., & Ribes, D. (2010). The Work of Sustaining Order in Wikipedia: The Banning of A Vandal. In *Proc CSCW 2010*. Atlanta, GA: Association for Computing Machinery. Retrieved from <http://portal.acm.org/citation.cfm?id=1718918.1718941>
- Geiger, R. S., & Ribes, D. (2011). Trace Ethnography: Following Coordination Through Documentary Practices. In *Proc HICSS 2011*. IEEE. Retrieved from <http://www.stuartgeiger.com/trace-ethnography-hicss-geiger-ribes.pdf>
- Gerlitz, C., & Helmond, A. (2013). The like economy: Social buttons and the data-intensive web. *New Media & Society*, 15(8), 1348–1365. <http://doi.org/10.1177/1461444812472322>
- Gillespie, T. (2010). The politics of “platforms.” *New Media & Society*, 12(3), 347–364. <http://doi.org/10.1177/1461444809342738>
- Gillespie, T. (2014). The Relevance of Algorithms. In T. Gillespie, P. Boczkowski, & K. Foot (Eds.), *Media Technologies: Essays on Communication, Materiality, and Society* (pp. 167–194). Cambridge, Mass.: The MIT Press. Retrieved from [http://6.asset.soup.io/asset/3911/8870\\_2ed3.pdf](http://6.asset.soup.io/asset/3911/8870_2ed3.pdf)
- Glorr, R., Schmidt, P., & Ghosh, R. (2010). *Wikipedia Survey*. Retrieved from [https://web.archive.org/web/20131209060146/http://wikipediastudy.org/docs/Wikipedia\\_Overview\\_15March2010-FINAL.pdf](https://web.archive.org/web/20131209060146/http://wikipediastudy.org/docs/Wikipedia_Overview_15March2010-FINAL.pdf)
- Gramsci, A. (1995). *The Prison Notebooks*. New York: International Publishers.
- Grosser, B. (2013). *How the Technological Design of Facebook Homogenizes Identity and Limits Personal Representation*. Retrieved from <http://bengrosser.com/wp-content/themes/grosser/files/GROSSER-facebook-identity.pdf>

- Grosser, B. (2014). WHAT DO METRICS WANT? HOW QUANTIFICATION PRESCRIBES SOCIAL INTERACTION ON FACEBOOK. *Computational Culture*, 1(4). Retrieved from <http://computationalculture.net/article/what-do-metrics-want>
- Habermas, J. (1989). *The Structural Transformation of the Public Sphere*. Cambridge, Mass: The MIT Press.
- Hacking, I. (2000). *The Social Construction of What?* Cambridge, Mass.: Harvard 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 sudden popularity is causing its decline. *American Behavioral Scientist*. Retrieved from <http://abs.sagepub.com/content/early/2012/12/26/0002764212469365.abstract>
- Halfaker, A., Geiger, R. S., & Terveen, L. (2014). Snuggle: Designing for Efficient Socialization and Ideological Critique. *Proc CHI 2014*.
- Halfaker, A., & Riedl, J. (2012). Bots and Cyborgs: Wikipedia's Immune System. *Computer*, 45(3), 79–82. Retrieved from <http://www.computer.org/csdl/mags/co/2012/03/mco2012030079-abs.html>
- Hammer-Lahav, E. (2010). RFC 5849: The OAuth 1.0 Protocol. *IETF Requests for Comment*. Retrieved from <https://tools.ietf.org/html/rfc5849>
- Haraway, D. (1991). A CYBORG MANIFESTO SCIENCE, TECHNOLOGY, AND SOCIALIST-FEMINISM IN THE LATE TWENTIETH CENTURY. In *Simians, Cyborgs, and Women* (pp. 149–181). New York: Routledge.
- Hayles, K. (1999). *How we became posthuman : virtual bodies in cybernetics, literature, and informatics*. Chicago Ill.: University of Chicago Press. Retrieved from [http://www.worldcat.org/title/how-we-became-posthuman-virtual-bodies-in-cybernetics-literature-and-informatics/oclc/39539341&referer=brief\\_results](http://www.worldcat.org/title/how-we-became-posthuman-virtual-bodies-in-cybernetics-literature-and-informatics/oclc/39539341&referer=brief_results)
- Heidegger, M. (1993). The Question Concerning Technology. In D. F. Krell (Ed.), *Basic Writings* (pp. 307–342). New York: HarperCollins.
- Henry, B. (2014). Imaginaries of the global age. “Golem and others” in the post-human condition. *Politica & Societa*, 2, 221–246.
- Hern, A. (2015, August 24). Twitter blocks access to political transparency organisation Politwoops. *The Guardian*. Retrieved from <http://www.theguardian.com/technology/2015/aug/24/twitter-blocks-access-political-transparency-organisation-politwoops>

- Heron, M. J., Belford, P., & Goker, A. (2014). Sexism in the circuitry. *ACM SIGCAS Computers and Society*, 44(4), 18–29. <http://doi.org/10.1145/2695577.2695582>
- Herring, S. C. (1999). The Rhetorical Dynamics of Gender Harassment On-Line. *The Information Society*. <http://doi.org/10.1080/019722499128466>
- Irani, L., & Silberman, M. (2013). Turkopticon: Interrupting worker invisibility in amazon mechanical turk. *Proceedings of the SIGCHI Conference on ...*, 611–620. <http://doi.org/10.1145/2470654.2470742>
- Jackson, S. (2014). Rethinking Repair. In T. Gillespie, P. Boczkowski, & K. A. Foot (Eds.), *Media Technologies: Essays on Communication, Materiality, and Society*. Cambridge: The MIT Press.
- James, R. (2015, January 27). Cloudy Logic. *The New Inquiry*. Retrieved from <http://thenewinquiry.com/essays/cloudy-logic/>
- Jervell, E. (2014, June 13). For This Author, 10,000 Wikipedia Articles Is a Good Day's Work. *The Wall Street Journal*. Retrieved from [http://www.wsj.com/articles/for-this-author-10-000-wikipedia-articles-is-a-good-days-work-1405305001?mod=\\_newsreel\\_4](http://www.wsj.com/articles/for-this-author-10-000-wikipedia-articles-is-a-good-days-work-1405305001?mod=_newsreel_4)
- Jurgenson, N. (2012). When Atoms Meet Bits: Social Media, the Mobile Web and Augmented Revolution. *Future Internet*, 4, 83–91. Retrieved from [www.mdpi.com/1999-5903/4/1/83](http://www.mdpi.com/1999-5903/4/1/83)
- Jurgenson, N. (2014). View from Nowhere. *The New Inquiry*, October. Retrieved from <http://thenewinquiry.com/essays/view-from-nowhere/>
- Keen, A. (2007). *The Cult of the Amateur: How Today's Internet is Killing Our Culture*. Doubleday Business. Retrieved from <http://www.amazon.com/dp/0385520808>
- Kelty, C. (2008). *Two Bits: The Cultural Significance of Free Software*. Durham: Duke University Press.
- Kennedy, K. (2010). Textual Machinery: Authorial Agency and Bot-Written Texts in Wikipedia. In M. Smith & B. Warnick (Eds.), *The Responsibilities of Rhetoric* (pp. 303–309). Long Grove: Waveland Press. Retrieved from <http://surface.syr.edu/wp/1>
- Kirschenbaum, M. G. (2008). New Media and the Forensic Imagination. *Media*, (60), 1–9. Retrieved from <http://books.google.com/books?hl=en&lr=&id=CT0oPmrciAC&pgis=1>
- Konieczny, P. (2009). Wikipedia: Community or Social Movement??. *Interface: A Journal for and about Social ...*. Retrieved from <http://interfacejournal.nuim.ie/wordpress/wp-content/uploads/2010/11/Interface-1-2-pp212-232-Konieczny.pdf>

- Konieczny, P. (2010). Adhocratic Governance in the Internet Age: A Case of Wikipedia. *Journal of Information Technology & Politics*, 7(4), 263–283. <http://doi.org/10.1080/19331681.2010.489408>
- Kostakis, V. (2010, March 12). Peer governance and Wikipedia: Identifying and understanding the problems of Wikipedia's governance. *First Monday*. Retrieved from <http://firstmonday.org/ojs/index.php/fm/article/view/2613/2479>
- Kriplean, T., Beschastnikh, I., & McDonald, D. (2008). Articulations of wikiwork: uncovering valued work in wikipedia through barnstars. In *Proc CSCW 2008* (pp. 47–56). ACM. Retrieved from <http://dub.washington.edu/djangosite/media/papers/tmpZ77p1r.pdf>
- Kriplean, T., McDonald, D. W., Beschastnikh, I., & Golder, S. A. (2007). Community, consensus, coercion, control: cs\*w or how policy mediates mass participation. In *Proc GROUP 2007* (pp. 167–176). ACM. <http://doi.org/10.1145/1316624.1316648>
- Kunda, G. (1992). *Engineering Culture: Control and Commitment in a High-Tech Corporation*. Philadelphia: Temple University Press.
- Kuznetsov, S. (2006). Motivations of contributors to Wikipedia. *SIGCAS Comput. Soc.*, 36(2), 1. <http://doi.org/10.1145/1215942.1215943>
- Lam, S. (Tony) K., Uduwage, A., Dong, Z., Sen, S., Musicant, D. R., Terveen, L., & Riedl, J. (2011). WP:clubhouse? An Exploration of Wikipedia's Gender Imbalance. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration - WikiSym '11* (p. 1). New York, New York, USA: ACM Press. <http://doi.org/10.1145/2038558.2038560>
- Lam, S. K., Karim, J., & Riedl, J. (2010). The effects of group composition on decision quality in a social production community. In *Proceedings of the 16th ACM international conference on Supporting group work - GROUP '10* (p. 55). New York, New York, USA: ACM Press. <http://doi.org/10.1145/1880071.1880083>
- Laniado, D., Tasso, R., Kaltenbrunner, A., Milano, P., & Volkovich, Y. (2011). When the Wikipedians Talk : Network and Tree Structure of Wikipedia Discussion Pages. In *Proceedings of the 5th International Conference on Weblogs and Social Media* (pp. 177–184).
- Latour, B. (1987). *Science in Action: How to Follow Scientists and Engineers Through Society*. Cambridge Mass.: Harvard University Press.
- Latour, B. (1988). Mixing Humans and Nonhumans Together: The Sociology of a Door-Closer. *Social Problems*, 35(3), 298–310. Retrieved from

<http://www.jstor.org/stable/800624>

- Latour, B. (1992). Where are the missing masses? The sociology of a few mundane artifacts. In J. Law & W. E. Bijker (Eds.), *Shaping Technology Building Society* (pp. 225–258). Cambridge Mass.: The MIT Press. Retrieved from <http://www-personal.si.umich.edu/~rfrost/courses/MatCult/content/Latour.pdf>
- Latour, B. (1996). *Aramis, or, The Love of Technology*. Cambridge Mass.: Harvard University Press.
- Latour, B. (1999a). Circulating Reference: Sampling the Soil in the Amazon Forest. In *Pandora's Hope* (p. 79). Cambridge, Mass.: Harvard University Press.
- Latour, B. (1999b). *Pandora's Hope: Essays on the Reality of Science Studies*. Cambridge Mass.: Harvard University Press.
- Latour, B. (2004). Why Has Critique Run out of Steam? From Matters of Fact to Matters of Concern. *Critical Inquiry*. <http://doi.org/10.1086/421123>
- Latour, B. (2009). *The Making of Law: An Ethnography of the Conseil d'Etat*. Cambridge: Polity.
- Latour, B., & Woolgar, S. (1979). *Laboratory Life: The Social Construction of Scientific Facts*. Beverly Hills: Sage Publications.
- 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>
- Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge: Cambridge University Press.
- Law, J. (2008). Actor Network Theory and Material Semiotics. In B. S. Turner (Ed.), *The New Blackwell Companion to Social Theory* (pp. 141–158). Oxford: Blackwell. Retrieved from [http://ewasteschools.pbworks.com/w/file/attach/45601674/Law\\_John\\_2008\\_On\\_Sociology\\_and\\_STS.pdf](http://ewasteschools.pbworks.com/w/file/attach/45601674/Law_John_2008_On_Sociology_and_STS.pdf)
- Law, J., & Bijker, W. (1994). *Shaping Technology / Building Society: Studies in Sociotechnical Change*. Cambridge, Mass.: The MIT Press. Retrieved from <http://www.amazon.com/dp/0262521946>
- Law, J., & Mol, A. (1995). Notes on Materiality and Sociality. *The Sociological Review*, 43, 274–294.

- Lessig, L. (1999). *Code and Other Laws of Cyberspace*. New York: Basic Books.
- Lessig, L. (2006). *Code 2.0*. New York: Basic Books.
- Lewis, N. (2014, October 15). IS BIG DATA SQUISHING OUR HUMANITY? *Spiked*. Retrieved from <http://www.spiked-online.com/newsite/article/is-big-data-squishing-our-humanity/16022#.VXn-Hs9VhBc>
- Lih, A. (2009). The Wikipedia revolution: How a bunch of nobodies created the world's greatest encyclopedia. Retrieved from [http://scholar.google.com/scholar?hl=en&q=wikipedia+meetups&btnG=&as\\_sdt=1%2C5&as\\_sdt=#1](http://scholar.google.com/scholar?hl=en&q=wikipedia+meetups&btnG=&as_sdt=1%2C5&as_sdt=#1)
- Lindtner, S., & Li, D. (2012). Created in China: the makings of China's hackerspace community. *Interactions*, 19(6), 18–22.
- Lundblad, N. (2007). e-Exclusion and Bot Rights: Legal aspects of the robots exclusion standard for public agencies and other public sector bodies with Swedish examples. *First Monday*, 12(8). Retrieved from <http://ojs-prod-lib.cc.uic.edu/ojs/index.php/fm/article/view/1974>
- MacKenzie, D. (1996). Marx and the Machine. In *Knowing Machines: Essays on Technical Change* (pp. 23–48). Cambridge, Mass: The MIT Press.
- Mackenzie, D. (2006). *An Engine, Not a Camera: How Financial Models Shape Markets*. Cambridge, Mass.: The MIT Press.
- Marcus, G. E., & Cushman, D. (1982). Ethnographies as Texts. *Annual Review of Anthropology*. <http://doi.org/10.1146/annurev.an.11.100182.000325>
- Marwick, A., & Miller, R. (2014). *Online Harassment, Defamation, and Hateful Speech: A Primer of the Legal Landscape*. New York.
- Marx, K. (1973). The Fragment on Machines. In *The Grundrisse* (pp. 690–712). London: Penguin. Retrieved from <http://thenewobjectivity.com/pdf/marx.pdf>
- Mathew, A., & Cheshire, C. (2010). The New Cartographers: Trust and Social Order within the Internet Infrastructure. Retrieved from <http://papers.ssrn.com/abstract=1988216>
- Matias, J. N., Johnson, A., Boesel, W. E., Keegan, B., Friedman, J., & DeTar, C. (2015). Reporting, Reviewing, and Responding to Harassment on Twitter. Retrieved from <http://papers.ssrn.com/abstract=2602018>
- Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software*

- Engineering and Methodology*, 11(3), 309–346. <http://doi.org/10.1145/567793.567795>
- Mody, C., & Kaiser, D. (2008). Scientific Training and the Creation of Scientific Knowledge. In *The Handbook of Science and Technology Studies* (Third Edit, pp. 377–402). Cambridge: The MIT Press.
- Morgan, J., Gilbert, M., McDonald, D., & Zachry, M. (2014). Editing Beyond Articles: Diversity & Dynamics of Teamwork in Open Collaborations. *Proceedings of CSCW 2014*. Retrieved from [http://dub.uw.edu/djangosite/media/papers/csw2014\\_wikiprojects\\_final\\_archival.pdf](http://dub.uw.edu/djangosite/media/papers/csw2014_wikiprojects_final_archival.pdf)
- Morozov, E. (2013). *To Save Everything Click Here: The Folly of Technological Solutionism*. New York: PublicAffairs.
- Morozov, E. (2014, October 22). The Real Privacy Problem. *MIT Technology Review*. Retrieved from <http://www.technologyreview.com/featuredstory/520426/the-real-privacy-problem/>
- Müller-Birn, C., Dobusch, L., & Herbsleb, J. D. (2013). Work-to-rule. In *Proceedings of the 6th International Conference on Communities and Technologies - C&T '13* (pp. 80–89). New York, New York, USA: ACM Press. <http://doi.org/10.1145/2482991.2482999>
- Murchison, J. (2010). Common Ethnographic Conventions. In *Ethnography Essentials: Designing, Conducting, and Presenting Your Research*. New York: John Wiley & Sons.
- Nader, L. (1969). Up the Anthropologist - Perspectives Gained from Studying Up. In *Reinventing Anthropology* (pp. 284–311).
- Nakamura, L. (2002). *Cybertypes: Race, Ethnicity and Identity on the Internet*. London: Routledge.
- Neff, G., & Stark, D. (2003). Permanently Beta. In *Society Online: The Internet in Context* (pp. 179–188). New York: Sage.
- Negroponte, N. (1995). *Being digital*. New York: Knopf. Retrieved from [http://www.worldcat.org/title/being-digital/oclc/31816356&referer=brief\\_results](http://www.worldcat.org/title/being-digital/oclc/31816356&referer=brief_results)
- Niederer, S., & Van Dijck, J. (2010). Wisdom of the crowd or technicity of content? Wikipedia as a sociotechnical system. *New Media & Society*, 12(8), 1368–1387. <http://doi.org/10.1177/1461444810365297>
- Novek, J. (2002). It, Gender, and Professional Practice: Or, Why an Automated Drug Distribution System Was Sent Back to the Manufacturer. *Science, Technology & Human Values*, 27(3), 379–403.

- O'Hara, D., & Mason, L. R. (2012, March 30). How bots are taking over the world. *The Guardian*. Retrieved from <http://www.theguardian.com/commentisfree/2012/mar/30/how-bots-are-taking-over-the-world>
- O'Leary, M. B., Cummings, J. N., & O'Leary, M. B. (2007). The spatial, temporal, and configurational characteristics of geographic dispersion in teams. *MIS Quarterly*, 31(3), 433–452. <http://doi.org/Article>
- O'Reilly, T. (2011). Government as a Platform. *innovations*2, 6(1), 13–40.
- O'Reilly, T. (2013). Open data and algorithmic regulation. In *Beyond transparency: Open data and the future of civic innovation* (pp. 289–300).
- Oreg, S., & Nov, O. (2008). Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values. *Computers in Human Behavior*, 24(5), 2055–2073. <http://doi.org/10.1016/j.chb.2007.09.007>
- Orlikowski, W. J., & Scott, S. V. (2008). 10 Sociomateriality: Challenging the Separation of Technology, Work and Organization. *The Academy of Management Annals*, 2(1), 433–474. <http://doi.org/10.1080/19416520802211644>
- Orr, J. (1996). *Talking about machines : an ethnography of a modern job*. Ithaca: ILR Press.
- Oudsshoorn, N., & Pinch, T. (2003). *How Users Matter: The Co-Construction of Users and Technology*. Cambridge: The MIT Press.
- Papacharissi, Z. (2002). The virtual sphere: The internet as a public sphere. *New Media Society*, 4(1), 9–27. <http://doi.org/10.1177/14614440222226244>
- Papacharissi, Z. (2014). Affective publics : sentiment, technology, and politics.
- Pariser, E. (2012). *The Filter Bubble. The Filter Bubble*. New York: Penguin.
- Pearce, K. (2014). TWO CAN PLAY AT THAT GAME: SOCIAL MEDIA OPPORTUNITIES IN AZERBAIJAN FOR GOVERNMENT AND OPPOSITION. *Demokratizatsiya*, 22(1), 39–66. Retrieved from <http://search.proquest.com/docview/1528366356/abstract?accountid=14496>
- Poster, M. (2001). *Cyberdemocracy: Internet and the public sphere* (pp. 259–271). Malden, MA: Blackwell Publishing. Retrieved from [http://books.google.com/books?hl=en&lr=&id=h3YjCcGfzLgC&oi=fnd&pg=PA259&dq=internet+“public+sphere”&ots=4Ju3MZmChr&sig=YriBApI\\_JX3ACGRRWYVMx-z8aDY](http://books.google.com/books?hl=en&lr=&id=h3YjCcGfzLgC&oi=fnd&pg=PA259&dq=internet+“public+sphere”&ots=4Ju3MZmChr&sig=YriBApI_JX3ACGRRWYVMx-z8aDY)



- Preece, J., & Shneiderman, B. (2009). The Reader-to-Leader Framework: Motivating Technology-Mediated Social Participation. *AIS Transactions on Human-Computer Interaction*, 1(1), 13–32. Retrieved from <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1004&context=thci>
- Reagle, J. (2008). *In Good Faith: Wikipedia Collaboration and the Pursuit of the Universal Encyclopedia*.
- Reagle, J. (2010). *Good Faith Collaboration: The Culture of Wikipedia*. Cambridge, Mass.: The MIT Press.
- Ribes, D., Jackson, S., Geiger, R. S., Burton, M., Finholt, T., Geiger, S., ... Finholt, T. (2013). Artifacts that organize: Delegation in the distributed organization. *Information and Organization*, 23(1), 1–14. <http://doi.org/10.1016/j.infoandorg.2012.08.001>
- Rossi, A., Gaio, L., den Besten, M., & Dalle, J.-M. M. (2010). Coordination and Division of Labor in Open Content Communities: The Role of Template Messages in Wikipedia. In *2010 43rd Hawaii International Conference on System Sciences* (pp. 1–10). IEEE. <http://doi.org/10.1109/HICSS.2010.122>
- Sankin, A. (2014). Bots have conquered Wikipedia—and that’s a good thing. *TheDailyDot*. Retrieved from <http://www.dailydot.com/technology/wikipedia-bots-editing-rambot/>
- Saxenian, A. (2006). *The New Argonauts: Regional Advantage in a Global Economy*. Harvard University Press. Retrieved from [http://books.google.com/books?id=djgO\\_Thn6dUC&pgis=1](http://books.google.com/books?id=djgO_Thn6dUC&pgis=1)
- Schneider, J., Passant, A., & Breslin, J. G. (2011). Understanding and improving Wikipedia article discussion spaces. *Proceedings of the 2011 ACM Symposium on Applied Computing - SAC '11*, 808–813. <http://doi.org/10.1145/1982185.1982358>
- Schneider, J., Passant, A., & Decker, S. (2012). Deletion Discussions in Wikipedia: Decision Factors and Outcomes. In *WikiSym 2012*. ACM. Retrieved from <http://www.wikisym.org/ws2012/p12wikisym2012.pdf>
- Seaver, N. (2013). *Knowing Algorithms. Media in Transition 8*. Retrieved from <http://nickseaver.net/papers/seaverMIT8.pdf>
- Shapin, S. (1992). Why the Public Ought to Understand Science-in-the-Making. *Public Understanding of Science*, 1(1), 27–30. Retrieved from [http://dash.harvard.edu/bitstream/handle/1/3353815/Shapin\\_Why.pdf?sequence=1](http://dash.harvard.edu/bitstream/handle/1/3353815/Shapin_Why.pdf?sequence=1)
- Shapin, S., & Schaffer, S. (1985). *Leviathan and the Air-Pump*. Princeton: Princeton University Press.

- Stacey, J., & Suchman, L. (2012). Animation and Automation – The liveliness and Labours of Bodies and Machines. *Body & Society*, 18, 1–46.
- Stallman, R. (2013, September). Why Free Software Is More Important Now Than Ever Before. *Wired*.
- Star, S. L. (1999). The Ethnography of Infrastructure. *American Behavioral Scientist*, 43(3), 377–391. <http://doi.org/10.1177/00027649921955326>
- Star, S. L., & Griesemer, J. R. (1989). Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social Studies Of Science*, 19(3), 387–420. <http://doi.org/10.1177/030631289019003001>
- Star, S. L., & Ruhleder, K. (1996). Steps toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research*, 7(1), 111–134.
- Star, S. L., & Strauss, A. (1999). Layers of Silence , Arenas of Voice : The Ecology of Visible and Invisible Work. *Computer Supported Cooperative Work*, 8, 9–30. <http://doi.org/10.1023/A:1008651105359>
- Strauss, A. (1985). Work and the division of labor. *The Sociological Quarterly*, 26(1), 1–19. <http://doi.org/10.1111/j.1533-8525.1985.tb00212.x>
- Stvilia, B., Twidale, M. B., Gasser, L., Smith, L. C., Stvilia, B., Twidale, M. B., & Stvilia, B. (2008). Information quality work organization in wikipedia. *Journal of the American Society for Information Science and Technology*, 59(6), 983–1001. <http://doi.org/10.1002/asi.20813>
- Suchman, L. (1995). Representations of work: Making work visible. *Association for Computing Machinery. Communications of the ACM*, 38(Sep), 56. <http://doi.org/http://doi.acm.org/10.1145/223248.223263>
- Suchman, L. (1996). Supporting Articulation Work. In R. Kling (Ed.), *Computerization and Controversy: Value Conflicts and Social Choices* (pp. 407–425). San Francisco: Morgan Kaufmann.
- Suchman, L. (2007). *Human-Machine Reconfigurations: Plans and Situated Actions*. Cambridge: Cambridge University Press.
- Suh, B., Convertino, G., Chi, E., & Pirolli, P. (2009). The Singularity is Not Near: Slowing Growth of Wikipedia. In *Proc WikiSym 2009*.
- Surowiecki, J. (2004). *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. Doubleday.

- Takhteyev, Y. (2012). *Coding Places: Software Practice in a South American City (Acting with Technology)*. The MIT Press. Retrieved from <http://www.amazon.com/Coding-Places-Software-Practice-Technology/dp/0262018071>
- Tapscott, D., & Williams, A. D. (2006). *Wikinomics: How Mass Collaboration Changes Everything*. Portfolio Hardcover. Retrieved from <http://www.amazon.com/dp/1591841380>
- Taraborelli, D., & Ciampaglia, G. L. (2010). Beyond notability. Collective deliberation on content inclusion in Wikipedia. In *Fourth IEEE International Conference on SelfAdaptive and SelfOrganizing Systems Workshops SASOW 2010* (pp. 122–125). IEEE. <http://doi.org/10.1109/SASOW.2010.26>
- Tkacz. (2013). Open sesame. *Aeon*. Retrieved from <http://www.aeonmagazine.com/world-views/nathaniel-tkacz-open-source-government/>
- Tkacz, N. (2015). *Wikipedia and the Politics of Openness*. Chicago: University of Chicago Press.
- Toffler, A. (1970). *Future Shock*. New York: Random House.
- Tufecki, Z. (2015). How Facebook’s Algorithm Suppresses Content Diversity (Modestly) and How the Newsfeed Rules Your Clicks. Retrieved from <https://medium.com/message/how-facebook-s-algorithm-suppresses-content-diversity-modestly-how-the-newsfeed-rules-the-clicks-b5f8a4bb7bab>
- Turner, F. (2006). *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. Chicago: University Of Chicago Press.
- Van Dijck, J., & Nieborg, D. (2009). Wikinomics and its discontents: a critical analysis of Web 2.0 business manifestos. *New Media & Society*, *11*(5), 855–874. <http://doi.org/10.1177/1461444809105356>
- van Liere, D., & Fung, H. (2011). *Editor Trends Study*. Retrieved from [https://strategy.wikimedia.org/wiki/Editor\\_Trends\\_Study](https://strategy.wikimedia.org/wiki/Editor_Trends_Study)
- Vaughan, D. (1997). *The Challenger Launch Decision: Risky Technology, Culture, and Deviance at NASA*. Chicago: University Of Chicago Press.
- Viegas, F., Wattenberg, M., Kriss, J., & van Ham, F. (2007). Talk Before You Type: Coordination in Wikipedia. In *Proc HICSS 2007*. Retrieved from <http://dl.acm.org/citation.cfm?id=1255643>
- von Ahn, L., Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008). reCAPTCHA: human-based character recognition via Web security measures. *Science (New York,*

- N.Y.), 321(5895), 1465–1468. <http://doi.org/10.1126/science.1160379>
- Vygotsky, L. (1968). *Thought and Language*. Cambridge: The MIT Press.
- Waldby, C. (2004). The Instruments of Life: Frankenstein and Cyberculture. In D. Tofts, A. Jonson, & A. Cavallaro (Eds.), *Prefiguring Cyberculture: An Intellectual History* (pp. 28–38). Cambridge: The MIT Press.
- Wattenberg, M., Viegas, F., & McKeon, M. (2007). The Hidden Order of Wikipedia. In *Proc OCSC 2007* (pp. 445–454). Retrieved from [http://dx.doi.org/10.1007/978-3-540-73257-0\\_49](http://dx.doi.org/10.1007/978-3-540-73257-0_49)
- Weber, M. (1968). The Bureaucracy. In *Economy and Society* (pp. 956–1005). Los Angeles: University of California Press.
- Weizenbaum, J. (1966). ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine. *Communications of the ACM*, 9(1), 36–45. Retrieved from <https://dx.doi.org/10.1145%2F365153.365168>
- Welser, H. T., Cosley, D., Kossinets, G., Lin, A., Dokshin, F., Gay, G., & Smith, M. (2011). Finding social roles in Wikipedia. In *Proceedings of the 2011 iConference on - iConference '11* (pp. 122–129). <http://doi.org/10.1145/1940761.1940778>
- Williams, R. (1977). *Marxism and Literature*. Oxford: Oxford University Press.
- Winner, L. (1986). Do Artifacts Have Politics? In *The whale and the reactor: a search for limits in an age of high technology*. (pp. 19–39). Chicago: University of Chicago Press.
- Wittgenstein, L. (1958). *Philosophical Investigations*. Oxford: Basil Blackwell.
- Wong, J., & Hong, J. (2008). What do we “mashup” when we make mashups? In *Proceedings of the 4th international workshop on End-user software engineering - WEUSE '08* (pp. 35–39). New York, New York, USA: ACM Press. <http://doi.org/10.1145/1370847.1370855>
- Yang, H. L., & Lai, C. Y. (2010). Motivations of Wikipedia content contributors. *Computers in Human Behavior*, 26(6), 1377–1383. <http://doi.org/10.1016/j.chb.2010.04.011>
- Zammuto, R. F., Griffith, T. L., Majchrzak, A., Dougherty, D. J., & Faraj, S. (2007). Information Technology and the Changing Fabric of Organization. *ORGANIZATION SCIENCE*, 18(5), 749–762. <http://doi.org/10.1287/orsc.1070.0307>
- Zeifman, I. (2013). *Bot traffic is up to 61.5% of all website traffic*. Retrieved from <https://www.incapsula.com/blog/bot-traffic-report-2013.html>