Content Management Systems, Bittorrent Trackers, and Large-Scale Rhetorical Genres: Analyzing Collective Activity in Participatory Digital Spaces

Justin Lewis

Abstract
Scholars of rhetoric and writing have long recognized the mediated nature of rhetorical action. From Plato’s early indictments of writing as enemy of memoria to Burke’s recognition of instrumental causes to recent analyses of digital mediation, the study of meaning-making refuses one-to-one, transparent theories of communication, instead recognizing that there is more to rhetorical action than humans. This article follows the trail of Haas, Swarts, and others arguing that analyses of mediation uncover much about human motives, digital communities, and rhetorical action. I argue that technologies often function as rhetorical genres, providing what Miller characterizes as “typified rhetorical actions based in recurrent situations” that occur in uniquely digital spaces. Working from sites of participatory archival creation and curation, I argue that invisible rhetorical genres operating at macroscopic levels of scale are central to shaping individual and communal activity in sites of distributed social production. To support this claim, I investigate two applications—a content management system called Gazelle and a bittorrent tracker called Ocelot—to demonstrate how largely invisible server-side software shapes rhetorical action, circumscribes individual agency, and cultivates community identity in sites of participatory
archival curation. By articulating content management systems and other macroscopic software as rhetorical genres, I hope to extend nascent investigations into the medial capacities of digital tools that shape our collective digital experience.

**Keywords**

Introduction

Technologies shape human action. Sometimes those technologies are everyday instruments that have been operationalized, made invisible in their role as mediators of everyday activity. As Heidegger (1977), working from Aristotle, points out, technologies effect (aition) human action, they are “responsible for something else” (p. 7, emphasis in the original). In myriad ways, technologies exert agency, shaping human activity by mediating our interactions with the lifeworld. From this perspective, we can‘t know the world except through technology. Our prostheses circumscribe our activity, making lifework happen through complex coordination that is often compounded by interleaved layers of human–technology action.

Scholars of rhetoric and writing have long recognized as much. From Plato’s early indictments of writing as enemy of memoria (Hackforth, 1952) to Burke’s (1969) recognition of instrumental causes to recent analyses of digital mediation (Haas, 1996; Spinuzzi, 2008; Swarts, 2008; Van Ittersum & Ching, 2013), the study of meaning-making refuses one-to-one, transparent theories of communication, instead recognizing that there is more to rhetorical action than the human–human dyad. This article follows the trail of Haas, Swarts, and others arguing that analyses of mediation uncover much about human motives, digital communities, and rhetorical action. I contend that technologies often function as rhetorical genres, providing what Miller characterizes as “typified rhetorical actions based in recurrent situations” that occur in uniquely digital spaces (Miller, 1994, p. 159). In so doing, technologies affect our experience of the world, transforming our intentions, mediating our motives, and anchoring us in contexts of use. Working from sites of participatory archival creation and curation, I argue that invisible rhetorical genres operating at macroscopic levels of scale are central to shaping individual and communal activity in digital spaces. To support this claim, I investigate two applications—a content management system (CMS) called Gazelle and a bittorrent tracker called Ocelot—to demonstrate how largely invisible server-side software shapes rhetorical action, coordinates individual actions, and cultivates communal inclusion in sites of participatory archival curation. Understanding technologies as rhetorical genres also allows researchers to connect work in Writing Studies to other technosocial fields, notably User Experience Design (UX). By articulating
CMSs and other macroscopic software as rhetorical genres, I hope to bring Rhetorical Genre Studies, technology studies, and UX into closer conversation with one another, fostering analysis of technologically mediated rhetorical action and design of technological mediators that enhance user experiences.

**Research Site**

To analyze the rhetorical and generic capacities of CMSs and other server-side software, I investigated a private, invite-only bittorrent community called Question.cd. Boasting nearly 150,000 registered users and approximately 2,000,000 individual torrent files, Question.cd is one of the more popular general purpose piratical communities on the internet. Question.cd’s servers connect users via the bittorrent protocol, allowing them to pirate various media, including books, music, applications, and eLearning materials. As of September 2014, Question.cd’s archive included 685,686 unique artists, 827,871 music releases, and 24,989 ebooks. Characterized by user uploaded materials, Question.cd functions much like Wikipedia or Archive.org inasmuch, as it relies on its users to expand and curate the archive. In this way, Question.cd is considered what Huvila (2008) calls a “participatory archive” or a space that is radically user-oriented and whose curation is decentralized and distributed. Participatory archives are also characterized by contextualization of archive items. Much like a Wikipedia “Talk” page, Question.cd users annotate and contextualize archival elements uploaded by themselves or by others. Taken together, decentralized curation, radical user orientation, and archival contextualization have allowed Question.cd users to build an archive of staggering size at a rate that rivals new page creation in the English version of Wikipedia.

I chose Question.cd to explore the rhetorical and generic qualities of software like CMSs because Question.cd community members are largely responsible for developing and maintaining both the Gazelle CMS and Ocelot tracker. Because these technologies were born from communal desire, subject to cultural–historical forces and meet the recurring needs of a specific group, they are ripe for further investigation as macroscopic rhetorical genres; further, because the software development process of both applications was conducted in community forums on Question.cd, negotiations over function, form, and capabilities were visible and preserved for future study. Finally, because of my own epistemic privilege as a Question.cd community member, my experiences on the site provided useful experiential evidence for this project.

**Macroscopic Tools: Coordinating Large-Scale Activity in Digital Spaces**

Macroscopic layers of activity involve the ways “workers, work communities, cultures, and societies understand structure, collaborate on, and execute their
cooperative enterprises” (Spinuzzi, 2003). Macroscopic activity could be considered the contextual or organizational layer in which large-scale activity occurs and constitutes the broader ecologies of collective action. While mesoscopic actions are ripe for investigations into shorter, more targeted investigations, macroscopic analysis of mediating technologies in participatory archives require a longitudinal orientation that traces the cultural–historical genesis of community goals and motives over time. The goal of this section will be to identify and analyze two mediating technologies that function as rhetorical genres in the macroscopic layer whose influence and importance extend beyond the individual goals of mesoscopic action or microscopic operation. To that end, I will first consider the importance of cultural–historical tracing when dealing with macroscopic rhetorical genres. This section will posit the centrality of compound mediation when considering interlocking activity systems or genre systems. Next, I will explore the role of CMSs and bittorrent trackers diachronically, paying close attention to the historical needs and cultural desires that gave rise to such software. Third, I will explore two macroscopic rhetorical genres that structure the activity system in participatory archival environments. The first is Gazelle, a CMS designed to increase site efficiency, incorporate community-building tools, and provide a fluid interface for user-initiated site navigation and utilization. The second is Ocelot, a lightweight bittorrent tracker that remains invisible to site users but facilitates content sharing in novel ways.

Theorizing Macroscopic Tools as Rhetorical Genres

Before exploring the Gazelle CMS as a macroscopic mediating tool that organizes genre ecologies in the mesoscopic layer, we should first explore how something like a bundle of .php code might function as a digital rhetorical genre. If you dig deep in the history of genre studies, you are eventually led to the work of Bakhtin, Medvedev, and Voloshinov. As philosophers of literature and language, the so-called Bakhtin circle reworked the notion of genre in the early 20th century, providing some key insights into the ways that genres function rhetorically at macroscopic levels of scale. In The Formal Method in Literary Scholarship, Bakhtin and Medvedev (1978) argue that genres are not static textual artifacts but instead of culturally grounded, historically bound means of “seeing and conceptualizing reality” (p. 134). In later work, Bakhtin would further explore how texts and speech are shot through with social motives. Theorizing the utterance in “The Problem of Speech Genres,” Bakhtin highlights the importance of addressivity and the socialized nature of communication. Noting that, “from the very beginning, the utterance is constructed while taking into account possible responsive reactions, for whose sake, in essence, it is actually created” he eventually cedes, “From the very beginning the speaker expects a response from them, an active responsive understanding. The entire utterance is constructed, as it were, in anticipation of encountering this
response” (Bakhtin, Holquist, McGee, & Emerson, 1986, p. 94). Bawarshi later links Bakhtin’s notion of addressivity to generic form, arguing that

[T]he speaker’s very conception of the addressee is mediated by genre, because each genre embodies its own typical conception of the addressee. In fact, the very word and its relation to other words is also mediated by speech genres. (Bawarshi, 2000, p. 348)

These speech genres are not, however, universal inasmuch as they are intensely rhetorical, being attuned to their prospective audiences, contexts, and chronotopic location. So, while Bakhtin’s theory of genre is expansive, it also recognizes that specific genres are bound by their cultural–historical genesis and present rhetorical situation(s). Reflecting on Bakhtin’s notion of genre, Morson and Emerson agree:

Each genre implies a set of values, a way of thinking about kinds of experience, and an intuition about the appropriateness of applying the genres in any given context. An enormous amount of unfinalized cognitive content is acquired each time we learn a new kind of social activity with its attendant genres, content whose very nature has remained largely unexamined. (Morson, & Emerson, 1990, pp. 291–292)

Morson and Emerson’s emphasis on the “largely unexamined” nature of the genres of social activity is particularly apropos to grasping the ways that rhetorical genres structure macroscopic activity. As Spinuzzi, relying on Leont’ev, argues in Tracing Genres Through Organizations, activity at the macroscopic level is largely invisible and operates without conscious acknowledgment on the part of individuals circulating through the system. Genres at this layer play important stabilizing roles in sustaining entire activity systems, allowing the collective action of numerous subordinate mesoscopic and microscopic actions and operations to aggregatively move the activity along; however, individuals are not typically cognizant of macroscopic activity beyond recognizing the shared social motive or object rendered. As such, macroscopic rhetorical genres are much more akin to Miller’s original formulation that genres respond to exigence, or the “socially objectified motive” that defines rhetorical situations (Miller, 1994, p. 31) or Bakhtin’s notion that genres embody “social memory.” What is perhaps most interesting about the largely invisible, socially inflected nature of mediating tools at macroscopic scale in this study is their literal invisibility: the Gazelle CMS renders interfaces, navigation, and tools that users engage everytime they visit Question.cd; yet, most users rarely contribute to the building, testing, and production of the CMS itself. While collages, requests, and tagging systems draw on user generated data to facilitate specific forms of user directed, goal-oriented action in the Question.cd ecosystem, Gazelle organizes mesoscopic tools through interfaces, navigation tools, and information architectures.
While the substantive function of macroscopic rhetorical genres like Gazelle meet recurrent needs of site users whose activity is communally directed, the structure and technological affordances of the software also play a role in articulating server-side software as rhetorical genres. As Yates and Orlikowski (1992) point out, the substantive action of any object can span across forms. For example, a letter of recommendation performs the social action of providing accolades; however, other rhetorical actions can perform the same function, namely epideictic speeches. We only know a recommendation letter is a recommendation letter because of a marriage of substance and form. By including the address, salutation, body, and closing on a sheet of paper might someone categorize a recommendation letter as such. Genres comprise both social actions, or substance, and structure, or form.

Amending Yates and Orlikowski’s work and following the tack of Miller and Shepherd (Miller & Shepherd, 2009), I want to also draw on technological affordance or medium as the third element that comprises digital rhetorical genres. In “Questions for Genre Theory from the Blogosphere,” the authors note that affordances of the medium of the internet are not determinative but must interact with social need before the genre comes into being. In other words, technological affordances of the medium of the internet function maieutically, “coaching—or coaxing—into being a latent social motivation that, when available, is instantly recognizable to large numbers of people” (Miller & Shepherd, 2009, p. 282). As Bernard Stiegler (1998) observed some time ago, technogenesis outpaces sociogenesis.

In short, by paying attention to the technological affordances of the medium while also tracing the substantive and structural characteristics of macroscopic digital tools, I hope to reveal how we might productively think of server-side applications as large-scale rhetorical genres that shape human action are rooted in cultural–historical activity and employ social data to structure the collective action of participatory archival creation. Further, thinking about the substance, structure, and affordances of software from a generic perspective also reveals a lot about the aesthetic and functional experiences users expect from the macroscopic technologies that undergird digital community.

**Compound Mediation, Activity Systems, and Genre Ecologies**

In a chapter from Bazerman and Russell’s edited collection *Writing Selves, Writing Societies*, Spinuzzi describes compound mediation as “the ways that people habitually coordinate sets of artifacts to mediate or carry out their activities” (Bazerman, 1994, p. 98). The notion of compound mediation is present in all activity systems, and Question.cd is no different. Yet, users do not act apart from organizational contexts; rather, they are imbricated in broader communities of practice that define the limits of acceptable action through rules,
divisions of labor, and shared, though often contentiously defined, motives and objectives. Further, the tools that individuals use in these organizational contexts are interconnected and co-evolve in fascinating ways. In this sense, complex multitool interaction should be considered _ecologically_, wherein the tool assemblage is greater than the sum of its individual parts and the tool ecology is the mediator of activity—not the individual tools themselves. Spinuzzi argues that compound mediations, or the rich tool assemblages that collectively mediate user experiences at the macroscopic layer, can be considered “genre ecologies” or “the genres of artifacts [that] collectively mediate the workers’ activities, and in so doing, become interconnected with each other in meditational relationships” (Spinuzzi, 2003, p. 101).

Spinuzzi’s genre ecology framework is not without precedent in Writing Studies. In “Inertextuality in Tax Accounting: Generic, Referential, and Functional,” Devitt (1991) argued that “genre sets” coordinate the activity of individuals in particular contexts. Working from Miller’s notion that genres are forms of social action that provide stabilized responses to recurrent situations, Devitt argues that disciplinarily developed, stabilized, and regulated texts interweave to direct the action of discourse communities in powerful, yet circumscribed, ways. Orlikowski and Yates’s (1994) article “Genre Repertoire: Structuring the Communicative Practices in Organizations” treads similar territory, arguing that while textual genres determine sequence in organizational communication, they also overlap. By tracing the sequences and overlaps of genres, the authors argue that “[T]o understand a community’s communicative practices, we must examine the set of genres that are routinely enacted by members of the community” (Orlikowski & Yates, 1994, p. 542). Bazerman’s “Systems of Genre and the Enactment of Social Intentions” is another iteration of the genre sets or genre repertoire framework. In this book chapter, Bazerman critiques speech act theory and provides an account of genre systems, or interrelated genres that interact with each other in specific settings. Only a limited range of genres may appropriately follow upon one another in particular settings, because the success conditions of the actions of each require various states of affairs to exist. (Bazerman, 1994, pp. 97–98)

Where Spinuzzi’s account of genre ecologies and the works by Devitt, Orlikowski and Yates, and Bazerman differ is his explicit connection between collections of rhetorical genres and the role of mediating tools in _digital activity systems_. The genre ecologies that mediate the collective activity of an entire digital ecosystem like Question.cd mirror Spinuzzi’s genre ecologies in that they comprise visible, interlocking tools that direct user action through compound mediation (Figure 1); however, this article argues that largely invisible rhetorical genres function at macroscopic scope, making such genre ecologies possible in the first place. Like mesoscopic tools, these mediators are also the
product of cultural–historical development in communities; however, instead of assists users in carrying out conscious, goal-directed actions, these tools embody social motives and collective desires, defining the bounds of agency community-wide while allowing Question.cd to exist at all. Going forward, the next section will explore two such macroscopic mediating technologies used to coordinate collective activity in the Question.cd activity system: the Gazelle CMS and the Oscelot bittorrent tracker. By analyzing their cultural–historical genesis as well as their structural transformation, this section argues that code-based digital rhetorical genres at the macroscopic layer of scale embody the collective social motives of the Question.cd participatory archive while also structuring the agenic capacities of other tools at meso- and micro-levels of practice.

Figure 1. Genre coordination across meso and macro layers of scope in the Question.cd activity system.
The Gazelle CMS: Coordinating MesoscopicEcologies

Cultural–Historical Background

CMS are computer programs that allow individuals to publish, edit, and modify their web content without direct .html modification. While web 1.0 sites contained static pages that required knowledge of .html and .css to change content and appearance, web 2.0 technologies such as the web CMS allow lay users to manipulate their content through a central WYSIWYG editor before publishing directly to the web. In this sense, the web CMS allows site authors to produce content, deliver content, and retrieve content without specialized coding literacies. Most CMS-powered websites utilize .php (PHP: Hypertext Processor), a scripting language designed to take in user inputs such as mouse clicks and output database driven .html documents. The use of .php is what makes CMS websites dynamic as the server-side scripting language receives signals to modify, retrieve, and display web content that is modified without direct manipulation of the .html code. PHP is ubiquitous; in fact, the world’s largest social network, Facebook, uses .php to serve its 1 billion plus users over 570 billion page views per month.

The most famous web-CMS currently in use is the WordPress platform. Initially released in 2003, WordPress has been downloaded and installed over 65 million times and is used as the CMS of choice in 22% of all new websites registered with ICANN (Usage of Content Management Systems for Websites, 2013). What makes WordPress especially unique is its clean user interface and relatively straightforward customizability. Plugins, widgets, and themes allow site administrators to expand site capacities, transforming this once humble blogging tool into a CMS capable of running enterprise-level websites. Other web-CMSs like Drupal and Joomla! include similar functionalities to WordPress but are designed to provide network capacities for enterprise applications. Academics are even putting web-CMSs to use with greater frequency. Created by the Public Knowledge Project, Open Journal Systems (OJS) is a well-respected, multilanguage CMS used to digitize the academic journal submission, review, and publication process. At present, OJS is used as the CMS of choice by at least 11,500 academic journals worldwide.

While piratical platforms like Napster and KaZaa offered massive media archives, they lacked functionalities that encouraged community. The first CMSs developed for digital pirates amended this problem, providing users with rudimentary means of connecting with one another. TBsource, an early all-in-one CMS-tracker platform included social components such as user profiles, public forums, and private messaging systems. While these site functionalities relatively basic today, in 2003, they were new to file sharing communities and tapped the power of emerging digital social networks. The social functionalities of TBsource allowed file sharers to connect in new ways, drawing together small user bases around niche media such as anime, eLearning materials, or Mac applications. With the demise of larger peer-to-peer networks and the integration of social
media technologies in private bittorrent communities, user bases exploded. In fact, before being taken down by European law enforcement organizations, the iconic Oink.me bittorrent community grew from 1,000 users to 100,000 users in just over a year. With the increase in users came an increase on server-side resources, resulting in a lack of performance from the site. After 2 years and nearly 150,000 registrations, Question.cd’s own site began to falter, returning 404 errors and crashing on a weekly basis. As a result, a team of community members at Question.cd decided to work together on a new CMS: Gazelle.

It Is Quick: Gazelle as Macroscopic Rhetorical Genre

Like any rhetorical genre, the Gazelle CMS undergoes transformations as social motives shift and contradictions in site-wide activity occur. In its initial development phase, Gazelle addressed contradictions created by the technical incapacities of the original TBsource CMS code. When the Question.cd user base ballooned after the inclusion of social profiles, forums, and private messaging, the original TBsource could not handle the new server loads and data transfers required to keep the site running. As a result, a group of site members worked feverishly over the course of 8 months to write a new CMS from scratch. The contradictions in macroscopic activity were huge, resulting in a newly emergent macroscopic rhetorical genre: Gazelle. In the sections that follow, I will briefly describe what social motives gave rise to Gazelle code changes over the course of its development and analyze its substantive, formal, and medial features. I will also illustrate Gazelle’s role as macroscopic rhetorical genre that mediates mesoscopic activity through genre ecology coordination.

Thankfully, the development process of Gazelle is preserved in extensive changelogs on the Project Gazelle website. By analyzing these changelogs, we get a peek into the incorporation of social motives and user desires in technological genre functioning at a macroscopic scale. To explore the substantive needs met by this digital rhetorical genre, I will investigate the development of artist permissions, subscription features, and the collector addition. While each of these tools are used by site members at the mesoscopic level of user-directed action, Gazelle coordinates all of these features in a broader genre ecology that emphasizes collective site members’ desire to expand the archive, discover new music, and acquire digital media.

One of the first tools developed for the Gazelle CMS was the “Artist Permissions” function. As the Question.cd community user base grew, many artists realized that exposure in such an environment might increase their visibility, and hence, their viability as creatives. Not long after the launch of Question.cd, well-known artists like Trent Reznor began releasing their tracks to the Question.cd community, encouraging users to share and reuse their creations to make new compositions. Little known artists followed his lead, betting on the importance of exposure and the power of long-tail guerrilla marketing to increase their viability. Because many
unknown artists were already uploading their content to the Question.cd archive, developers of the Gazelle CMS incorporated “Artists Permissions” allowing original contributors to claim a special user class that rewarded them with decreased ratio limits and requirements while also pushing their contributions to the “Vanity House” section of the community landing page. By encouraging artist uploading of their own content, the Gazelle CMS fulfilled the broader community desire to reward content producers with higher network visibility.

The “Artists Permissions” designation exploited the social nature of the site, highlighting the works of particular artists whose personal contributions to the community increased their visibility. Yet, because of a lack of RSS integration or other notifications, site users were still unable to keep up with archival supplementation of specific artists and genres beyond manually searching them out upon every visit to the site. As a remedy, Gazelle developers introduced various subscription services to the Question.cd frontend. The subscription tool is embedded at various locations on the site, including specific artist pages, genre pages, and collages. Much akin to the subscription tools available in most conventional internet forums, the subscription tool allowed users to passively keep track of archival additions to their areas of interest. By utilizing a RSS module tethered to specific user accounts, the subscription tool provides users the ability to passively catch up with the artists, genres, and collages that most interest them. By writing subscriptions into Gazelle, developers further met the substantive needs of the CMS as macroscopic rhetorical genre, increasing ease of archival navigation and new music discovery.

One of the last new tools designed for the Gazelle CMS is the “Collector” addition (Figure 2). Not incorporated until release candidate two, the collector tool allows users to automatically sift and download media based on technical specifications. To use the “Collector” addition users, first specify preferences for

![Figure 2. The collector tool.](image)
bit rates and media formats in their profile settings. After initial setup, users can visit an artist page and automatically download all torrents that meet their technical requirements. Because the Question.cd community comprises many audiophiles who prefer lossless quality audio rips, the collector addition fulfills the substantive communal desire to curate personal media collections with content of precise quality and format.

While I have only explored three mesoscopic tools integrated into the Gazelle CMS, there are many, many more. What is important to note here is not necessarily the individual tools themselves as they function at the mesoscopic layer of action. Rather, macroscopically, Gazelle coordinates tool-use across the entire Question.cd activity system. By incorporating sharing technologies into its design, the developers of Gazelle made conscious choices about how to best facilitate the collective motives of the entire site, namely, the desire to share information, discover new music, and connect to fellow niche music aficionados. In this way, the CMS operates substantively as a mediating technology at the macroscopic level of scale, harnessing aggregated social data to promote community inclusion, artist discovery, and archival expansion. By meeting these social motives with novel navigation designs, interfaces and information architectures, the Gazelle coding team architected contextually bound, communally specific technologies that shaped user experience in profound ways, facilitating user engagement through threaded forum conversations, feedback mechanisms, and collaborations among users to create digital media collages.

The technological affordances of the Gazelle CMS are remarkable and hinge on their interfacial pliability. Taking direction from long-established CMSs like WordPress and Drupal, Gazelle allows users to redesign the pages to better meet their information sifting habits. Customizable pages on the Gazelle CMS include artist pages, forums, requests, colleges, and formal documentation, among others. The interfacial design affordances of Gazelle allow users to manipulate information architectures, customizing them for individual navigational purposes. Each of these pliable spaces is marked with a black arrow in Figure 3. This malleability is an essential feature of digital rhetorical genres at the macroscopic level of scale because it allows for specific user customization that directs mesoscopic action while still facilitating the collective action of the entire system. As what Brooke calls an “ever-elastic middle,” the Gazelle CMS interface renders static outputs but exists, at the macro level, as a process-structuring agent in the Question.cd ecosystem (Brooke, 2000, p. 25).

Oscelot: Mediating Connections

Cultural–Historical Background

Gazelle’s coordination of multiple mesoscopic genres is relatively visible inasmuch as users directly manipulate the interface, forging personalized means of
Lewis
Figure 3. Continued.
navigation and discovery. As a digital rhetorical genre, Gazelle facilitates the broader social motive of the site: participatory archival extension and curation, while also coordinating multiple mediating technologies that encourage community formation and division of labor. Yet, for all its novelty as a macroscopic genre that organizes and commands multiple interlocking genre ecologies, Gazelle is incapable of actually connecting distributed users in the act of sharing. To achieve this collective social motive, all bittorrent-based piracy communities require a tracker. Trackers are lightweight pieces of software that facilitate sharing among bittorrent users, functioning as the only peer-to-peer point of connection in bittorrent ecosystems. As such, the tracker—even more than the CMS—moves piracy communities away from niche affinity groups toward file-sharing publics. Yet, private piracy communities like Question.cd are not the only organizations that use trackers to facilitate file sharing and distribution. Linux, Wikipedia, Project Gutenberg, and the contents of Archive.org can also be distributed via bittorrent as the distributed nature of the protocol reduces resource load, allowing for massive distributions with relatively small technological demands.

As a digital rhetorical genre functioning at the macroscopic level of scope, Oscelot was developed because of destabilizations in the Question.cd activity system. After the demise of the legendary Oink’s Pink Palace bittorrent community in October 2007, new bittorrent communities sprung up to take its place. In fact, in the first 2 weeks after the Oink.cd shutdown, at least six private, invite-only bittorrent sites launched, providing safe haven to pirates who once considered Oink’s Pink Palace home. Utilizing the same bittorrent tracker software as Oink’s, Question.cd was one of those communities; however, Question’s popularity would rise exponentially in a relatively small amount of time, creating problems for both the hardware and software that facilitated connection among site members.

After its initial launch, Question.cd purposefully kept its user base small to ensure that pages loaded quickly, and people found the content they were looking for in a timely manner. But as the site population grew, so did the load on the tracker responsible for coordinating connections among community members. As of 2008, the tracker’s technological capacities were nearing their threshold. Only a year after its launch, the Question.cd tracker software, called XBTT, was responsible for coordinating over 5 million daily connections at a rate of 3,500 per sec. This meant that when an individual user’s bittorrent client software called out for connection to other sharers, the tracker had only 80 ms to search through its database of over 900,000 torrents and 5,000,000 peers to compute a response and send it back to the user. XBTT could handle this kind of load; however, as the community grew, it was obvious that a new tracker solution was necessary to ensure the stability of the Question.cd piractical activity system.

The Question.cd administrators anticipated the technological incapacity of XBTT and began to explore solutions soon after the site launch in 2007. First, a
developer called Yubaba tried to rewrite the tracker in .php instead of C++. The resulting tracker, Lioness, was very efficient but could not scale to the user base site administrators anticipated in the coming years. In early 2009, another developer Kamaji coded a preliminary version of Oscelot in C++; unfortunately, upon further testing, Kamaji’s tracker software also failed to support the load required to connect the hundreds of thousands of users of Question.cd. By early 2010, the XBTT tracker simply could not support the size of the community and Question.cd began to time out. 404 s, tracker errors, and delayed tracker announcements became so common that users began to speculate about the scalability of tracker software and the technological limits of bittorrent itself. As a remedy, Question.cd administrators began doubling and tripling instances of XBTT. By multiplying the number of XBTT trackers running at the same time, site developers bought another few months; however, this solution was risky as the stability of archival connections began to waver. Site administrators risked crashing the entire tracker by continuing to stack XBTT on the server hardware.

**Destabilization, Contradiction and Restabilization: Rise of Oscelot**

The Question.cd administrators faced what Spinuzzi calls “destabilizations” in their macroscopic activity system. Until the beginning of 2010, the XBTT tracker functioned as a macroscopic rhetorical genre, facilitating the collective motive of file sharing by providing the structural means of connection and coordination. The destabilization caused by expanding user bases and archives created a central contradiction and, as a result, the annihilation of the entire community and archive became a distinct possibility. Because varying levels of scope are coconstitutive, the destabilizations of XBTT as a macroscopic mediating technology bore breakdowns at the mesoscopic and microscopic levels as well: because the tracker was incapable bearing the required peer connection load, users lost interest in continually curating and extending the archive through tools of navigation and discovery; further, because of their disengagement at the level of user-directed action, community members also neglected microscopic operations like logging in and browsing the archive. Due to inactivity at the microscopic and mesoscopic level, many users lost their Question.cd community membership and were banned for not being good citizens. The archive suffered as well as new uploads decreased as tracker errors increased. In total, the destabilizations created by the technological inadequacy of the XBTT tracker code resulted in systemic contradictions at all three levels of scale, challenging the viability of the entire Question.cd ecosystem.

In August of 2010, Question.cd community member Yubaba returned to the Oscelot project and recoded the earlier work of various contributors, creating the first complete build of the new tracker software. Using an agile software development philosophy, Yubaba integrated many of Kamaji’s key design choices,
reworking the original C++ Ocelot to reduce code bloat and integrate various bittorrent client-end features. Site administrator “Kaonashi” recounts the initial testing of Yubaba’s redesign of Ocelot thusly:

On September 1st, ocelot was ready for performance testing. We replaced one xbtt instance with it, and it scaled. So we replaced two, and it scaled. We tweaked it a bit, and then replaced the third and fourth instances, tweaked it a bit more, and replaced the load balancer. What four XBTT instances and a load balancer were failing to handle before, was now being handled by one, singlethreaded instance of the latest ocelot. Then we pushed it harder - we lowered the announce interval to 40 minutes, and then to 30, and it scaled. Then we lowered it to 20 minutes, and linux broke before ocelot did. It was beautiful. The dev team rejoiced, and banded together to add the reaming features and fix the remaining bugs. By September 3rd, ocelot was considered feature complete, and we let it run the entire swarm - one tracker for five million peers, at a 30 minute announce interval.

Yubaba’s rewritten Ocelot confronted the technological incapacity of XBTT, providing an alternative response to mediate the macroscopic interconnections of the entire Question.cd community. It answered the contradiction of the activity system with a new, improved mediating technology that restabilized Question.cd and facilitated more expansive forms of connection among individual users. After developing, testing, and deploying Ocelot in late 2010, the entire Question.cd development team decided to open-source license the software, allowing other bittorrent communities to integrate Ocelot into their own activity systems. In this way, Ocelot not only mediates peer connections and facilitates peer-to-peer activity on Question.cd but also on nearly all private bittorrent communities on the net.

**Summing Up: Macroscopic Digital Rhetorical Genres**

Taken together, Ocelot and Gazelle function as digital objects that mediate the collective activity of the entire Question.cd community. Gazelle provides the architecture and functionality necessary to sustain collective activity in distributed systems. By introducing social technologies to participatory archival creation and curation, Gazelle operates as a mediating technology through which users interface, navigate, and explore the Question.cd archive. As a rhetorical genre operating at the macroscopic level of activity, Gazelle embodies the substantive motives of the site, architecting the activity of site members by directing their individual motives toward the sociocultural motives of archival expansion and community building. By coordinating multiple genre ecologies whose individual tools include creation and customization of navigational structures, interfaces, and information architectures, Gazelle shapes possibility and circumscribes agency in the Question.cd ecosystem. In so doing, Gazelle
constructs a unique user experience that is contextually bound, structurally specific, dynamic, and dependent on various technological affordances. The community members who first designed Gazelle confronted numerous systemic contradictions, reworking the vision of peer-to-peer file sharing by incorporating the social. Eliminating the destabilizations that plagued previous file sharing technologies, Gazelle provides a stabilized-for-now response to a recurrent rhetorical situation: how to encourage, recognize, and reward creation and curation of materials outside the bounds of capital and work. As a digital rhetorical genre functioning at the macro level of scope, Gazelle circumscribes user agency through its coordination of mesoscopic tool ecologies, mediating individual user action while also shaping user identity.

While Gazelle structures social navigation and discovery by coordinating mediating technologies on the frontend of the Question.cd activity system, Ocelot coordinates the backend networks that make content sharing possible. By condensing the C++ code and reworking the technological capacities of bittorrent software, Ocelot carries out the communal desire to acquire and share digitized media. As the backbone of the Question.cd network, Ocelot stabilizes macroscopic activity by addressing destabilizations and contradictions that arise as participatory archives scale. Further, because Ocelot was open-source licensed, it is now coordinating the technical demands of bittorrent communities across the web, allowing users in hundreds of niche participatory archive communities to carry out activity at the macroscopic level of scale.

A Prolegomenon to Collaboration Between Rhetorical Genre Theorists and User Experience Designers

The developers of Gazelle and Ocelot created software at macroscopic scales to transform interfaces, customizing navigation, and information architecture to serve the substantive needs and desires of the broader Question.cd community. In contemporary software engineering parlance, these developers were responsible for transforming user experience. UX designers are responsible for architecting the experiences that software creates for people who use them in the real world. As UX designers, community members who contributed to the creation and modification of Gazelle and Ocelot looked past the strictly functional capacities of the CMS and tracker, instead focusing on how the aesthetics and functions of the new software met the psychological and behavioral needs of individual users and the community and culture of Question.cd. In other words, by making use of particular technological affordances, the substantive desires of Question.cd community were reified in the design of the software, creating contextual, rhetorically effective macroscopic technologies that structured user experiences on the site. It is unsurprising that community members working as software engineers were able to design successful technologies for use on Question.cd. Working in what Kennedy calls the “wilds” of technical
communication, these users relied on their communal membership to design effective navigation tools, interfaces, and information architectures that utilized local knowledge and user-generated metadata (Kennedy, Forthcoming). In so doing, the multiple dev teams on Question.cd tailored macroscopic and mesoscopic technologies to the prevailing attitudes of individual site users and the *ethos* of the community.

In workplace contexts, UX designers employ a range of more formal approaches to software development. These include Waterfall and Spiral development models, Mayhew’s usability engineering lifecycle and the LUCID framework for interaction design. Earlier models using the Waterfall and Spiral schema and later models like the Capability Maturity Model or Rational Unified Process model are often characterized as too “techno-centric”; in other words, they are overly concerned with the functional aspects of the technology and fail to address how the software is used in situ. To address this techno-centricity, or what one might call the *rhetorical* nature of software development lifecycles, UX designers have recently turned to iterative universal abstract activity cycles (Hartson & Pyla, 2012, pp. 53–56) to better assess and address the human-centered contexts where software operate. These software development cycles are iterative and consist of (a) an analysis phase; (b) a design phase; (c) a prototype phase; and (d) an evaluation phase. Embedded in each of these phases are the various design fields that technical communicators and professional writers are increasingly confronting in their pedagogy, research, and practice, including information architecture, information design, navigation design, and interaction or interface design.

Although not in a workplace, the software engineers of Gazelle and Ocelot employed an analysis–design–prototype–evaluate (ADPE) process for architecting user experiences at macroscopic scale on Question.cd. Although the iterative design, prototype, and evaluate phases mirrored what one might observe in a workplace, the analysis phase was particularly successful because of the epistemic privilege exercised by coders-as-community members in the Question.cd ecosystem. To say it another way, when the CMS and tracker are conceptualized as rhetorical genres, composed of substance, structure, and technological affordances, the development teams for both pieces of software were able to map the social motives and communal desires of the community. Next, they created a CMS and tracker to coordinate the host of mesoscopic tools functioning ecologically to meet the substantive demands of site users. These included tools that facilitated novel forms of navigation, information architecture, and interface design. Although the software developers in this example likely would not characterize their role as UX designers, they did design a host of technologies that harnessed the social–communal contours of technology use in context, thereby architecting more successful user experiences for the entire Question.cd community.

By turning to Rhetorical Genre Studies as a heuristic organizing the analysis phase of universal activity cycle software development, future work in UX
design might be able to better understand how software design is bound to contexts-of-use. Mapping the interplay of structure and substance in tools as they already exist, UX designers can then deploy a host of rhetorically appropriate technological affordances to better meet the social motives and communal desires of digital communities. As a radically user and community-centered method for understanding the ways that texts and technologies shape our individual and collective experience, RGS provides a rhetorically-grounded methodology for mapping the broader activity systems here mediating technologies like software operate. With any luck, RGS can render technologies more human inasmuch as it acknowledges how cultural–historical activity is inscribed in the design of technologies we use to mediate our collective existence; further, after the work of critique, an RGS approach to UX could be used to inform future technology creation throughout the ADPE software development lifecycle.

By proposing an intersection of Rhetorical Genre Studies and UX, I do not presuppose that rhetoricians cannot learn something from this productive exchange; rather, I am also interested in the ways that UX can inform rhetorical theory. Of particular use to RGS is UX Design’s articulation of software as composed of (a) functionality and (b) information. These two areas deeply inform interface design, navigation design, information architecture, and interaction design (Garrett, 2010); further, the resonances between this bifurcation and Rhetorical Genre Theory’s theorization of structure and substance warrant further investigation.

According to UX experts, the functional aspects of software are linked to the structural elements that create particular interactions via interface design. For example, Amazon has established what we might call the functional archetype of most e-commerce websites. Conventions integrated at Amazon and common across the structure of other digital storefronts include a centrally located search bar, hierarchical product organization, product reviews, and virtual shopping carts. These structural conventions undergo incremental shifts over time to better meet the generic expectations users have of particular technologies and facilitate particular kinds of interactions that are familiar to users. At Amazon.com, this includes product sifting, reviewing, and purchasing.

From a UX design perspective, the informational aspects of software serve particular social and individual motives and are intimately wrapped up in information architecture and navigation design. In this sense, information is contextual and analogous to Rhetorical Genre Theory’s notion of “substance.” Through various forms of contextual inquiry, ethnography and autoethnography, UX designers gather data to make choices about information architecture and navigation to better serve the needs of prospective users. These various methods for collecting contextual data in the analysis phase could be of particular use to technical communication practitioners and pedagogues charged with conducting needs assessment, task analysis, and usability testing in their workplaces and classrooms.
Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Notes

1. Russell (1997) describes operationalization as the internalization of conscious actions that result in unconscious activity. A good example of operationalization for many individuals is the act of typing. While learning to type, each keystroke requires conscious effort on the part of the individual; however, over time and through repetition, typing becomes an operation or “learned habits or responses on which a worker unconsciously draws” (Spinuzzi, 2003, p. 144).

2. For more on textual curation and the rhetorical space of the encyclopedic tradition, see Kennedy (Forthcoming).

3. Question.cd is a pseudonym.

4. In the first five years of its existence (2007–2012), Question.cd members uploaded 1,457,817 torrents at a rate of over 1,000 torrents per day. During 2011, 2012, and 2013, new article creation of English Wikipedia articles averaged 902 per day (Wikipedia, 2014).

5. IRB approval for this project was secured through Syracuse University on February 17, 2011.

6. Following Spinuzzi vis-à-vis Russell vis-à-vis Engeström, I understand human–object activity through an integrated scope approach that includes macroscopic, mesoscopic, and microscopic divisions (Engeström, 1987; Russell, 1997; Spinuzzi, 2003). Microscopic activity is unconscious human operation. Mesoscopic activity is goal-directed individual action. Macroscopic activity is unconscious cultural–historical activity that is sometimes characterized as “context” (Kuutti & Liam, 1993).

7. Following Engestrom (1992), I define contradictions as “continuous transformations [whereby] the activity system incessantly reconstructs itself” quoted in Spinuzzi (2003, p. 67). As contradictions occur, destabilizations of the entire activity system require novel workarounds of a formal or informal nature.

8. All user handles have been changed to preserve anonymity.

9. For a brief introduction to each of these different software development philosophies, see Royce (1970), Boehm (1986), Mayhew (1999), and Kreitzberg (2008).

10. See Paulk (1993) for more information.


12. Future references to iterative universal abstract activity cycles that rely on the analysis–design–prototype–evaluate process will be abbreviated as ADPE.

13. In the period following the deployment and ongoing redesign of Gazelle and Ocelot, Question.cd membership continued to rise at a rapid rate. Although speculative and correlative, one could surmise that an increase in site membership and user activity is largely related to enhanced tracker responsivity and the deployment of a host of social tools coordinated by the new CMS.
References


**Author Biography**

Justin Lewis, 
