The Political Economy of Communication
4(1), 66–84

http://www.polecom.org

Software and Artificial Scarcity in Digital Media

John L. Sullivan, Muhlenberg College, Department of Media and Communication

Keywords: software, scarcity, new media, economics, applications, APIs, political economy, media industries

Abstract

This article outlines a critical research agenda for understanding software as a key element in the development and structure of online media markets. First, I argue that media technologies are increasingly becoming software-driven and web-enabled, thereby blurring the distinctions between media hardware and software. Second, I explore some of the techniques that media companies have utilized to commodify digital media distribution by creating artificial scarcities within the software code itself (via user interfaces as well as the underlying physical and technological supports for the distribution of media on the Internet). As more media consumption occurs via software interfaces, forms of artificial scarcity such as paywalls, search engine traffic redirection, application programming interfaces (APIs), and other software tools are increasingly shaping our experiences with audiovisual media. These technological controls allow corporations to restrict access to media content, thereby enabling differential pricing models and expanded policing of digital copyright. The implications of these trends for political economic studies of media industries are explored in the conclusion.

The transition from analog to networked, digital media has greatly expanded consumers' access to the tools that create, remix, and mass distribute cultural artifacts. However, capitalist modes of production and value creation are also slowly transforming (and indeed, curtailing) the democratic potential of these new technologies. This article maps out a research agenda which links the growing importance of software in media production, distribution, and exhibition with the economics of capitalist markets. To do so, I make two interrelated arguments. First, software has become an increasingly important intermediary layer between media content and end users since it often serves as consumers' primary entry point to media content. While we still speak of distinct media industries such as film, radio, and television, it is clear that much of the content provided by these industries are increasingly being exhibited on small-screen digital devices with Internet capability. Therefore, grasping the central dynamics of 21st Century media industries requires an understanding of the functions, internal logics, and economic foundations of software. The nascent field of software

studies can be helpful in analyzing these trends because it trains scholars' attention on the shifting architectures of the interconnected digital ecosystems that structure users' interactions with digital media.

Software is not produced, distributed, and utilized in a vacuum, however. The production of software has become a major industry, growing from a value of $149 billion in 1997 to a reported $425 billion in 2012 (Shapiro, 2014). Powerful trade associations such as the Software & Information Industry Association (SIIA; http://www.siia.net/) and The Software Alliance (http://www.bsa.org) have emerged over the past two decades to join the content industries in lobbying governments for stricter copyright protections and to promote the image of the industry to investors and the public. While software studies usefully explore the cultural codes that underlie software – along with the enablements and constraints associated with specific technical ecosystems – it has thus far ignored broader issues of structural power. Core concepts from political economy are therefore necessary to address these concerns. The second part of the article, therefore, utilizes the concepts of commodification and artificial scarcity to explore some of the methods through which media corporations have attempted to forestall the "culture of abundance" that threatened to undermine the fundamentals of the capitalist marketplace in the 1990s. The primary concern voiced by a loose confederation of media corporations and artists was that the open architecture of the Internet threatened to eliminate the scarcity of access to information goods (Boyle, 2008; Mansell, 2012). Without scarcity, corporations would find it impossible to regulate the provision of information services or establish market pricing structures around this information.

Media corporations responded to this threat in a number of ways. First, they successfully lobbied Congress to pass increasingly stringent copyright laws (such as the Copyright Term Extension Act of 1998 and the Digital Millennium Copyright Act of 2000) which governed the creation and distribution of digital information services (Boyle, 2008; Lessig, 2008; Litman, 2006). Second, media corporations also turned their efforts toward the creation of both hardware and software interfaces for media content as part of their larger efforts to control the distribution and exhibition of media content to consumers. Some examples of this included the invention of portable digital media players such as the iPod and the (now discontinued) Microsoft Zune, as well as the creation of digital portals for the accumulation and distribution of media content (such as YouTube, Flickr, and others). Each of these digital media platforms introduced their own built-in features for media uploading and distribution, while disabling other types of features by design. More recently, mobile device applications (or "apps") have furthered this drive to control the provision of online media to consumers via software channels.

As this article explores, design limitations that have been coded into consumer-friendly media software applications (the "app-ification" of digital media - Zittrain, 2009 - and the adoption of corporate-friendly standards in online protocols) have introduced new opportunities for commodification of the Internet. Because they are often pre-installed in end-user software or baked into the core code of the Internet, these shifts have been largely opaque to consumers. The expertise necessary to unpack these subtle shifts in media have rested largely with software developers, hackers, standards organizations, and regulators. Nevertheless, corporate efforts to continually shift the capabilities of "cultural software" have had market consequences. Over the last 20 years, despite the rhetoric about networked digital forms of media, traditional media industries have been quietly and successfully developing tools to commodify online media distribution: by creating artificial scarcities in the provision of digital information services to consumers and by changing the

underlying computer codes and protocols that allow for the seamless integration of online information, thereby preserving and extending capitalist accumulation models.

Software as 21st century media technologies

Any analysis of media in the 21st Century must take into account the growing pre-eminence of software (Berry, 2011; Chun, 2013; Fuller, 2008; Kitchin and Dodge, 2011; Manovich, 2013). As Manovich has argued, since early 2000s, "software has replaced a diverse array of physical, mechanical, and electronic technologies used before the 21st Century to create, store, distribute, and access cultural artifacts" (2013: 2). The sub-field of "software studies," coined by Manovich (2002) in The Language of New Media, closely examines the increasingly central role of software interfaces, logic algorithms, and interactive object environments in the creation, distribution, and audience experience of culture. Software studies offers scholars some helpful tools for better understanding this new landscape of media experience for two main reasons. First, it explores specific applications and computer languages as the objects of inquiry rather than viewing software monolithically. Second, it considers the myriad ways in which software is deeply interconnected in socio-cultural networks. Berry notes that "computer code is not solely technical though, and must be understood with respect to both the 'cultures of software' that produce it, but also the cultures of consumption that surround it" (2011: 17).

This article makes two central claims, the first of which is that computer software has become an increasingly critical interface through which consumers access media content and through which media producers and practitioners manage their creative outputs. The new primacy of software in media industries becomes apparent when we notice that media technologies (e.g., the devices that are utilized to exhibit electronic media content such as television, radio, recorded music, motion pictures, and print and photographs) have begun to directly compete with their digital analogues for consumers' time and money. According to the Recording Industry Association of America (RIAA), for example, physical sales of compact discs dwindled to just over a third of the overall sales of music in 2013, whereas digital sales and streaming subscriptions for music accounted for 64 percent of the market share (Friedlander, 2014). The transition to Internet-delivered content is not nearly as advanced in the motion picture industry, but data from the Digital Entertainment Group reveals that digital music purchases in 2013 grew 47 percent from the previous year to $1.19 billion (Fritz, 2014). For Hollywood, digital-only video sales and royalties paid by online streaming services such as Netflix and Amazon Prime are becoming increasingly important sources of revenue. These industry trends reflect larger generational shifts in how consumers access media content. For instance, a recent survey by the financial firm Deloitte found that, while 88 percent of "Boomers" (viewers aged 48-66) watched movies or television shows on a TV screen, only 44 percent of "Trailing Millenials" (viewers aged 14-24) utilized a traditional TV screen to access video content, relying instead on computers, smartphones, and tablets to access this content (Chmielewski, 2014). This shift in media consumption has another consequence. Consumers' experiences with media content will be increasingly mediated via computer software, regardless of whether this software runs on a desktop computer, tablet, mobile phone, or other portable device. Software has emerged as the key digital interface since it is the last stop before mediated content reaches the end user.

On the production side, software has also changed how media producers go about creating cultural products. Before the 1990s, for example, graphic designers, filmmakers, and animators utilized very different (and incompatible) technologies. Now, thanks to widely distributed end-user software such

as Adobe Illustrator (for vector graphics), After Effects and Blender (for 3D rendering), Premiere, Final Cut Pro, and Lightworks (for non-linear video editing), media professionals have developed radically different workflows and are able to better share their labors thanks to the inter-compatibility between these systems (Manovich, 2008).

Software "consists of lines of code – instructions and algorithms that, when combined and supplied with appropriate input, produce routines and programs capable of complex digital functions" (Kitchin and Dodge, 2011: 3). While software is typically associated with rather utilitarian functions such as data processing and presentation, and word processing, consumers are increasingly interacting with software as an intermediary layer to access forms of culture. Software tools also enable users to create their own cultural forms or to remix others. Finally, the Internet provides an instantaneous means for distributing content to hundreds of millions of other people, potentially bypassing the commercial media apparatus. Software studies point scholars to the ways in which computer languages, mathematical algorithms, and graphical interfaces shape our experiences with technology. These studies understand software as "a social-material production with a profound influence on everyday life" (Kitchin and Dodge, 2011: 13). Software is conceptualized here not as a static collection of lifeless computer code, but instead as the product of complex social, organizational, and economic processes. The effects of software are also "not deterministic or universal," but are enmeshed within specific social and cultural contexts (2011: 23). Software acts upon the world, though often in unpredictable ways.

Software as media technologies

Thanks to the rise of networked computing and the convergence of digital technologies in the 1990s and beyond, media technologies are becoming increasingly transformed into software. The differences between hardware and software are blurring, such that information content (which has been traditionally defined as "soft") now appears in many instances as immutable as hardware (Chun, 2013: 6). This has several important consequences. Certainly, the shift toward software has enabled a greater expansion in the capabilities of media technologies by leveraging the dynamic nature of their own internal structures as well as their interconnections with other information services delivered via the Internet (web audio services such as Spotify and Last.FM illustrate this nicely). Perhaps more importantly for scholars, however, software interfaces also allow media corporations unprecedented control over the actual technology itself. Media software can change the very nature of the user interface on an ongoing basis and shape the types of content provided via the software to suit the financial or legal prerogatives of corporations. Yet to the end user these changes appear material and concrete because they are essentially "baked in" to the software interface. Thus, both the content and the technology which transmits content to consumers are commodified.

While digital media tools such as MP3 players, mobile phones, and tablets are forms of hardware, they differ significantly from traditional forms of media (such as AM/FM radios and televisions, for example). Digital media software (such as MP3 players, online radio streamers and podcast catchers) operates by mimicking the functions of traditional media hardware such as CD and DVD players, radios, and televisions. There are of course many similarities between audio software like iTunes and a CD player (they both play music, for instance), but software-driven media technologies differ in several important ways from hardware. First, while media hardware is designed to perform a specific set of defined tasks (such as playing music or accessing specific channels of audiovisual materials), media software features a much broader range of functionality, thereby adding new capabilities and

features which can shape the end-user experience in numerous ways. YouTube, for example, allows users to view audiovisual content, but it goes further than that by allowing viewers to log in and post comments or "likes" for the current video, while also presenting the viewer with a sidebar containing numerous "related videos" to watch.

Second, unlike their offline hardware progenitors, software is much more malleable because it is not defined by an underlying physical infrastructure which must be re-designed and re-manufactured to accommodate changes and updates. Instead, software can be rewritten and distributed almost instantaneously via computer networks to deal with problems (software bugs), to add new capabilities to the software, or to disable or remove functionality. Web-based software production is increasingly centered around APIs, or Application Programming Interfaces. According to the Software Engineering Institute, an API is:

> an older technology that facilitates exchanging messages or data between two or more different software applications. API is the virtual interface between two interworking software functions, such as a word processor and a spreadsheet. An API is the software that is used to support system-level integration of multiple commercial-off-the-shelf (COTS) software existing or new applications (De Souza et al., 2004: 2).

Stated more simply, APIs are "a set of rules by which one software program can communicate with another software program" (Aitamurto and Lewis, 2013: 316). In essence, the APIs enable data exchange when one computer "'calls' another—which often includes some sort of authentication to make sure that the two should talk to each other—and then data is exchanged" (Rooney, 2012). Thanks to APIs, core software applications and services become more easily integrated with other software packages, thereby providing additional functionality to those packages. Additionally, programmers are incentivized to develop new applications for specific software platforms created by companies such as Apple (iOS) and Google (Android) because these companies provide tools to software developers (open APIs). Popular social networking sites such as Facebook, Twitter, and Instagram, for example, release open APIs which will allow software developers to more easily create new software tools that can also benefit from access to the data stream from these services. Open APIs highlight the inherently *social* nature of Web 2.0 cultural production due to the technological compatibilities built into web standards. While APIs can streamline the process of innovation and collaboration, they can also effectively constrain the types of software that developers can create by placing limits on specific features of the API or by revoking or changing the access terms at any time. These limits introduce some "hard edges" to the development potential of these web services, something that will be discussed later in this article.

Third, software is not simply a series of coded instructions for computers to perform. Instead, it is a constellation of processes enmeshed in complex socio-technical systems that shape its purposes, uses, and effects. Here we can employ concepts from Latour's actor-network theory (ANT) to better understand the role of software as a media technology. Latour's work sought to better understand how scientific theories became widely accepted within scientific communities by employing a social constructivist approach (Latour, 2005). Actor-network theory (ANT) has also been used quite extensively in technology studies to explore the integral relationships between technical and social systems (Bijker et al., 1987). In ANT, technical artifacts are seen as not as exogenous "black boxes", but instead as part of a "network of heterogeneous materials" which includes agents, social institutions, machines, and organizations (Law, 1992: 381). Latour argues that technology is not diffused in society, but is instead "translated" into myriad forms in different actor networks (Latour,

1984). Software too can be usefully understood as part of a network of social relations and as being mutually constituted by those networks of actors. Far from being a static recipe of machine code that instructs computers to perform specific functions, software is created, distributed, and expanded upon within large networks of developers, organizations, corporations, and end users.

One key aspect of software is that developers can effectively obscure its architecture by preventing outside users from seeing the underlying computer language, or source code (Gay and Buckman, 2002). The desire of computer hackers at MIT in the 1960s and 1970s to alter and improve existing software developed into a broader free software movement in the 1980s (and later, the open source movement in the 1990s) (see Bollier, 2008; Levy, 1984). Reacting to the extension of copyright protection to cover computer software in 1980, free software activists developed competing license regimes such as Berkeley Software Distribution (BSD) and the General Public License (GPL) to ensure that software developed and released using these licenses would remain in the public domain (Kelty, 2008; Weber, 2004, 2005). Beginning in 1996, a group of politically progressive computer enthusiasts called "circuit riders" began to introduce nonprofit and grassroots organizations to open source software, though they were more effective in creating a political consciousness among free software advocates than extending the use of free software tools in the nonprofit sector (McInerney, 2009). More recently, software companies that profit mainly from closed source, proprietary software (such as Sun Microsystems) have released open source versions of their products and welcomed contributions from free software hackers. However, Birkinbine (2015) has noted that these contributions are often incorporated back into the copyrighted, proprietary product, resulting in the capitalist appropriation of free labor. As these examples illustrate, software has become not just an important form of media, but one that is central to political, economic, and social conflicts around technology.

Commodification and artificial scarcity in online media

As outlined above, the focus of software studies on specific technical environments (including user interfaces) and the social and cultural contexts lurking within the computer code is helpful for scholars. Software studies unfortunately leave some important gaps when it comes to the transformations that are being wrought by capital in computer-based media. For example, issues of ownership, structural constraints, power, and end user privacy are noticeably absent. Here we must turn to Marxist concepts of commodification and the notion of artificial scarcity in order to address the pressing issues facing software and online media. Commodification, or the process of transforming use values into exchange values, lies at the core of Marx's analysis in *Capital*. Commodification operates in at least two distinct forms via software. First, software is a type of communication technology that "contribute(s) to the general process of commodification in the economy" (Mosco, 1996: 130) by supporting the process of transforming information into a discrete product that is sold and distributed. Utilizing television as an example, Mosco argues that "each step toward the digitalization of television has refined the commodification of content, allowing for the flow to be captured or, more precisely, for the commodity to be measured, monitored, and packaged in ever more specific or customized ways" (1996: 136). Second, software allows for minute control over the provision of information resources to the public, creating exchange value for those resources. One important method through which this is accomplished – the creation of artificial scarcity – is explored below. Lastly, because the operations of software are buried invisibly within the code, they

appear to the end user as natural, thereby obscuring the capitalist systems of production. This, too, is part of the broader commodification process.

The low transaction costs for creating and circulating cultural materials on the Internet has created a new forum for citizen-to-citizen discourse in competition with the traditional mass media; it seems that "anyone can be a publisher" (Bollier, 2008; Shirky, 2008). The potential for so much information bounty should logically hinder the development of markets within the online arena. As Adam Smith noted in The Wealth of Nations, "A produce of which the value is principally derived from its scarcity, is necessarily degraded by its abundance." (Smith, 1910: 158). How is value created from software, a digital information good that is easily copied and transferable from one user to the next? Software, along with other information products, is a public good. Public goods, according to Samuelson's (1954) definition, are goods that are non-rivalrous (use by one individual does not reduce the utility of that good by someone else) and non-excludable (it is difficult to prevent others from consuming the good). Samuelson suggested that public goods would often be found in public sector production due to their non-excludable nature and that the private sector would not be efficient at producing these goods. However, there are plenty of private companies that have developed sophisticated market models around the manufacture and distribution of public goods like software (Holcombe, 1997). Indeed, waves of high profile mergers and acquisitions along with steady profits for high tech firms suggest that markets for software are robust. Typically, private software producers rely upon exogenous means of creating excludability via some form of government intervention (such as copyright) in order to create the scarcity that allows markets to exist.

Private software producers can also rely on endogenous means for introducing excludability, however, and this can be understood as a form of artificial scarcity. By artificial scarcity I am referring to technical restrictions purposefully designed into software structures that hinder or otherwise degrade the core functionality of the software, including its ability to interoperate with other systems. During the industrial era, artificial scarcity was typically achieved by manufacturers purposefully slowing production (a type of managed inefficiency) in order to decrease the supply of goods in the market in an effort to inflate the exchange value. In the digital era, artificial scarcities can be built into the actual design of the good itself. As noted above, the most common method of achieving excludability in software is to withhold the source code such that users cannot alter or otherwise evaluate the constituent parts of the software. Increasingly, however, the value of software is heavily dependent upon its interoperability with other systems, thereby expanding the ability of software producers to introduce excludability for their product. Dyer-Witheford (2014) utilizes the concept of "value chains" to describe how capitalist enterprises organize subordinate systems to link them in a chain of value creation. Daubs and Manzerolle (2016) suggest further that the value network system in today's software and app production is underpinned by a preindustrial "putting out" process whereby merchants provide the raw materials to artisans in exchange for a small wage upon completion of the final product. This results in labor exploitation. As will be discussed later, features of modern web-enabled software such as web traffic redirects, standards, APIs, paywalls, and encryption (digital rights management) leverage the power of artificial scarcity to extract value from the software product.

The need to introduce artificial scarcity into media and information markets was noted by Nicholas Garnham, who reasoned that the high costs of initial production for cultural materials, coupled with the quite marginal costs of reproduction, resulted in "a contradiction at the heart of the cultural commodity" (1990: 160). In order to achieve audience maximization without becoming

victim to widespread piracy, the goal of commercialized media production was to develop strategies for "artificially limiting access in order to create scarcity" (160). Some of these strategies included:

i. Establishing monopoly or oligopolistic controls over distribution channels.

ii. Subsidizing the provision of hardware (such as radios or television receivers) as temporary loss leaders in order to encourage adoption and increase access to media (or software).

iii. Creating audiences for sale to advertisers as commodities.

iv. Creating media commodities which require constant reconsumption.

This analysis left Garnham with one central conclusion: "It is cultural distribution, not cultural production, that is the key locus of power and profit" (1990: 161–162; emphasis in original). Even though Garnham was observing a pre-Internet media era, his observations about the necessity of scarcity are highly relevant to today's online digital landscape. The key to the transition between 20th Century and 21st Century media markets can be broadly understood as a shift in content distribution, which has had recursive effects on media production practices as well. The goal of these shifts, as Garnham predicted, has been to impose scarcity on a media resource that in the 1990s seemed potentially boundless. Mansell (1999) identified several types of potential artificial scarcity in the Internet era, including the development of "set-top boxes" for accessing content, stringent enforcement of intellectual property rights, concentration in Internet domain name registrations, and the restriction of web search and portal access points for consumers wanting to locate information on the web. All of these sources of artificial scarcity have since become standard features of a market-oriented Internet environment, in ways that Mansell could not have foreseen in the years prior to Web 2.0. Indeed, as Robert McChesney has observed, "at what seems every possible turn, the Internet has been commercialized, copyrighted, patented, privatized, data-inspected, and monopolized; scarcity has been created" (2013: 218).

There are numerous forms of artificial scarcity that can be observed in online media, many of which have been well-documented by scholars. These forms of resource scarcity are "artificial" in the sense that they have been introduced by corporations, governmental agencies, the courts, technical standards boards, and software developers in order to manage the provision of information and services online. These resource scarcities exist at different layers of the Internet. Zittrain (2009: 67–68) has identified a three-layered model to help understand how online media can limit the choices of users: the physical connections, protocols, and the application layers of software. Instead of the three layer system, Meinrath et al. (2010: 427) utilized the Open Systems Interconnection Model (OSI) for their analysis, which divides communication systems into seven logical layers which demonstrate the "hierarchical dependencies of networked technology" (see Figure 1).

Figure 1. The OSI Model

		Layer	Function
←UPPER LAYERS→	7	Application	Message format, Human-Machine interfaces
	6	Presentation	Coding into 1's and 0s; Encryption, compression
	5	Session	Authentication, permissions, session restoration
← TRANSPORT SERVICE→	4	Transport	End-to-end error protocol
	3	Network	Network addressing; Routing or switching
	2	Data link	Error detection, flow control on physical link
	1	Physical	Bit stream: Physical medium, binary transmission

Source: Adapted from Cisco Systems. (2013, March 27). *What is the OSI Model & the Overall Explanation of ISO 7 Layers*. Retrieved from http://www.cisco1900router.com/what-is-ios-model-the-overall-explanation-of-ios-7-layers.html

Scholars who have taken a political-economic orientation toward the Internet have focused their attention largely on the underlying physical and technological supports for the distribution and storage of media on the Internet, including the consolidation of internet service providers (ISPs), content providers, cloud storage, and access speeds to the network (Crawford, 2013; Galloway, 2001, 2004; Lessig, 2001; McChesney, 2013; Mosco, 2014; Schiller, 1999; Zittrain, 2009). This research focuses primarily on the physical and protocol layers that Zittrain outlines (which includes the data link, network, and transport layers of the OSI model). Broadband access is one area where scholars have documented some of the impacts of artificial scarcity. Large ISPs like Comcast, Charter, and Time-Warner Cable regularly ration their networks by offering consumers tiered pricing for different speeds (Meinrath et al., 2010). More recently these companies have begun throttling consumers' speeds for video providers like Netflix in order to extract extra fees from these streaming services (Brodkin, 2014). Another less publicized form of artificial scarcity occurs when these ISPs impose bandwidth usage caps on consumers instead of allowing them unfettered access to the Internet as seen in other countries (Crawford, 2013). Some ISPs utilize the Internet's internet protocol (IP) addressing infrastructure to "to disrupt communications by enabling network administrators to censor particular content, specific users, or even entire regions of the Internet" (Meinrath et al., 2010: 441). Peer-to-peer networking services such as Bit Torrent, for example, are often blocked on the assumption that consumers utilizing this service are trafficking in pirated media content.

Examining commodification practices in software design and implementation

The preceding forms of manufactured scarcity in online media affect the transport infrastructure of the Internet in order to create pricing structures and competitive advantage for some companies in the online marketplace. While battles over transport infrastructure, particularly the net neutrality debate, have been studied closely by political economy scholars, the "new frontier" of artificial scarcity is in

the code which underlies the software interfaces and applications that govern the web. The top-most layer of the OSI model – the application layer – includes the software applications, websites, as well as the underlying software languages that guide computer and mobile operating systems. Here, there is a continual tug of war between inter-operability with web standards and the corporate strategy of creating self-contained ecosystems to maximize audience share and profits. I will now outline several important aspects of artificial scarcity through software code that should feature more prominently in scholars' understanding of online media markets.

Distribution and web traffic redirects as forms of artificial scarcity

The methods through which software directs, redirects, or actively blocks web traffic is a key form of artificial scarcity that is often used to create moments of commodification. Firewalls or paywalls that block access to specific web content are perhaps the most common manifestations of this type of scarcity. Streaming video services such as Netflix and Amazon Prime utilize subscription-based paywalls, and the New York Times most famously instituted a paywall for much of its online content in 2011. The latest company reports from the Times indicate that digital subscriptions have now outpaced digital ad revenue, which suggests some success in achieving profits through this form of basic distribution scarcity (Chittum, 2013). Paywalls and subscriptions are no panacea, however, since new distribution forms can arise to challenge existing market incumbents. Nevertheless, subscription models and paywalls, particularly in the newspaper industry, have become a popular method to extend resource scarcity into cyberspace.

While paywalls are evident in their deployment, software-guided redirects of Internet traffic are often invisible to the end user, yet these forms of restriction are becoming an increasingly critical tool for the commodification of the web. This practice is most widely utilized in web-based advertising by enticing users to digital content via "click throughs". Search portals are best positioned to profit from web traffic redirects. The Internet's largest and most profitable search engine, Google, for instance, can provide enormous leverage over websites who wish to appear in the first page of search results. According to published figures in January 2014 from ComScore, a digital traffic measurement firm, Google captures over 67 percent of the "explicit core search" market (original searches as opposed to contextually driven searches), followed by Microsoft's Bing at a distant 18 percent (McGee, 2014). The dominance that Google enjoys in the search market is not necessarily a natural expression of consumer choice, however. As Matthew Hindman (2009) discovered by scrutinizing Google's balance sheets, the company pays out billions of dollars per year in order to encourage other web sites to funnel visitors to their search engine. When, in 2013, the website RapGenius.com attempted to bolster its own web search rankings by gaining Google's algorithm (by making deals with other websites to inflate the number of hyperlinks back to RapGenius), Google modified its algorithm to place RapGenius webpages much further back in the search results. Effectively, Google meted out a manual penalty for this otherwise rational business behavior (Finch, 2013). Within a week, RapGenius' web traffic plummeted from 1.5 million hits per day to under 200,000.

These examples raise pressing issues that political-economy of communication research, informed by software studies, should investigate further. How are software algorithms being specifically employed to direct traffic on the Internet and how do these re-directs affect the dynamics of power within the Internet industries? Moreover, what are the potential impacts of these forces on end users for their freedom of web movement and for the privacy of their data? Robert Gehl's (2011; 2014)

work on the "affective processing" of users' personal data by social media companies as a tool for commodification, for example, points to the utility of combining software studies with a political-economy perspective.

APIs and online software ecosystems

Software studies directs scholars' attention to the specific software ecosystems and the technical and socio-cultural contexts that have developed around those ecosystems. As mentioned in the introduction, one of the key aspects of 21st Century software is that it provides seamless interfaces with other software thanks to the compatibilities built into the underlying code. These compatibilities allow independent software developers to utilize standardized tools to more easily create new software interfaces which can leverage the benefits of existing services. Web-based software companies like Facebook, Twitter, and Google, for example, adopt open APIs in order to enable seamless digital content-sharing between content providers and third-party developers. With open APIs, companies such as Facebook and Google invite external developers to build other software services and web applications around their content − for instance, by republishing the original content in a new environment or by adding functionality to the core software service in question. Some newspapers have adopted open APIs for their online versions in order to drive traffic back to their websites. In their analysis of The New York Times, The Guardian, USA Today, and NPR, Aitamurto and Lewis (2013) found that releasing open APIs for software developers spurred more innovative research and development, which had recursive positive impacts on the newspapers by generating new means for the commercialization of their content.

The interconnectedness of Web 2.0 systems through APIs, while tremendously enabling, can also act as an artificial constraint on third party software developers by setting the technical terms of software ecosystems. One of the consequences of APIs is that software developers are not necessarily "aware of their colleagues' actions" and this effects a "reification of organizational boundaries" (De Souza et al., 2004: 7). Indeed, as Bodle (2011) found in his analysis of the Facebook API, these software ecosystem models are often utilized to ensure market dominance by one company through the selection of specific features to release through the API. This is evident in the tussle between tech giants Google and Facebook. Worried about the concentration of lucrative user data on Facebook's servers, Google partnered with Yahoo! and MySpace in 2007 to launch the Open Social API, which would allow user contact data to be shared across these services. Facebook prevented its user data from being shared with these services via its own API for strategic reasons. Instead, Facebook signed an agreement with Microsoft to share its user contact details via the Windows Live Contacts API (Farber, 2008). Bodle also noted that users' privacy was placed at risk thanks to Facebook's open API, which funneled more user data back to Facebook than was apparent when consumers utilized third party Facebook applications, particularly on mobile platforms such as iOS and Android. In response to the negative publicity about its privacy settings, Facebook altered the terms of its API in 2014, removing the ability of third-party applications to access users' friends' data without the consent of those friends (Constine, 2015).

As the number of their users has grown along with their potential value to advertisers, social media companies have begun shifting the terms of use for access to their data via software APIs. In 2012, for example, Twitter announced the release of version 1.1 of its API, which not only capped the amount of data that third-party software developers could receive from Twitter, but also made it much more difficult for competing applications to gain access to Twitter data, thereby imperiling their

existence (Warren, 2012). Thus, the interoperability of Web 2.0 software, while enabling innovation through the development of software which leverages web services like YouTube and Facebook, can also extend the market power of those services through the open API (by restricting access and determining the type of third-party software applications that interface access their data).

The extensive deployment of APIs across the software industry has also reshaped the nature of production in the capitalist system. Daubs and Manzerolle (2016: 57) have argued that online cultural production comprises a form of "cognitive capitalism" wherein the tools and raw materials of production (software code and APIs) are provided by merchant capital (such as Apple and Google) to "harness and articulate the capacities of immaterial labor(ers). The latter includes coders and app developers, but also the more general category of prosumers." While these cultural laborers are encouraged to unleash their creative energy through a system that provides them with access to tools and a sense of autonomy, in reality the software products that result "are sold back to corporate capital at significant disadvantage, both in the 30 percent 'commission' and in the cost of basic tools, infrastructure, and education" (58). For the app developers who rely on the back-end data provided via APIs to bring material value to their labor, this dynamic concentrates market power within the hands of the technology firms, neatly obscured by the prepackaged software libraries.

APIs themselves have now been rebranded by industry players as a highly valued commodity that requires strategic management. A recent article in the Harvard Business Review (Iyer and Subramaniam, 2015), for example, noted that APIs represented a key strategic asset that can drive revenues of the core business. Technology companies have begun pursuing intellectual property protections for their APIs via that court system. The most famous and wide-reaching case has been that of Oracle, owner of the Java programming language. In August 2010, Oracle sued Google for copyright and patent infringement, arguing that they had infringed on Oracle's copyright by utilizing Java APIs to write its own version of the Java language which was implemented in its mobile Android operating system. On May 31, 2012, the court issued a decision vindicating Google, arguing that APIs were more like "how to" instructions than the underlying codebase of Java (Mullin, 2012). Oracle appealed, and in May, 2014, the U.S. Court of Appeals for the Federal Circuit in Washington, D.C. reversed the lower court ruling, with Judge Kathleen O'Malley concluding that "a set of commands to instruct a computer to carry out desired operations may contain expression that is eligible for copyright protection" (Levine and Bartz, 2014). Since Google's request for a review by the Supreme Court was denied in June 2015 (Tofel, 2015), the ruling makes it more likely that other technology companies will begin to leverage their APIs to police the behaviors of other programs written with those APIs. This will centralize more control through the interconnected nature of web software. These practices of software commodification and commodification through software interfaces are worthy of further attention by scholars.

Software standards and market power

The genesis and development of technical standards in software are key to the software studies literature as well, and the strategic deployment of specific types of standards can also introduce artificial scarcities in online markets. In their seminal 2010 overview of "digital feudalism", Meinrath et. al. note that important universal software standards for the web, such as the HTTP standard, are part of the software layer that is being squeezed by proprietary standards. For example, they point to the use of the proprietary H.264 video codec, which was incorporated into video streaming software such as Flash (Adobe) and Silverlight (Microsoft). There are other examples regarding software

standardization that bear closer scrutiny by critical scholars as well. The development of the HTML5 web standard is one such issue. Bowing to pressure from the content industries to include digital rights management (DRM) software, encrypted media extension (EME) protocols were built into the video streaming codes of the new standard in 2013 (Mullin, 2012). In order to ascertain whether the end user has the permission to view copyrighted content, these protocols allow a third party application to potentially send identifying information about what is being streamed, from where it is being streamed (the IP address), as well as a specific "rights token" about the permission to stream that content. This has the potential to transform a video streaming protocol into a potential new source of commodified consumer data and to pose unforeseen risks to end user privacy.

Software standards generation and their likely impacts, therefore, are important concerns for political-economy scholars. What factors and institutions shape the development and adoption of these standards? Recent scholarship by Jonathan Sterne (2012) on the development of the MP3 audio codec as a ubiquitous online software format and its connection to standards-making bodies, such as the Moving Picture Experts Group (MPEG), demonstrates the value in considering how software is interconnected with standards bodies, market forces, and regulators.

App-ification of digital media

Access to the web has been increasingly filtered via mobile applications (often utilizing the API interfaces outlined above). This can limit speed and restrict end users' ability to access particular types of information. As more consumers adopt mobile platforms like tablets and smartphones, their experiences with the web are funneled through these apps, which are often stripped down versions of web services with a limited range of functionality. Zittrain (2009) expressed deep skepticism about the "app-ification" of the Internet, since this places real limits on the functionality of mobile devices such as the Apple's iPhone. This lockdown strategy for mobile devices is achieved via software restrictions, both at the level of the operating system (iOS) and within individual apps. These restrictions extend the hold of Apple's business model into the hardware itself, and has been effective in maximizing Apple's revenue even though Android is the more ubiquitous operating system on mobile handsets.

Along with restrictions that are part of the creation of smaller apps for mobile platforms, the real restraints emerge with app stores enabling consumers to access them. Apple's app store, for example, is the largest, with sales of $18 million per day for the top 200 apps, followed by the Google Play store with $12 million in sales per day (Jones, 2013). Given these extraordinary revenues, access to the top software app stores carries restrictions for developers. Some apps are denied access to the store, often for reasons that are opaque to the public and the app developer. For example, in 2012 Apple changed its app store policies to outlaw apps that linked to other apps, which was widely seen as an attempt to prevent software from duplicating the features of Apple's app store (Panzarino, 2012). Similar to the market for cable and satellite television channels, there is also a funneling effect for mobile software apps that are distributed via the two most popular app stores, the Apple App Store and Google Play. By restricting some apps and marketing others, these online marketplaces – wherein the apps themselves are commodified – work to create another type of online scarcity.

The structure of apps also creates a limiting effect on the end users. In order to broaden their user base as widely as possible, mobile app developers have widely adopted a "freemium" model, whereby the app is freely available for download from the app store but its functionality is limited unless the user purchases an upgrade to the software. Thus, the software that is initially distributed to end users

disables many important features until the user fully "unlocks" or enables them by paying money, thereby commodifying specific software instructions within the app. This freemium model is well established in mobile app markets. A recent survey of more than 2,000 mobile software developers found that 92 percent of their revenue came from this sort of monetization model (Koetsier, 2014). This model is being pushed to its logical extreme by Microsoft, which officially launched mobile versions of their profitable Microsoft Office suite in March of 2014. On tablets such as Apple's iPad, users can download the apps for free, but Microsoft disabled all editing functionality to the software unless users subscribed to Office 365 for a $100 annual fee (Wingfield, 2014).

This strategy of purposefully "gimping" software apps can also be utilized to enforce DRM (digital rights management) restrictions on specific types of media content. Here again, scholars can usefully turn to software studies for guidance by closely examining both the conditions of production for mobile apps as well as their use by consumers. The rising importance of software apps able to access web-related media content and services foreshadows a future in which companies increasingly control the tools to access the Internet.

Conclusion: software and web commodification

Computer software has emerged as a critical aspect of 21st Century media production, distribution, and consumption. Increasingly, consumers are accessing audiovisual media via web-enhanced software interfaces rather than stand-alone media technologies (hardware). The differences between media hardware and software are beginning to erode in the online space, however. The fungible nature of software code is hardening thanks to the complex technical ecosystems and economic markets that have been constructed around specific software applications and protocols. The emergent field of software studies can provide scholars with useful guideposts for understanding these new realities in online media. In particular, software studies focuses scholarly attention on the development and implementation of specific software systems, applications, and protocols and their impacts on cultural production and consumption. While the construction and operation of computer systems are no doubt the raison d'être of software studies, scholars working in this area do not subscribe to technological determinism. Instead, software is conceptualized as a set of practices that are deeply enmeshed in social, organizational, and economic processes.

For all that it has to offer us in understanding our current media environment, software studies exhibits some blindspots that necessitate the addition of political-economic concepts. For example, issues of power, commodification, and control have, to date, been left largely unexamined by software studies. When we look closely at the recent changes in online media, it is clear that web-based software protocols are being designed to further the logics of capitalism. This has been accomplished through the careful management of information flows on the web to create artificial scarcities which are then commodified and turned into profit. Political economists have had a long tradition of considering the methods through which manufactured scarcities have commodified the physical, transport, and network layers of the internet. As I have argued in this article, these tactics are now being increasingly applied to the application layer of the internet through the use of web traffic redirects, the strategic deployment of APIs, the inclusion of proprietary protocols within open web standards, and the monetization of discrete software functions via app-ification. Software studies can therefore provide an important empirical focus for political economy. Future work in this area will need to examine more closely the technical infrastructures built into software protocols and functions in order to look for modes of capitalist accumulation. The "value networks" being exploited here are

not only comprised of organizations, processes, and labor, but the essence of the software product itself.

The fact that software is becoming the front line in the broader commodification of the web represents both challenges and opportunities for political economists. The sheer ubiquity of software interfaces in our media environment and the proliferation of potential "billable moments" renders these practices visible but potentially difficult to capture in their complexity. The fact that many of these practices are deeply embedded within the code and protocols presents something of a challenge for scholars and the public because it will require a greater knowledge of, and facility with computer programming languages. The importance of the underlying protocols for the distribution of media on the web also points to the necessity of re-examining internet governance and the social construction of technical standards through the lens of political economy.

Author Bio

John L Sullivan is Professor of Media and Communication at Muhlenberg College in Allentown, Pennsylvania. His research explores the links between media industries and systems of social and economic power. More specifically, he focuses on the constructions of audiences within media organizations, on the implementation of US media policies and on the political economy of online cultural production. Recently, he has begun a longer-term project to study podcasting as an online cultural industry. He is the author of Media Audiences (Sage, 2012).

References

Aitamurto T and Lewis SC (2013) Open innovation in digital journalism: Examining the impact of Open APIs at four news organizations. *New Media & Society* 15(2): 314–331.

Berry DM (2011) *The philosophy of software: Code and mediation in the digital age*. Basingstoke, Hampshire; New York: Palgrave Macmillan.

Bijker WE, Hughes TP, Pinch T, et al (Eds) (1987) *The social construction of technological systems: New directions in the sociology and history of technology*. Cambridge, MA: The MIT Press.

Birkinbine BJ (2015) Conflict in the commons: Towards a political economy of corporate involvement in free and open source software. *The Political Economy of Communication* 2(2). Available from: http://www.polecom.org/index.php/polecom/article/view/35 (accessed 20 March 2016).

Bodle R (2011) Regimes of sharing. *Information, Communication & Society* 14(3): 320–337.

Bollier D (2008) *Viral spiral: How the commoners built a digital republic of their own*. New York, NY: New Press.

Boyle J (2008) *The public domain: Enclosing the commons of the mind*. New Haven and London: Yale University Press.

Brodkin J (2014) Netflix users on Verizon and AT&T get raw deal, have little reason for hope. *Ars Technica*. Available from: http://arstechnica.com/business/2014/03/netflix-users-on-verizon-and-att-get-raw-deal-have-little-reason-for-hope/ (accessed 1 April 2014).

Chittum R (2013) The NYT's paywall overtakes digital ads. *Columbia Journalism Review*. Available from: http://www.cjr.org/the_audit/the_stand-alone_new_york_times.php (accessed 1 April 2014).

Chmielewski D (2014) Millennials spend more time watching TV on other devices. *Re/code*. Available from: http://recode.net/2014/03/26/teens-twentysomethings-spend-more-time-watching-tv-on-other-screens/ (accessed 26 March 2014).

Chun WHK (2013) *Programmed visions: Software and memory*. Boston, MA: The MIT Press.

Constine J (2015) Facebook is shutting down its API for giving your friends' data to apps | TechCrunch. *Tech Crunch*. Available from: http://techcrunch.com/2015/04/28/facebook-api-shut-down/ (accessed 2 March 2016).

Crawford SP (2013) *Captive audience: The telecom industry and monopoly power in the new gilded age*. New Haven, CT: Yale University Press.

Daubs MS and Manzerolle VR (2016) App-centric mobile media and commoditization: Implications for the future of the open Web. *Mobile Media & Communication* 4(1): 52–68.

De Souza CR, Redmiles D, Cheng L-T, et al (2004) How a good software practice thwarts collaboration: The multiple roles of APIs in software development. In: *ACM SIGSOFT Software Engineering Notes*, pp. 221–230. Available from: http://dl.acm.org/citation.cfm?id=1029925 (accessed 1 April 2014).

Dyer-Witheford N (2014) App worker. In: Miller PD and Matviyenko S (Eds), *The imaginary app.* Cambridge, MA: The MIT Press.

Farber D (2008) Facebook ignores OpenSocial, embraces Windows Live Contacts API. *CNET*. Available from: http://www.cnet.com/news/facebook-ignores-opensocial-embraces-windows-live-contacts-api/ (accessed 2 March 2016).

Finch D (2013) RapGenius - What does it look like to get a Google smackdown? *Tech Rampage*. Available from: http://techrampage.com/google/rapgenius-look-like-get-google-smackdown-438 (accessed 1 April 2014).

Friedlander JP (2014) *News and notes on 2013 RIAA music industry shipment and revenue statistics*. Washington, DC: Recording Industry Association of America. Available from: http://www.riaa.com/newsitem.php?content_selector=newsandviews&news_month_filter=3&news_year_filter=2014&id=34CE7808-40B2-5E5A-A91A-7165FCEF4B54 (accessed 23 June 2014).

Fritz B (2014) Sales of digital movies surge. *Wall Street Journal*, 7 January. Available from: http://online.wsj.com/news/articles/SB10001424052702304887104579306440621142958 (accessed 26 March 2014).

Fuller M (2008) *Software studies: A lexicon*. Cambridge, MA: MIT Press.

Galloway AR (2001) Protocol, or, how control exists after decentralization. *Rethinking Marxism* 13(3–4): 81–88.

Galloway AR (2004) *Protocol: How control exists after decentralization*. Cambridge, MA: MIT Press.

Garnham N (1990) *Capitalism and communication: Global culture and the economics of information*. The Media, culture and society series, London: Sage.

Gay J and Buckman RE (2002) A note on software. In: *Free software, free society: Selected essays of Richard M. Stallman.* Boston, MA: Free Software Foundation, pp. 3–5.

Gehl RW (2011) The archive and the processor: The internal logic of Web 2.0. *New Media & Society* 13(8): 1228–1244.

Gehl RW (2014) *Reverse engineering social media: Software, culture, and political economy in new media capitalism*. Philadelphia, PA: Temple University Press.

Hindman MS (2009) *The myth of digital democracy*. Princeton: Princeton University Press.

Holcombe RG (1997) A theory of the theory of public goods. *The Review of Austrian Economics* 10(1): 1–22.

Iyer B and Subramaniam M (2015) The Strategic Value of APIs. *Harvard Business Review*. Available from: https://hbr.org/2015/01/the-strategic-value-of-apis (accessed 2 January 2016).

Jones C (2013) Google play catching up to Apple's app store. *Forbes*. Available from: http://www.forbes.com/sites/chuckjones/2013/12/19/google-play-catching-up-to-apples-app-store/ (accessed 1 April 2014).

Kelty CM (2008) *Two bits: The cultural significance of free software*. Durham: Duke University Press.

Kitchin R and Dodge M (2011) *Code/space software and everyday life*. Cambridge, Mass.: MIT Press.

Koetsier J (2014) Mobile app monetization: Freemium is king, but in-app ads are growing fast. *VentureBeat*. Available from: http://venturebeat.com/2014/03/27/mobile-app-monetization-freemium-is-king-but-in-app-ads-are-growing-fast/ (accessed 28 March 2014).

Latour B (1984) The powers of association. *The Sociological Review,* 32: 264–280.

Latour B (2005) *Reassembling the social: An introduction to actor-network-theory*. Oxford; New York: Oxford University Press.

Law J (1992) Notes on the theory of the actor-network: Ordering, strategy, and heterogeneity. *Systems practice* 5(4): 379–393.

Lessig L (2001) *The future of ideas: The fate of the commons in a connected world*. New York: Random House.

Lessing L (2008) *Remix: Making art and commerce thrive in the hybrid economy*. New York: Penguin Press.

Levine D and Bartz D (2014) Oracle wins copyright ruling against Google over Android. *Reuters*, 9 May. Available from: http://www.reuters.com/article/us-oracle-google-ruling-idUSBREA480KQ20140509 (accessed 18 March 2016).

Levy S (1984) *Hackers: Heroes of the computer revolution,* 1st ed. Garden City, N.Y: Anchor Press/Doubleday.

Litman J (2006) *Digital copyright*. Amherst, NY: Prometheus Books.

Manovich L (2002) *The language of new media*. Leonardo, Cambridge, MA: MIT Press.

Manovich L (2008) Import/export. In: Fuller M (Ed.), *Software studies: A lexicon*, Cambridge, MA: MIT Press, pp. 119–124.

Manovich L (2013) *Software takes command*. International texts in critical media aesthetics, New York: Bloomsbury.

Mansell R (1999) New media competition and access: The scarcity-abundance dialectic. *New Media & Society* 1(2): 155–182.

Mansell R (2012) *Imagining the internet: Communication, innovation, and governance*. Oxford: Oxford University Press.

McChesney RW (2013) *Digital disconnect: How capitalism is turning the Internet against democracy*. New York: The New Press.

McGee M (2014) Bing ends 2013 with all-time high in US market share, but Google also up [comScore]. *Search Engine Land*. Available from: http://searchengineland.com/bing-ends-2013-with-all-time-high-in-us-market-share-but-google-also-up-comscore-181876 (accessed 1 April 2014).

McInerney PB (2009) Technology movements and the politics of free/open source software. *Science Technology Human Values* 34(2): 206–233.

Meinrath SD, Losey JW and Pickard VW (2010) Digital feudalism: Enclosures and erasures from digital rights management to the digital divide. *CommLaw Conspectus* 19: 423–479.

Mosco V (1996) *The political economy of communication: Rethinking and renewal*. London: Sage Publications.

Mosco V (2014) *To the cloud: Big data in a turbulent world*. Boulder, CO: Paradigm Publishers.

Mullin J (2012) Google wins crucial API ruling, Oracle's case decimated. *Ars Technica*. Available from: http://arstechnica.com/tech-policy/2012/05/google-wins-crucial-api-ruling-oracles-case-decimated/ (accessed 3 March 2016).

Panzarino M (2012) Apple tells developers not to duplicate the app store. *The Next Web*. Available from: http://thenextweb.com/apple/2012/10/01/apples-new-rule-developers-apps-promoting-apps-duplicating-app-store/ (accessed 2 April 2014).

Rooney B (2012) APIs: The power transforming the web. *The Wall Street Journal*. Available from: http://blogs.wsj.com/tech-europe/2012/10/04/apis-the-power-transforming-the-web/ (accessed 2 April 2014).

Samuelson PA (1954) The pure theory of public expenditure. *The Review of Economics and Statistics* 36(4): 387–389.

Schiller D (1999) *Digital capitalism: Networking the global market system*. Cambridge, Mass: MIT Press.

Shapiro RJ (2014) *The US software industry as an engine for economic growth and employment (SIIA White Paper)*. Software & Information Industry Association. Available from: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2541673 (accessed 31 December 2015).

Shirky C (2008) *Here comes everybody: The power of organizing without organizations*. New York: Penguin Press.

Smith A (1910) *The wealth of nations*. Reprint. London: J.M. Dent and Sons.

Sterne J (2012) *MP3: The meaning of a format*. SST: Sign, storage, transmission, Durham, NC: Duke University Press.

Tofel K (2015) Sorry Google, the Supreme Court won't appeal that massive Oracle case. *ZDNet*. Available from: http://www.zdnet.com/article/sorry-google-supreme-court-wont-appeal-that-massive-oracle-case/ (accessed 3 March 2016).

Warren C (2012) Twitter's API update cuts off oxygen to third-party clients. *Mashable*. Available from: http://mashable.com/2012/08/16/twitter-api-big-changes/ (accessed 2 January 2016).

Weber S (2004) *The success of open source*. Cambridge, MA: Harvard University Press.

Weber S (2005) The political economy of open source software and why it matters. In: Latham R and Sassen S (Eds), *Digital formations: Information technologies and new architectures in the global realm.* Princeton, NJ: Princeton University Press, pp. 178–211.

Wingfield N (2014) Microsoft to offer office for iPad, maybe a bit late. *The New York Times*, 27 March. Available from: http://www.nytimes.com/2014/03/28/technology/microsofts-office-suite-for-ipad.html (accessed 28 March 2014).

Zittrain J (2009) *The future of the internet–and how to stop it*. New Haven, CT: Yale University Press.