Social Epistemology
Vol. 26, Nos. 3–4, October 2012, pp. 379–398

The Social Epistemologies of Software

David M. Berry

This paper explores the specific questions raised for social epistemology encountered in code and software. It does so because these technologies increasingly make up an important part of our urban environment, and stretch across all aspects of our lives. The paper introduces and explores the way in which code and software become the conditions of possibility for human knowledge, crucially becoming computational epistemes, which we share with non-human but crucially knowledge-producing actors. As such, we need to take account of this new computational world and think about how we live today in a highly mediated code-based world. Nonetheless, here I want to understand software epistemes as a broad concept related to the knowledge generated by both human and non-human actors. The aim is to explore changes that are made possible by the installation of code/software via computational devices, streams, clouds or networks. This is what Mitcham calls a "new ecology of artifice". By exploring two case studies, the paper attempts to materialise the practice of software epistemologies through a detailed analysis. This analysis is then drawn together with a notion of compactants to explore how studying tracking software and streams is a useful means of uncovering the agency of software and code for producing these new knowledges.

Keywords: Computational; Epistemes; Software; Compactants

Introduction

This paper explores the relationship between knowledge, code and software. It does so because these technologies increasingly make up an important part of our urban environment, and indeed stretch even to very remote areas of the world. The paper introduces and explores the way in which code and software become the conditions of possibility for human living, crucially becoming computational ecologies, which we inhabit with non-human actors (see Fuller 2005). As such, we need to take

David M. Berry is Senior Lecturer in Digital Media at Swansea University, Swansea. Correspondence to: David M. Berry, Department of Political and Cultural Studies, COAH, Swansea University, Swansea, Wales, SA2 8PP, UK.Email: d.m.berry@swan.ac.uk

ISSN 0269-1728 (print)/ISSN 1464-5297 (online)
http://dx.doi.org/10.1080/02691728.2012.727191

account of this new computational world and think about how we understand knowledge in a highly mediated code-based world. Computer code and software are not merely mechanisms, they represent an extremely rich form of media (e.g. see Servin 2010). They differ from previous instantiations of media forms in that they are highly processual. They can also have agency delegated to them, which they can then prescribe back onto other actors, but which also remain within the purview of humans to seek to understand (see Giere 2007 for an interesting discussion of this). As Kitchin argues:

> The phenomenal growth in software creation and use is due to its emergent and executable properties: how it codifies the world into rules, routines, algorithms, and databases, and then uses these to do work in the world to render aspects of everyday life programmable. Whilst it is not fully sentient and conscious, software can exhibit some of the characteristics of "being alive" (Thrift and French, 2002). This property is significant because code enables technologies to do work in the world in an autonomous fashion——that is, it can process data, evaluate situations, and make decisions without human oversight or authorization. (Kitchin 2011, 945)

This deeply interactive characteristic of code and software, as computational media, makes it highly plastic for use in everyday life, and as such it has inevitably penetrated more and more into the lifeworld. This has created, and continues to create, specific tensions in relation to knowledge and understanding, as well as problems for managing and spectacularising the relations of the public to the science, economics and politics, etc. The key question for this paper is to explore the conditions under which groups of agents (from generations to societies) acquire, distribute, maintain and update (claims to) belief and knowledge through the active mediation of code/software. There is also considered to be a radical, if not revolutionary kernel within the softwarisation project linked to its ability to disseminate information and knowledge more broadly, for example displayed in the open source and open access movements (see Antonelli 2011; Berry 2008; de Laat 2011). This is due to the relative affordance code/software appears to give for individual autonomy within networks of association to share knowledge and communicate. Indeed, as Deuze, Blank, and Speers (2012) have argued:

> Considering the current opportunity a media life gives people to create multiple versions of themselves and others, and to endlessly redact themselves (as someone does with his/her profile on an online dating site in order to produce better matches), we now have a entered a time where … we can in fact see ourselves live, become cognizant about how our lifeworld is "a world of artifice, of bending, adapting, of fiction, vanity, a world that has meaning and value only for the man who is its deviser" [Pirandello 1990, pp. 39]. But this is not an atomized, fragmented, and depressing world, or it does not have to be such a world.

Of course, we should also be attentive to the oversharing or excessive collection of data too within these device ecologies that is outside of the control of the user to "redact themselves", as represented by the recent revelation of Path and Hipster that they were automatically harvesting user address book data (*BBC* 2012). Nonetheless, here I want to understand software ecologies as a broad concept related to

the environmental habitus of both human and non-human actors. The aim is to explore changes that are made possible by the installation of code/software via computational devices, streams, clouds or networks. This is what Mitcham (1998, 43) calls a "new ecology of artifice". The proliferation of contrivances that are computationally based is truly breathtaking, and each year we are given statistics that demonstrate how the new computational world is growing in size (see Pew 2012). These devices, of course, are neither static nor mute. Their inter-connections, communications, operation, effects and usage remain to be properly studied. Such study is made much more difficult both by the staggering rate of change, thanks to the underlying hardware technologies, which are becoming ever smaller, more compact, more powerful and less power-hungry; and by the increase in complexity, power, range and intelligence of the software that powers it. This state of affairs raises interesting new questions for an understanding of a social epistemology that includes both human and non-human actors, particularly where the reading of the new forms of digital writing are increasingly difficult for humans to access and understand directly.

Software/code also enables the assemblage of new social ontologies and the corresponding computational social epistemologies that we have increasingly grown to take for granted in computational society, including Wikipedia, Facebook and Twitter (see Dietrich 2008; Pfister 2011). The extent to which computational devices, and the computational principles on which they are based and from which they draw their power, have permeated the way in which we use and develop knowledges in everyday life is historically unique, if we had not already discounted and backgrounded its importance. For example, see Zax (2011) for the extent to which computational methods like n-gramming are being utilised to decode everyday life.[1] The ability to call up information instantly from a mobile device, combine it with others, subject it to debate and critique through real-time social networks, and then edit, post and distribute it worldwide would be incredible if it had not already started to become so mundane to us.[2]

Today, it should hardly come as a surprise that code/software lies as a mediator between ourselves and our corporeal experiences, disconnecting the physical world from a direct coupling with our physicality, whilst managing a looser softwarised transmission system. Called "fly-by-wire" in aircraft design, in reality fly-by-wire is the condition of the computational environment we increasingly experience, and I elsewhere term *computationality* (Berry 2011). This highly mediated existence has been a growing feature of the (post) modern world. Whilst many objects remain firmly material and within our grasp, it is easy to see how a more softwarised simulacra lies just beyond the horizon——the notion of "enchanted objects" has been coined to describe these new computationally enabled objects, for example. Not that software is not material, of course, certainly it is embedded in physical objects and the physical environment and requires a material carrier to function at all. Nonetheless, the materiality of software is without a doubt, *differently* material, more *tenuously* material, almost less *materially material* and this requires a careful description and critical attention. In part, this critical attention is necessary given

software's increasing tendency to hide its depths behind glass rectangular squares which yield only to certain prescribed forms of touch-based interfaces. Here, I am thinking both of physical keyboards and trackpads, as much as haptic touch interfaces, like those found in the iPad and other tablet computers. Another way of putting this, as N. Katherine Hayles (2004) has accurately observed, is that print is flat and code is deep, which calls attention to the need to look beneath the surface or screenic dimension of code-based devices.

Web Bugs, Beacons and Trackers

Some examples will help to demonstrate how this code-based world is increasingly being spun around us. Firstly, we might consider the growing phenomena of what are called "web bugs" (also known as "web beacons"), that is, computer programming code that is embedded in seemingly benign surfaces but which is actively and covertly collecting data and information about us.[3] As Madrigal (2012) explains:

> This morning, if you opened your browser and went to NYTimes.com, an amazing thing happened in the milliseconds between your click and when the news about North Korea and James Murdoch appeared on your screen. Data from this single visit was sent to 10 different companies, including Microsoft and Google subsidiaries, a gaggle of traffic-logging sites, and other, smaller ad firms. Nearly instantaneously, these companies can log your visit, place ads tailored for your eyes specifically, and add to the ever-growing online file about you … the list of companies that tracked my movements on the Internet in one recent 36°h period of standard web surfing: Acerno. Adara Media. Adblade. Adbrite. ADC Onion. Adchemy. ADiFY. AdMeld. Adtech. Aggregate Knowledge. AlmondNet. Aperture. AppNexus. Atlas. Audience Science … And that's just the As. My complete list includes 105 companies, and there are dozens more than that in existence.

Web bugs are automated data collection agents that are secretly included in the web pages that we browse. Often held within a tiny one-pixel frame or image, which is therefore far too small for the naked eye to see, they execute code to secrete cookies onto your computer so that they can track user behaviour, but also send various information about the user back to their servers.[4]

Originally designed as "HTTP state management mechanisms" in the early 1990s, these data storage processes were designed to enable webpages and sites to store the current collection of data about a user, or what is called "state" in computer science, known as "web bugs for web 1.0" (Dobias 2010, 245). They were aimed at allowing website designers to implement some element of memory about a user, such as a current shopping basket, preferences or username. It was a small step for companies to see the potential of monitoring user behaviour by leaving tracking information about browsing, purchasing and clicking behaviour through the use of these early "cookies".[5] The ability of algorithms to track behaviour, collect data and information about users raises important privacy implications but also facilitate the rise of so-called behaviour marketing and nudges (see Eyal 2012 for a behaviourist approach). These technologies have become much more

sophisticated in the light of Web 2.0 technologies and developments in hardware and software, in effect web bugs for web 2.0 (Dobias 2010, 245).

Fortunately, we are seeing the creation of a number of useful software projects that allow us to track the trackers, Collusion, Foxtracks and Ghostery, for example.[6] If we look at the Ghostery log for the ChartBeat company (http://chartbeat.com) it is described as:

> Provid[ing] real-time analytics to web sites and blogs. The interface tracks visitors, load times, and referring sites on a minute-by-minute basis. This allows real-time engagement with users giving publishers an opportunity to respond to social media events as they happen. ChartBeat also supports mobile technology through APIs. (Ghostery 2012b)

Web bugs perform these analytics by running code run in the browser usually without the knowledge of the user, which if it should be observed, looks extremely complicated.[7] Here are two early web bugs (web 1.0) collected by the Electronic Frontier Foundation EFF (1999):

```
<img     src=http://ad.doubleclick.net/ad/pixel.quicken/NEW
width=l height=l border=0>
```

```
<IMG WIDTH=l HEIGHT=l border=0
SRC=http://media.preferences.com/ping?ML_SD=IntuitTE_ Intuit_
lxl_RunOfSite_Any  &db_afcr=4B3l-C2FB-lOE2C&event=reghome
&group=register& time=l999.lO.27.20.5 6.37>
```

Later web bugs (web 2.0) are not included here due to the complexity and length of the code (but see the 3rd-party elements or "3pes" at http://www.knowyourelements.com/).[8] This code is noticeably opaque and difficult to understand, even for experienced computer programmers. Indeed, one suspects an element of obfuscation, a programming technique to reduce the readability of the code and which is used to essentially shield the company from observation. So far in checking a number of web bugs on a variety of websites, I have been unable to find one that supplies any commentary on what exactly the code is doing, beyond a short privacy policy statement. Again Ghostery (2012b) usefully supplies us with some general information on the web bug, such as the fact that it has been found on over 100,000 websites across the Internet and that the data collected are "anonymous (browser type), pseudonymous (IP address)", the data are not shared with third parties but no information is given on their data retention policies. As at 2 March 2012, Ghostery reported that it was tracking 829 different web bugs across the Internet. In a separate study, Kennish (2011) found 6926 third-party web-bug code fragments on 201,358 web pages from the top 1000 websites which he breaks into (i) Social networking services, which use individual identifiable data; (ii) Advertising/analytics/content services, which usually collect anonymous data; and (iii) Duplicate services, e.g. other unrelated services (see Efrati 2011; Milian 2011). This is a relatively unregulated market in user behaviour,

tracking and data collection, which currently has a number of self-regulatory bodies, such as the Network Advertising Initative (NAI). As Madrigal reports:

> In essence, [the NAI] argued that users do not have the right to *not* be tracked. "We've long recognized that consumers should be provided a choice about whether data about their likely interests can be used to make their ads more relevant," [they] wrote. "But the NAI code also recognizes that companies sometimes need to continue to collect data for operational reasons that are separate from ad targeting based on a user's online behavior."… Companies "need to continue to collect data," but that contrasts directly with users desire "not to be tracked." (Madrigal 2012)

These web bugs, beacons, pixels and tags, as they are variously called, form part of the darknet surveillance network that users rarely see even though it is profoundly changing their experience of the internet in real time by attempting to second guess, tempt, direct and nudge behaviour in particular directions.[9] Ghostery ranked the web bugs in 2010 and identified these as the most frequently encountered (above average): Revenue Science (250x), OpenX (254x), AddThis (523.6x), Facebook Connect (529.8x), Omniture (605.7x), Comscore Beacon (659.5x), DoubleClick (924.4x), QuantCast (1042x), Google Adsense (1452x) and Google Analytics (3904.5x) (Ghostery 2011). As can be seen in terms of relative size of encounter, Google is clearly the biggest player in the area of the collection of user statistics by a long distance. These data are important because, as JP Morgan's Imran Khan explained, a unique visitor to each website at Amazon (e-commerce) is generating $189 per user, at Google (search) it is generating $24 per user and although Facebook (social networking) is only generating $4 per user, this is a rapidly growing number (Yarrow 2011). Keeping and holding these visitors, through real-time analytics, customer history, behavioural targeting, etc. is increasingly extremely profitable. Indeed, Amazon has calculated that knowing and responding to customer needs is very important for profitability and "that a page load slowdown of just one second could cost it $1.6 billion in sales each year" (Eaton 2012). Correspondingly, "Google has calculated that by slowing its search results by just four tenths of a second they could lose 8 million searches per day–meaning they'd serve up many millions fewer online adverts", and hence make less money (Eaton 2012).

Where companies are more explicitly collecting data and information they often have in place data collection and privacy policies, for example see Facebook (2012) or Google (2012). An analysis by Cranor and McDonald (2008) found that it would take on average 201 hours per year to read privacy policies that users find in their everyday use of the Internet and which are extremely complicated legal documents. Unsurprisingly, few read them. Users are therefore often agreeing to certain data usage, collection, reselling and aggregation without explicitly being aware of it. For example, whilst you are logged in, Facebook collects,

> a timestamped list of the URLs you visit and pairs it with your name, list of friends, Facebook preferences, email address, IP address, screen resolution, operating system, and browser. When you're logged out, it captures everything except your name, list of

> friends, and Facebook preferences. Instead, it uses a unique alphanumeric identifier to track you. (Love 2012)

Ghostery (2010) has performed a useful analysis of their web bug database that attempts to categorise the web bugs found into 16 different types, which I have re-categorised into four main types, (1) Advertiser/Marketing Services, (2) Analysis/Research Services, (3) Management Platforms and (4) Verification/Privacy Services:

(1) Advertiser/Marketing Services:

  (a) Advertiser: A company sponsoring advertisement and ultimately responsible for the message delivered to the consumer. Example: *AT&T*
  (b) Exchange: A provider of marketplace connecting advertisers to ad networks and data aggregators (online and off), often facilitating multiple connections and bidding processes. Example: *Right Media*
  (c) Network: A broker and often technology provider connecting advertisers and publishers. (website operators) Example: *Burst Media*
  (d) Publisher: Website operator who displays ads for advertiser(s) in various types campaigns. Example: *The New York Times*

(2) Analysis/Research Services:

  (a) Online Data Aggregator: Collects data from online publishers and provides it to advertisers either directly or via exchange. Example: BlueKai
  (b) Offline Data Aggregator: Collects data from a range of offline sources and provides data to advertisers directly or via exchange. Experian
  (c) Optimiser: Provider of analytics technology and services for ROI assessment and content optimisation purposes. Example: ROILabs
  (d) Research: Collects data for market research purposes where no ads are serviced through this data. Example: Example: Safecount
  (e) Analytics Provider: Provider of cross-platform statistical analysis to understand market effectiveness and audience segmentation. Example: Google Analytics
  (f) Retargeter: Providers of technologies that allow publishers to identify their visitor when they place ads on third party sites. Example: FetchBack

(3) Management Platforms:

  (a) Demand-Side Platform: A technology provider that allows marketers to buy inventory across multiple platforms or exchanges. DSPs often layer in custom optimisation, audience targeting, real-time bidding and other services. Example: Invite Media
  (b) Supply-Side Platform: A technology provider that allows publishers to access advertiser demand across multiple platforms or exchanges. SSPs

often layer in custom yield optimisation, audience creation, real-time bidding and other services. Example: AdMeld

(c) Ad Server: Technology that delivers and tracks advertisements independently of the web site where the ad is being displayed. Example: DoubleClick DART

(d) Agency: Provider of creative and buying services (both audience and data) for advertisers. Example: MediaCom

(4) Verification/Privacy Services:

(a) Ad Verification: Certifies or classifies webpages in an effort to prevent advertisers' campaigns from running on unsavory or blocked content, and/or protects advertisers from having other companies run their ads incorrectly. Example: ClickForensics

(b) Online Privacy: Technology providers that deliver information and transparency to consumers on how third-party companies gather and use their data. Example: Better Advertising

Ghostery gives a useful explanation of how these companies interoperate to perform and variety of services for advertising and marketing clients:

> A company like Turn Media is a technology provider that allows marketers to buy inventory across multiple platforms or exchanges, or a Demand-Side Platform. They provide services for marketers and agencies to centrally manage buying, planning, targeting, and optimizing media opportunities. Reasonably speaking, however, you could also technically classify them as an Optimizer because this process is included under the umbrella of the platform. Turn [Media] is deeply data driven and partners with multiple data providers including BlueKai, TargusInfo, eXelate, and others. (Ghostery 2010)

Of course, one element missing from this typology is that of surveillance, and indeed it is no surprise that web bugs perform part of the tracking technologies used by companies to monitor staff. For example, in 2006, Hewlett Packard used web bugs from readnotify.com to trace insider leaks to the journalist Dawn Kawamoto and later confirmed in testimony to a US House of Representatives subcommittee that it is "still company practice to use e-mail bugs in certain cases" (Evers 2006; Fried 2006).

As can be seen, this extremely textured environment currently offers little in terms of diagnosis or even warnings to the user. The industry itself, which prefers the term "clear GIF" to web bug, certainly is keen to avoid regulation and keeps itself very much to itself in order to avoid raising too much unwarranted attention. Some of the current discussions over the direction of regulation on this issue have focused on the "do not track" flag, which would signal a user's opt-out preference within an HTTP header. Unfortunately, very few companies respect the do not track header and there is currently no legal requirement that they do so in the USA, or elsewhere (W3C 2012). There have been some moves towards *self-regulation* in the technology industry with a recent report from the US Federal Trade

Commission (Tsukayama 2012). Although see the current debate over the EU ePrivacy Directive where the Article 29 Working Party (A29 WP) has stated that "voluntary plans drawn up by Europe's digital advertising industry representatives, the European Advertising Standards Alliance (EASA) and IAB Europe, do not meet the consent and information requirements of the recently revised ePrivacy Directive" (Baker 2012). Legislation may therefore be introduced into the EU before elsewhere.

One of the newer, and perhaps indicative direction of travel of these new web bugs under development is called PersianStat (http://www.persianstat.ir/), which claims "an eye on 1091622 websites", an Iranian web tracking and data analytics website which shows that this new code ecology is not purely a western phenomenon. With the greater use of computational networked devices in everyday life, from mobile phones to GPS systems, these forms of tracking systems will only become more invasive and more aggressive in collecting data from our everyday life and encounters. Indeed, it is unsurprisingly to find that Americans, for example, are not comfortable with the growth in use of these tracker technologies, Pew (2012) found,

> that 73 percent of Americans said they would "not be okay" with being tracked (because it would be an invasion of privacy) … Only 23 percent said they'd be "okay" with tracking (because it would lead to better and more personalized search results) … Despite all those high-percentage objections to the idea of being tracked, less than half of the people surveyed – 38 percent – said they knew of ways to control the data collected about them. (Garber 2012; Pew 2012)

This contradiction between the ability of these computational systems and surfaces to supply a commodity to the user, and the need to raise income through the harvesting of data which is in turn sold to advertisers and marketing companies shows that this is an unstable situation. It also serves to demonstrate the extent to which users are just not aware of the subterranean depths of their computational devices and the ability for these general computing platforms to disconnect the user interface from the actual intentions or functioning of the device, whilst giving the impression to the user that they remain fully in control of the computer. As Garber observes,

> underground network, surface illusion … How much do we actually want to know about this stuff? Do we truly want to understand the intricacies of data-collection and personalisation and all the behind-the-screen work that creates the easy, breezy experience of search … or would we, on some level, prefer that it remain as magic? (Garber 2012)

Indeed, as Aron (2012) reports, "up to 75 per cent of the energy used by free versions of Android apps is spent serving up ads or tracking and uploading user data". That is, on free versions of popular apps most of the processing work in the app is spent monitoring user activities and sending it back home to servers (see also Pathak, Hu, and Zhang 2012). This ability for code/software to monitor the user covertly and even obscure its processing activities will undoubtedly become a

growing political and economic as well as technical issue (see some examples from Goodale 2012).[10]

The increased ability of software and code via computational devices to covertly monitor, control and mediate, both positively and negatively, is not just a case of interventions for deceiving the human and non-human actors that make up part of these assemblages. In the next section, I want to look at the willing compliance with data collection, indeed the enthusiastic contribution of real-time knowledge and data to computational systems as part of the notion of lifestreams, and more particularly the quantified self-movement.

Lifestreams

There has been over the past decade a growth in the use of self-monitoring technologies called lifestreaming, often under the rubric of the "quantified self".[11] This movement has exploded in recent years as the "real-time streams" platforms have expanded, like Twitter and Facebook (see Bucher 2012, for a discussion of the EdgeRank algorithm, for example). Indeed, some argue that

> we're finally in a position where people volunteer information about their specific activities, often their location, who they're with, what they're doing, how they feel about what they're doing, what they're talking about… We've never had data like that before, at least not at that level of granularity. (Rieland 2012)

This has been usefully described by *The Economist*, who argued that the,

> idea of measuring things to chart progress towards a goal is commonplace in large organisations. Governments tot up trade figures, hospital waiting times and exam results; companies measure their turnover, profits and inventory. But the use of metrics by individuals is rather less widespread, with the notable exceptions of people who are trying to lose weight or improve their fitness … But some people are doing just these things. They are an eclectic mix of early adopters, fitness freaks, technology evangelists, personal-development junkies, hackers and patients suffering from a wide variety of health problems. What they share is a belief that gathering and analysing data about their everyday activities can help them improve their lives—an approach known as "self-tracking", "body hacking" or "self-quantifying." (Economist 2012)

This phenomenon of using computational devices to monitor health signals and to feed them back into calculative interfaces, data visualisations, real-time streams, etc. is the next step in social media. This practice closes the loop of personal information online, which, although it remains notionally private, is stored and accessed by corporations who wish to use this biodata for data mining and innovation surfacing. For example:

> The Zeo [headband], for example, has already generated the largest-ever database on sleep stages, which revealed differences between men and women in REM-sleep quantity. Asthmapolis also hopes to pool data from thousands of inhalers fitted with its Spiroscout [asthma inhaler] sensor in an effort to improve the management of asthma.

> And data from the Boozerlyzer [alcohol counting] app is anonymised and aggregated to investigate the variation in people's response to alcohol. (Economist 2012)

Lifestreams were originally an idea from David Gelernter and Eric Freeman in the 1990s (Freeman 1997; Gelernter 2010), which they described as:

> A *lifestream* is a time-ordered stream of documents that functions as a diary of your electronic life; every document you create and every document other people send you is stored in your lifestream. The tail of your stream contains documents from the past (starting with your electronic birth certificate). Moving away from the tail and toward the present, your stream contains more recent documents — papers in progress or new electronic mail; other documents (pictures, correspondence, bills, movies, voice mail, software) are stored in between. Moving beyond the present and into the future, the stream contains documents you *will* need: reminders, calendar items, to-do lists. You manage your lifestream through a small number of powerful operators that allow you to transparently store information, organize information on demand, filter and monitor incoming information, create reminders and calendar items in an integrated fashion, and "compress" large numbers of documents into overviews or executive summaries (Freeman 2000).

Gelernter originally described these "chronicle streams" (Gelernter 1994), highlighting both their narrative and temporal dimensions related to the storage of documentation and texts. Today, we are more likely to think of them as "real-time streams" and the timeline functions offered by systems like Twitter, Facebook and Google+. These are increasingly the model of interface design that is driving the innovation in computation, especially in mobile and locative technologies. However, in contrast to the document-centric model that Gelernter and Freeman were describing, there are also the micro-streams of short updates, epitomised by Twitter, which has short text message sized 140 character updates. Nonetheless, this is still enough text space to incorporate a surprising amount of data, particularly when geo, image, weblinks and so forth are factored in. Stephen Wolfram was certainly one of the first people to collect their data systematically, as he explains he started in 1989:

> So email is one kind of data I've systematically archived. And there's a huge amount that can be learned from that. Another kind of data that I've been collecting is keystrokes. For many years, I've captured every keystroke I've typed——now more than 100 million of them. (Wolfram 2012)

This kind of self-collection of data is certainly becoming more prevalent and in the context of reflexivity and self-knowledge, it raises interesting questions. The scale of data that is collected can also be relatively large and unstructured.[12] Nonetheless, better data management and techniques for searching and surfacing information from unstructured or semi-structured data will no doubt be revealing about our everyday patterns in the future.[13]

This way of collecting and sending data has been accelerated by the use of mobile "apps", which are small relatively contained applications that usually perform a single specific function. For example, the Twitter app on the iPhone allows the user to send updates to their timeline, but also search other timelines, check

out profiles, streams and so on. When created as apps, however, they are also able to use the power of the local device, especially if it contains the kinds of sophisticated sensory circuitry that is common in smartphones, to log GPS geographic location, direction, etc. In this instance, lifestreaming becomes increasingly similar to the activity of web bugs in monitoring and collecting data on the users that are active on the network. Indeed, activity streams have become a standard which is increasingly being incorporated into software across a number of media and software practices (see ActivityStreams n.d.). An activity stream essentially encodes a user event or activity into a form that can be computationally transmitted and later aggregated, searched and processed,

> In its simplest form, an activity consists of an *actor*, a *verb*, an *object*, and a *target*. It tells the story of a person performing an action on or with an object – "Geraldine posted a photo to her album" or "John shared a video". In most cases these components will be explicit, but they may also be implied. (ActivityStreamsWG 2011, original emphasis)

This data and activity collection is only part of the picture, however. In order to become reflexive data, it must be computationally processed from its raw state, which may be structured, unstructured or a combination of the two. At this point, it is common for the data to be visualized, usually through a graph or timeline, but there are also techniques such as heat-maps, graph theory and so forth that enable the data to be processed and reprocessed to tease out patterns in the underlying data-set. In both the individual and aggregative use case, in other words for the individual user (or lifestreamer) or organisation (such as Facebook), the key is to pattern match and compare details of the data, such as against a norm, a historical data-set or against a population, group or class or others.[14]

The patterned usage is therefore a dynamic real-time feedback mechanism in terms of providing steers for behaviour, norms and so forth, but also offering a documentary narcissism that appears to give the user an existential confirmation and status. Even in its so-called gamification forms, the awarding of competitive points, badges, honours and positional goods more generally is the construction of a hierarchical social structure within the group of users. It also encourages the user to think of themselves as a set of partial objects, fragmented dividuals or loosely connected properties, collected as a time series of datapoints, and subject to intervention and control. This can be thought of as a computational care of the self, facilitated by an army of oligopticans (Latour 2005) in the wider computational environment that observe and store behavioural and affective data. However, this self is reconciled through the code and software that makes the data make sense. The code and software are therefore responsible for creating and maintaining the meaning and narratives through a stabilisation and web of meaning for the actor.[15]

Conclusions

I now want to turn to how we might draw these case studies together to think about the social epistemologies generated and maintained by code and software and the implications for wider study in terms of research and the theorisation of these issues. It seems to me that a connecting thread runs through web bugs and lifestreaming itself: social knowledge collection, monitoring and real-time feedback, whether overt or covert. Whilst we can continue to study these phenomena in isolation, and indeed there can be very productive knowledge generated from this kind of research, it seems to me that we need to attend to the knowledge represented in code and software to better understand software ecologies such as these (Berry 2011).

One of the most interesting aspects to these systems is that humans in many cases become the vectors that enable the data transfers, whilst also becoming the vectors that carry the data that fuel the computational economy. Our movements between systems, carrying USB sticks and logging into email accounts and distant networks creates the channels through which the data flow, or an infection is spread. The ability of viruses to take on some of the features of web bugs and learn our habits and preferences in real time whilst secreting themselves within our computer systems raise important questions (see Fields 2008). However, users are actively downloading apps that advertise the fact that they collect these data and seem to genuinely find an existential relief or recognition in their movements being recorded and available for later playback or analysis. Web bugs, in many ways, are lifestreams——albeit lifestreams that have not been authorised by the user whom they are monitoring. This collection of what we might call *compactants* is designed to *passive-aggressively* record data.[16] With the notion of *compactants* (computational actants), I want to particularly draw attention to this passive-aggressive feature of computational agents that are collecting information. Both in terms of their passive quality——under the surface, relatively benign and silent——but also the fact that they are aggressive in their hoarding of data——monitoring behavioural signals, streams of affectivity and so forth.[17] The word *compact* also has useful overtones of having all the necessary components or functions neatly fitted into a small package, and compact as in conciseness in expression. The etymology from the Latin *compact* for closely put together, or joined together, also nearly expresses the sense of what web bugs and related technologies are. The term compactants are also useful in terms of the notion of *companion actants* (see Harraway 2003).

Interestingly, compactants are structured in such a way that they can be understood as having a dichotomous structure of data collection/visualisation, each of which is a specific mode of operation. Naturally, due to the huge quantities of data that are often generated, the computational processing and aggregation is often offloaded to the "cloud", or server computers designed specifically for the task and accessed via networks. Indeed, many viruses, for example, often seek to "call home" to report their status, upload data or offer the chance of being updated,

perhaps to a more aggressive version of themselves or to correct bugs. As Kitchin (2011, 945) explains:

> As a result, across a diverse set of everyday tasks, domestic chores, work, shopping, travelling, communicating, governing, and policing, software makes a difference to how social, spatial, and economic life takes place. Such is software's capacities and growing pervasiveness that some analysts predict that we are entering a new phase of "everyware" (Greenfield, 2006); that is, computational power will be distributed and available at any point on the planet.

We might also think about the addressee of these wider computational systems made up of arrays or networks of compactants, which in many cases is a future actor. Within the quantified self-movement, there is an explicit recognition that the "future self" will be required to undo bad habits and behaviours of the present self. That is, that there is an explicit normative context to a *future* self, who you, as the *present* self may be treating unfairly, immorally or without due regard to, what has been described as "future self continuity" (Tugend 2012). This inbuilt tendency towards the *futural* is a fascinating reflection of the internal temporal representation of time within computational systems that is time series structured streams of real-time data and knowledge, often organised as structured lists. Therefore, the past (as stored data), present (as current data collection, or processed archival data) and future (as both the ethical addressee of the system and potential provider of data and usage) are often deeply embedded in the code that runs these systems. In some cases, the future also has an objective existence as a probabilistic projection, literally a *code-object*, which is updated in real time and which contains the major features of the future state represented as a model; computational weather prediction systems and climate change models are both examples of this.

There are many examples of how attending to the code and software that structures many of the life, memory and biopolitical systems and industries of contemporary society could yield similarly revealing insights into both our usage of code and software, but also the structuring assumptions, conditions and affordances that are generated.[18] Our use of computational models is growing, and our tendency is to confuse the screenic representation visualised by code/software with what we might call the real——not to mention our failure to appreciate the ways in which code's mediation is co-constructive of, and deeply involved in, the stabilisation of everyday social knowledge that we take for granted. Even so, within institutional contexts, code/software has not fully been incorporated into the specific logics of these social systems, and in many ways also undermines these structural and institutional forms.[19] We must remain attentive to the fact that software engineering itself is a relatively recent discipline and its efforts at systematisation and rationalisation are piecemeal and incomplete, as the many hugely expensive software system failures attests. Of course, this code/software research is not easy, the techniques needed are still in their infancy, and whilst drawing on a wide range of scholarly work from the sciences, social sciences and the arts and humanities, we are still developing our understanding as to what exactly is the function of

software-based knowledge in society. But this should give hope and direction to the critical theorists, both of the present looking to provide critique and counterfactuals, but also *of* the future, as code/software is a particularly rich site for intervention, contestation and the *unbuilding* of code/software systems towards greater understanding of the social epistemologies of software.[20]

Notes

[1] An n-gram is a list of "n" items from a given sequence of textual materials or speech. The basic units can be letters, words, syllables, etc. Google n-gram viewer is a good example of using this technique to search textual corpora: http://books.google.com/ngrams.

[2] Naturally this includes all forms of knowledge encoded as textual, video, film, photography and so forth. New forms of "mash-up" data platforms are constantly emerging, such as Google Maps and Mixel, which enable very sophisticated knowledge and metadata to be combined in interesting ways.

[3] These include HTTP cookies and Locally Stored Objects (LSOs) and document object model storage (DOM Storage).

[4] Here, I am concerned with the collection of data through web bugs and bracket out other kinds of "malware" such as botnet, Trojan, viruses and so forth. An interesting Q&A with a botnet hacker, which demonstrates the extent of vulnerability of the average computer user is described in throwaway236236 (2012).

[5] Cookies are small pieces of text that servers can set and read from a client computer in order to register its "state". They have strictly specified structures and can contain no more than 4 KB of data each. When a user navigates to a particular domain, the domain may call a script to set a cookie on the user's machine. The browser will send this cookie in all subsequent communication between the client and the server until the cookie expires or is reset by the server (Mittal 2010, 10).

[6] Ghostery describes itself on its help page: "Be a web detective. Ghostery is your window into the invisible web – tags, web bugs, pixels and beacons that are included on web pages in order to get an idea of your online behavior. Ghostery tracks the trackers and gives you a roll-call of the ad networks, behavioral data providers, web publishers, and other companies interested in your activity" (Ghostery 2012a). See also https://disconnect.me/.

[7] For an example see, http://static.chartbeat.com/js/chartbeat.js.

[8] Also see examples at: (1) Chartbeat: http://static.chartbeat.com/js/chartbeat.js; (2) Google Analytics: http://www.google-analytics.com/ga.js; (3) Omniture: http://o.aolcdn.com/omni-unih.js and (4) Advertising.com: http://o.aolcdn.com/ads/adsWrapper.js.

[9] For example the page-scraping of data from open access web pages using "robots" or "spiders" in order to create user repositories of data through aggregation and data analysis. Interestingly this is the way in which Google collects the majority of the index data it uses for its search results. This is also becoming a digital method in the social sciences and raises particular digital research ethics that have still to be resolved, see https://www.issuecrawler.net/, http://socscibot.wlv.ac.uk/, http://webatlas.fr/wp/navicrawler/.

[10] See these commercial examples of user control software for controlling user public exposure to trackers, web bugs and compactants, although the question is raised as to why you would choose to trust them: https://cloudcapture.org/register/ and http://www.abine.com.

[11] See http://quantifiedself.com/.

[12] An example of pre-computational collection of data about the self as a lifestream is represented by Roberts (2004). One of the criticisms that recur in the peer-review section is that Roberts fails to account for his own anticipation of his experimentation and previous experimentation colouring his results. Nonetheless, this kind of self-knowledge through

collection is made both easier, and arguably more rigorous by the collection through *compactants*. Especially, if the collection is of wide rather than narrow width, it enables a *post hoc* analysis and hypothesis surfacing to occur. Clearly, compactants also make the collection far easier with mobile devices.

[13] Wolfram further writes: "It's amazing how much it's possible to figure out by analyzing the various kinds of data I've kept. And in fact, there are many additional kinds of data I haven't even touched on in this post. I've also got years of curated medical test data (as well as my not-yet-very-useful complete genome), GPS location tracks, room-by-room motion sensor data, endless corporate records——and much much more … And as I think about it all, I suppose my greatest regret is that I did not start collecting more data earlier. I have some backups of my computer filesystems going back to 1980. And if I look at the 1.7 million files in my current filesystem, there's a kind of archeology one can do, looking at files that haven't been modified for a long time (the earliest is dated June 29, 1980)" (Wolfram 2012).

[14] Some examples of visualisation software for this kind of lifestreaming quantification and visualisation are shown on these pages from the Quantified Self-website: Personal Data Visualisation, Jaw-Dropping Infographics for Beginners, A Tour Through the Visualisation Zoo, Visual Inspiration.

[15] See http://open.sen.se/ for a particularly good example of this: "Make your data history meaningful. Privately store your flows of information and use rich visualizations and mashup tools to understand what's going on" (Sense 2012).

[16] Computational actants, drawing the notion of actant from actor–network theory.

[17] Of course compactants are not just "internal" data collection agents. They may also be outside of your data resources and networks probing to get in, this kind of unauthorised access to personal data is on the rise and has been termed the industrialisation of data theft (see Fulton 2012). Indeed, Fulton argues that "scripts, bots, and other non-social means for obtaining access [to data] remains statistically more effective than direct, personal contact - although even these automated means remain astoundingly simple" (Fulton 2012).

[18] For example see Bamford (2012) who writes about the Utah Data Center is being built for the National Security Agency: "A project of immense secrecy, it is the final piece in a complex puzzle assembled over the past decade. Its purpose: to intercept, decipher, analyze, and store vast swaths of the world's communications as they zap down from satellites and zip through the underground and undersea cables of international, foreign, and domestic networks. The heavily fortified $2 billion center should be up and running in September 2013. Flowing through its servers and routers and stored in near-bottomless databases will be all forms of communication, including the complete contents of private emails, cell phone calls, and Google searches, as well as all sorts of personal data trails——parking receipts, travel itineraries, bookstore purchases, and other digital 'pocket litter'. It is, in some measure, the realization of the 'total information awareness' program created during the first term of the Bush administration——an effort that was killed by Congress in 2003 after it caused an outcry over its potential for invading Americans' privacy" (Bamford 2012).

[19] What we might call "outsider code" or "critical code" is an interesting development in relation to this. A number of websites offer code that data-scrapes, or screen-scrapes information to re-present and analyse it for the user, some examples include: (1) Parltrack software, which is designed to improve the transparency of the EU parliamentary legislative process, http://parltrack.euwiki.org/ and (2) TheyWorkForYou, which screen-scrapes the UK Parliamentary minutes, Hansard, http://www.theyworkforyou.com/.

[20] Here I tentatively raise the suggestion that a future critical theory of code and software is committed to *un*building, *dis*assembling, and *de*formation of existing code/software

systems, together with a necessary intervention in terms of a positive moment in the formation and composition of future and alternative systems.

References

ActivityStreams. n.d. Activity streams [cited 4 March 2012]. Available from http://activitystrea.ms/; INTERNET.

ActivityStreamsWG. 2011. JSON Activity Streams 1.0, Activity Streams Working Group [cited 4 March 2012]. Available from http://activitystrea.ms/specs/json/1.0/; INTERNET.

Antonelli, P. 2011. States of design 03: Thinkering, *domus* [cited 28 March 2012]. Available from http://www.domusweb.it/en/design/states-of-design-03-thinkering-/; INTERNET.

Aron, J. 2012. Free apps eat up your phone battery just sending ads. *New Scientist* [cited 28 March 2012]. Available from http://www.newscientist.com/article/mg21328566.400-free-apps-eat-up-your-phone-battery-just-sending-ads.html; INTERNET.

Baker, J. 2012. European watchdog pushes for do not track protocol [cited 10 March 2012]. Available from http://www.pcworld.com/businesscenter/article/251373/european_watchdog_pushes_for_do_not_track_protocol.html; INTERNET.

Bamford, J. 2012. The NSA is building the country's biggest spy center (Watch What You Say). *Wired* [cited 19 March 2012]. Available from http://www.wired.com/threatlevel/2012/03/ff_nsadatacenter/all/1; INTERNET.

BBC. 2012. iPhone apps Path and Hipster offer address-book apology. *BBC* [cited 28 March 2012]. Available from http://www.bbc.co.uk/news/technology-16962129; INTERNET.

Berry, D. M. 2008. *Copy, rip, burn: The politics of copyleft and open source*. London: Pluto Press.

————. 2011. *The philosophy of software: Code and mediation in the digital age*. London: Palgrave.

Bucher, T. 2012. Want to be on the top? Algorithmic power and the threat of invisibility on facebook. *New Media and Society*, 1–17 [cited 22 April 2012]. Available from http://nms.sagepub.com/content/early/2012/04/04/1461444812440159.abstract; INTERNET.

Cranor, L. F., and A. M. McDonald. 2008. The cost of reading privacy policies. *I/S: A Journal of Law and Policy for the Information Society*, 2008 Privacy Year in Review issue [cited 26 March 2012]. Available from http://lorrie.cranor.org/pubs/readingPolicyCost-authorDraft.pdf; INTERNET.

de Laat, P. B. 2011. Open source production of encyclopedias: Editorial policies at the intersection of organizational and epistemological trust. *Social Epistemology: A Journal of Knowledge, Culture and Policy* 26 (1): 71–103.

Deuze, M., P. Blank, and L. Speers. 2012. A life lived in media. *Digital Humanities Quarterly* 6 (1) (Winter) [cited 29 February 2012]. Available from http://digitalhumanities.org/dhq/vol/6/1/000110/000110.html; INTERNET.

Dietrich, E. 2008. Computationalism. *Social Epistemology: A Journal of Knowledge, Culture and Policy* 4 (2): 135–54.

Dobias, J. 2010. Privacy effects of web bugs amplified by Web 2.0. In *Privacy and identity management for life*, edited by S. Fischer-Hübner, P. Duquenoy, M. Hansen, R. Leenes, and G. Zhang, pp. 244–57. London: Springer.

Eaton, K. 2012. How one second could cost Amazon $1.6 billion in sales, fast company. *Fast Company* [cited 24 March 2012]. Available from http://www.fastcompany.com/1825005/impatient-america-needs-faster-intertubes; INTERNET.

Economist. 2012. Counting every moment. *The Economist* [cited 2 March 2012]. Available from http://www.economist.com/node/21548493; INTERNET.

EFF. 1999. The web bug FAQ [cited 2 March 2012]. Available from http://w2.eff.org/Privacy/Marketing/; INTERNET.

Efrati, A. 2011. "Like" button follows web users. *Wall Street Journal* [cited 2 March 2012]. Available from http://online.wsj.com/article/SB10001424052748704281504576329441432995616.html; INTERNET.

Evers, J. 2006. How HP bugged e-mail [cited 2 March 2012]. Available from http://news.cnet.com/How-HP-bugged-e-mail/2100-1029_3-6121048.html; INTERNET.

Eyal, N. 2012. How to manufacture desire, TechCrunch [cited accessed 5 March 2012]. Available from http://techcrunch.com/2012/03/04/how-to-manufacture-desire/; INTERNET.

Facebook. 2012. Data use policy [cited 2 March 2012]. Available from http://www.facebook.com/about/privacy/; INTERNET.

Fields, C. 2008. Human-computer interaction: A critical synthesis. *Social Epistemology: A Journal of Knowledge, Culture and Policy* 1 (1): 5–25.

Freeman, E. T. 1997. The lifestreams software architecture. Ph.D. Dissertation, Yale University Department of Computer Science, May 1997 [cited 2 March 2012]. Available from http://www.cs.yale.edu/homes/freeman/dissertation/etf.pdf; INTERNET.

————. 2000. Welcome to the yale lifestreams homepage! [cited 9 March 2012]. Available from http://cs-www.cs.yale.edu/homes/freeman/lifestreams.html; INTERNET.

Fried, I. 2006. Dunn grilled by congress [cited 2 March 2012]. Available from http://news.cnet.com/Dunn-grilled-by-Congress/2100-1014_3-6120625.html; INTERNET.

Fuller, M. 2005. *Media ecologies: Materialist energies in art and technoculture*. Cambridge, MA: MIT Press.

Fulton, S. M. 2012. The industrialization of data theft: Verizon's staggering new data. *ReadWrite Enterprise* [cited 26 March 2012]. Available from http://www.readwriteweb.com/enterprise/2012/03/the-industrialization-of-data.php?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+readwriteweb+%28ReadWriteWeb%29; INTERNET.

Garber, M. 2012. Americans love google! Americans hate google! *The Atlantic* [cited 2 March 2012] Available from http://m.theatlantic.com/technology/archive/2012/03/americans-love-google-americans-hate-google/254253/; INTERNET.

Gelernter, D. 1994. The cyber-road not taken. *The Washington Post*, April.

————. 2010. Time to start taking the internet seriously. *The Edge* [cited 2 March 2012]. Available from http://www.edge.org/3rd_culture/gelernter10/gelernter10_index.html; INTERNET.

Giere, R. N. 2007. Distributed cognition without distributed knowing. *Social Epistemology: A Journal of Knowledge, Culture and Policy* 21 (3): 313–20.

Ghostery. 2010. The many data hats a company can wear [cited 2 March 2012]. Available from http://purplebox.ghostery.com/?p=948639073; INTERNET.

————. 2011. Ghostrank planetary system [cited 2 March 2012]. Available from http://purplebox.ghostery.com/?*p*=1016021670; INTERNET.

————. 2012a. About Ghostery [cited 2 March 2012]. Available from http://www.ghostery.com/about; INTERNET.

————. 2012b. About chartbeat [cited 2 March 2012]. Available from http://www.ghostery.com/apps/chartbeat; INTERNET.

Goodale, G. 2012. Jedi knights of online privacy strike back at data-mining empires. *The Christian Science Monitor* [cited 20 March 2012]. Available from http://www.csmonitor.com/Innovation/2012/0314/Jedi-knights-of-online-privacy-strike-back-at-data-mining-empires; INTERNET.

Google. 2012. Privacy policy [cited 2 March 2012]. Available from http://www.google.com/policies/privacy/; INTERNET.

Harraway, D. 2003. *The companion species manifesto: Dogs, people, and significant otherness*. Chicago, IL: Prickly Paradigm Press.

Hayles, N. K. 2004. Print is flat, code is deep: The importance of media-specific analysis. *Poetics Today* 25 (1): 67–90.

Kennish, B. 2011. Tracking the trackers: How our browsing history is leaking into the cloud. *Youtube* [cited 26 March 2012]. Available from http://www.youtube.com/watch?v=BK_E3Bjpe0E; INTERNET.

Kitchin, R. 2011. The programmable city. *Environment and Planning B: Planning and Design* 38 (6): 945–51.

Latour, B. 2005. *Reassembling the social: An introduction to actor-network-theory*. Oxford: Oxford University Press.

Love, D. 2012. Here's the information facebook gathers on you as you browse the web. *Business Insider* [cited 2 March 2012]. Available from http://www.businessinsider.com/facebook-tracking-2011–11; INTERNET.

Madrigal, A. 2012. I'm being followed: How google—and 104 other companies—are tracking me on the web. *The Atlantic* [cited 2 March 2012]. Available from http://m.theatlantic.com/technology/archive/2012/02/im-being-followed-how-google-and-104-other-companies-are-tracking-me-on-the-web/253758/; INTERNET.

Milian, M. (2011). Making it harder for ads to track you online. *CNN* [cited 4 March 2012]. Available from http://edition.cnn.com/2011/TECH/web/06/21/ad.tracking/; INTERNET.

Mitcham, C. 1998. The importance of philosophy to engineering. *Teorema*, XVII/3: 27–47.

Mittal, S. 2010. User privacy and the evolution of third-party tracking mechanisms on the world wide web. Thesis [cited 4 March 2012]. Available from http://www.stanford.edu/~sonalm/Mittal_Thesis.pdf; INTERNET.

Pathak, A., Y. C. Hu, and M. Zhang. 2012. Where is the energy spent inside my App? Fine grained energy accounting on smartphones with eprof [cited 20 March 2012]. Available from http://research.microsoft.com/en-us/people/mzh/eurosys-2012.pdf; INTERNET.

Pew. 2012. Search engine use 2012 [cited 9 March 2012]. Available from http://pewinternet.org/Reports/2012/Search-Engine-Use-2012/Summary-of-findings.aspx; INTERNET.

Pfister, D. S. 2011. Networked expertise in the era of many-to-many communication: On Wikipedia and invention. *Social Epistemology: A Journal of Knowledge, Culture and Policy* 25 (3): 217–31.

Rieland, R. 2012. So what do we do with all this data? *The Smithsonian* [cited 4 March 2012]. Available from http://blogs.smithsonianmag.com/ideas/2012/01/so-what-do-we-do-with-all-this-data/; INTERNET.

Roberts, S. 2004. Self-experimentation as a source of new ideas: Examples about sleep, mood, health, and weight. *Behavioral and Brain Sciences* 27: 227–62 [cited 21 March 2012]. Available from http://escholarship.org/uc/item/2xc2h866#page-1; INTERNET.

Sense 2012. Feel. Act. Make sense [cited 4 March 2012]. Available from http://open.sen.se/; INTERNET.

Servin, J. 2010. David Hockney's fresh flowers. *Vogue* [cited 28 March 2012]. Available from http://www.vogue.com/culture/article/david-hockneys-fresh-flowers/; INTERNET.

throwaway236236. 2012. IAmA a malware coder and botnet operator, AMA, *Reddit* [cited 30 March 2012]. Available from http://www.reddit.com/r/IAmA/comments/sq7cy/iama_a_malware_coder_and_botnet_operator_ama/; INTERNET.

Tsukayama, H. 2012. FTC releases final privacy report, says "Do Not Track" mechanism may be available by end of year. *Washington Post* [cited 28 March 2012]. Available from http://www.washingtonpost.com/business/technology/ftc-releases-final-privacy-report-says-do-not-track-mechanism-may-be-available-by-end-of-year/2012/03/26/gIQAzi23bS_story.html; INTERNET.

Tugend, A. 2012. Bad Habits? My future self will deal with that [cited 4 March 2012]. Available from http://www.nytimes.com/2012/02/25/business/another-theory-on-why-bad-habits-are-hard-to-break-shortcuts.html?_r=3&pagewanted=all; INTERNET.

W3C. 2012. Tracking protection working group [cited 14 March 2012]. Available from http://www.w3.org/2011/tracking-protection/; INTERNET.

Wolfram, S. 2012. The personal analytics of my life [cited 9 March 2012]. Available from http://blog.stephenwolfram.com/2012/03/the-personal-analytics-of-my-life/; INTERNET.

Yarrow, J. 2011. Chart of the day: Here's how much a unique visitor is worth. *Business Insider* [cited 2 March 2012]. Available from http://www.businessinsider.com/chart-of-the-day-revenue-per-unique-visitor-2011–1; INTERNET.

Zax, D. 2011. You can't keep your secrets from twitter, fast company [cited 28 March 2012]. Available from http://www.fastcompany.com/1769217/there-are-no-secrets-from-twitter; INTERNET.