

This is an Author Accepted Manuscript of a book chapter published by Palgrave Macmillan (available online 20 July 2018) in Maller C. and Strengers Y. (eds) (2019) *Social Practices and Dynamic Non-Humans: Nature, Materials and Technologies*. Available at: https://link.springer.com/chapter/10.1007/978-3-319-92189-1_9

How software matters: Connective tissue and self-driving cars

Janine Morley

Abstract

Drawing on the example of self-driving and connected cars, this chapter explores how the software that is being integrated into, and transforming, everyday objects, might be conceptualised within theories of practice. It argues that, although software is an especially dynamic and intangible 'material', it can still be accommodated within existing conceptualisations of materiality in practice theories. The automation that software enables can be positioned as part of practice complexes, even when it does not play a direct, constitutive role in any single practice. In addition, through performing varied work in connecting practices and enabling 'feedback' over time and space, software can be understood to form part of the connective tissue by which practice complexes hang together and change.

Introduction

In its own right, software has become a topic of social science research, not just as a black-boxed component of the devices, infrastructures and systems that it enables. By unpacking and peering into the processes by which software is made and by which it operates in everyday life, researchers are exploring the distinctive 'work' that software does in the world: they are asking how "software matters" (Kitchin and Dodge 2011, 3). To date, this has included research into algorithms and their roles in political, personal and work life (Gillespie, Boczkowski, and Foot 2014, Zarsky 2015, Pasquale 2015, Beer 2017), the production of software (Mackenzie 2006), its introduction and embedding into organisational settings (Pollock and Williams 2008), and its role in the production of urban space (Thrift and French 2002, Kitchin and Perng 2016). Many writers have also grappled with the nature of software itself, and the distinctive challenges it poses for thinking about technological agencies (Thrift 2004, Leonardi 2010, Berry 2011, Kitchin and Dodge 2011). For instance, software is seen to have dynamic, almost living, qualities that render the world "increasingly lively" (Thrift 2004, 461)

For theories of practice, software is a challenging 'material' to conceptualise and study. This is, firstly, because some forms of software 'enliven' and automate processes in complex ways that reduce, remove or otherwise de-centre the active and practical doings of people. Yet practice theories, such as those of Schatzki (2002, 2010) and Shove, Pantzar and Watson (2012) on which I focus in this chapter, emphasise direct bodily activity as the very *basis* of practices, and therefore of social life, stability and change. So how is software, and the automated work it does in the world, to figure within these approaches?

The second challenge for theories of practice arises in the unusually dynamic nature of software itself: many forms of software can change in significant ways as they are used. In particular, software can be responsive to aggregate, past interactions or live data from beyond the immediate environment, bringing this to bear within situated performances. It can be updated remotely,

globally and quickly in ways that significantly change what objects or systems do. Some forms of software also 'learn' by themselves, as code is re-written through machine learning, without immediate oversight from people. In many applications of practice theory, particularly within studies of consumption, there is a tendency to define 'materiality' as comprised of discrete, relatively static and bounded objects: often limited to the "things handled in a social practice" (Reckwitz 2002, 221). Yet software is not well-bounded, nor is it tangible or fixed in the way that objects like cups, bicycles or buildings are.

In addressing these two related challenges, this chapter argues that Schatzki's (2002, 2010) and Shove, Pantzar and Watson's (2012) versions of practice theory are not necessarily at odds with the idea of a dynamic materiality that is 'active' beyond moments of direct human practical activity. Indeed, such material doings, happenings or processes often provide necessary conditions for performances. Moreover, the apparent degree of autonomous 'performance', or the work that software does, appears less independent from practices once it is considered as part of wider complexes (Shove, Pantzar, and Watson 2012) or systems of practice (Watson 2012). If the aim is a fuller consideration of the work that software does, it is therefore necessary to explore how software is positioned in relation to a range of practices within such complexes. This shifts the focus beyond the single practices which appear to be 'taken over' and 'carried on' by software technologies, and thereby the question of whether such technologies are performing these practices.

To develop an alternative, and relational, response to the automating and dynamically adaptive qualities of software, the chapter explores how software acts as a kind of connective tissue between practices: implicated in how practice complexes reproduce and extend over time. In this light, software appears to play a distinctive role compared to more fixed, object-like technologies, raising new lines of enquiry into how software matters.

Overall, the chapter responds to two key themes of the present volume: the question of how to account for 'non-human performance' within theories of practice and the relations that dynamic 'materials' facilitate between practices. The first section of the chapter delineates different types of software and identifies a handful of qualities that are of particular interest when internet-connected software is integrated into familiar everyday objects. These include a) the data-based extension and re-making of networks which link domestic and private practices to those of service providers, regulators and other organisations; b) the responsive adaptivity of software; and c) the fluid and rapid ways in which it circulates. This is illustrated through the case of self-driving and connected vehicles. The chapter then moves to the question of whether, and how, such qualities can be accommodated within theories of practice. After arguing that this is possible by attending to the relations between software and practices, the third section explores one approach to this: how software might be conceptualised as part of the 'connective tissue' through which practices hang together and extend over time. Finally, the chapter concludes with reflections on how this may help to orient empirical investigations of how software comes to matter, including understandings of large-scale, systemic change.

Soft dynamics and self-driving cars

Software infuses everyday life in diverse ways and through many different kinds of devices, infrastructures and assemblages that "mediate, supplement, augment, monitor, regulate, and

ultimately produce collective life” (Kitchin and Dodge 2011, 9). To help situate and differentiate these varied roles, Dodge and Kitchin (2009) propose a taxonomy of ‘coded’ objects in which software is embedded. This includes: objects such as cookers that do not depend on code for their operation but are ‘peripherally coded’ enabling supplementary functions like timers; and objects, referred to as ‘codejects’, that only function through code. This can be in more or less extensive and connected ways. For instance, some codejects, such as USB memory sticks, have firmware that is hard-wired and static; some codejects, like DVD players are programmable but in relatively fixed ways; and other codejects, like air conditioning units, detect and respond to their immediate environment. Further coded objects, referred to as ‘logjects’, monitor and record their own use. Some of these, like MP3 players, are relatively stand-alone whilst others, such as smartphones, are more dependent on networked connections in order to function fully. This latter category are ‘networked logjects’, and it is through them, in particular, that software transgresses the boundaries of otherwise self-contained objects, connecting them and traces of their use to networks that extend, often instantaneously, over space.

With the arrival of the Internet of Things (IoT), code is increasingly folded into existing categories of everyday objects, like thermostats, TVs and cars (Macrorie, Kragh-Furbo, and Morley 2016); endowing them with a range of properties (and concerns) previously restricted to computer-like devices, such as smartphones and laptops, and the online services to which they allow access. So although software already matters in many ways, the extension of software into further realms of everyday life, and into previously well-bounded objects, raises new questions as to how practice theoretical approaches might conceptualise and investigate software. Quite unlike the object-like artefacts that applications of practice theory are more used to dealing with, these new kinds of software are associated with a range of ‘softer’, more fluid and responsive dynamics.

In this chapter, I illustrate these dynamics by focusing on a particular networked logject: the car. In itself, the integration of software into cars is nothing new: most cars are already codejects which depend on software to ‘drive’ functions like power-assisted breaking, diagnostic functions, and the navigation, entertainment and information ‘centres’. But as cars are increasingly designed to connect directly to mobile communications and Wi-Fi networks, this allows not only for Internet access to be provided for devices taken into the car, but also for diagnostic, navigation and entertainment systems to collect, send and receive data via the Internet.

In some cases, connected tracking devices are added to cars and other vehicles for insurance, safety, navigation and logistics purposes. For instance, in the European Union, new cars sold from 2018 onwards must be fitted with an emergency tracking system, called eCall, that will detect when an accident has occurred and automatically call the emergency services, sending data about the vehicle’s location, direction of travel and time of the crash (European Commission 2015). It is likely that this data will then be subject to a range of algorithmic operations, for instance, to predict the severity of injuries (Wallace 2017).

Car companies are also starting to collect data from the vehicles they sell; but exactly what data can be unclear. Tesla, one of the most prominent companies in the development of self-driving cars, releases selected data in the event of accidents to show that the car itself was not at fault (Thielman 2017). By implication, such data as is ordinarily captured and sent ‘back’ to Tesla provides a sufficiently rich picture of how each of ‘their’ cars is being driven and in what road conditions as to

diagnose blame. In Europe, where data protection laws are more stringent, Audi is also starting to collect data from cars in the company's own fleet, before voluntarily enrolling wider populations.

But why are car manufacturers like Tesla and Audi collecting data from connected cars? It seems it is essential to the development of 'intelligent' vehicles. For instance, Audi plans to develop 'personal assistants' that will make a range of suggestions, depending on traffic and weather conditions about when to leave for work, or to call home if you're likely to be late (Gitlin 2016). Machine learning is key to these aspirations: it will allow software in the vehicle to monitor and infer a pattern of use as a basis for generating personalised suggestions for drivers. Just as with 'learning thermostats' like the Nest, this 'tailored support' will also adapt to incoming data about local conditions. The collection of data from current driving performances helps develop the algorithms that could eventually perform such assistance roles.

More hotly anticipated and debated, however, is the development of autonomous or self-driving vehicles. Trials are currently underway in many countries, and connected cars and the data they are already collecting are key to these developments. Tesla's Model S and Model X cars already run 'Autopilot' software which allows for driver-supervised, hands-free driving on motorways. In early 2017, the hardware of the latest Model S was declared to be ready for fully autonomous driving; it is just the software that remains to catch-up, as "it is still learning how to drive" (Stilgoe 2017, 1). This is through a process involving deep, machine learning operating on vast datasets generated from many actual performances of driving available to Tesla through their connected fleet. From this data, software can learn the 'rules' and generate algorithms that reliably replicate the same results as real drivers: each driver thereby becomes an "expert trainer for how the autopilot should work" (Fehrenbacher 2015, para. 7, quoted by Stilgoe, 2017). This doesn't happen in an isolated car-by-car fashion: rather, according to Elon Musk, the head of the company, "[t]he whole Tesla fleet operates as a network. When one car learns something, they all learn it" (Fehrenbacher 2015, para. 7).

In other words, this is a two way 'feedback' process. Not only does analysis of vast aggregated data from past driving performances inform the ongoing development of driving algorithms, but those algorithms are then fed back into *all* the cars on an ongoing basis, changing how they function. This responsive adaptation in functionality on a mass scale is possible because software is updateable. Results can be dramatic. Autopilot software was first rolled-out in 2015 to tens of thousands of vehicles already in use in the United States, via car owners' Wi-Fi networks: turning them overnight into (partially) self-driving cars (McHugh 2015). When it is ready, the software for fully autonomous driving will be similarly distributed. This mass adaptation is not just a question of the updatability of software per se, but also the way in which updates can be circulated. Thanks to near-constant (or routine) broadband connectivity, much software is now designed to be updated automatically and frequently at a global scale (de Decker 2017).

Compared to other non-coded material artefacts, software brings a remarkable dynamism to networked logjects. It means that objects like cars, and how they work, can be re-designed almost on the fly, much like apps or operating systems. Whilst this has become a familiar quality for computing devices, it brings potential for an increasing rate of change in the 'physical' functionality of everyday objects, and how they are used. Software can circulate and change in ways that are quite unlike physical products that have to be designed, tested, manufactured, shipped, stored, purchased and delivered: in other words, thanks to software, the scale and speed by which networked logjects may change transcends that of more 'concrete', self-contained technologies.

Thus, when it comes to developing accounts of change over time, this makes software an exceptionally interesting yet challenging ‘material’ to work with. It suggests, on the one hand, that software-based innovations like self-driving cars could bring widespread and rapid change to the practices in which networked logjects are embedded: to driving as well as other practices associated with car travel (McDonald 2016, Lin 2016). On the other hand, many changes in software proceed without discernible impact for the practices in which they are embedded; for instance, security requires a process of near-constant change to maintain stability within associated practices.

In sum, this section has highlighted several of the qualities of software that underpin its status as a distinctively dynamic ‘material’. First, it highlighted the extended networks in which data and updates may flow to and from networked logjects like connected cars, serving as resources or inputs to practices of driving and a range of related practices, such as emergency services and car design and production. Second, it explained how software can adapt (itself) both locally and across whole fleets of vehicles in response to data about past performances of driving. Thirdly, it highlighted the significance of the rapid and global way in which software can circulate, and with it bring adapted forms of functionality to everyday objects, which may in turn enable rapid changes in practice. I contend that these properties represent modes of feedback that actively contribute to the reproduction of practice complexes in ways that would not be possible without software. Before developing this argument, however, I will briefly turn to the question, pertinent to the themes of this book, of whether and how practice theories can conceptualise the dynamic and apparently ‘performative’ qualities of software outlined above.

Accounting for the ‘dynamism’ and ‘agency’ of software

In what follows, I will argue that the adaptive and fluid qualities of software are not incompatible with the way that practice theorists (in particular Schatzki (2010)) conceptualise materiality. The idea that ‘materials’ *only* feature as direct, object or even tool-like elements of practice performances is not necessarily the case. Instead, I suggest that materiality plays important roles as part of complexes of practice, even when not directly embedded in any single practice. This implies that the work that software does in the world – or what it ‘performs’ – can be understood through its relationality to multiple practices. Argued in this way, the question of whether automated technologies, like cars, perform practices, like driving, does not logically arise. Indeed, in the conclusion, I will suggest that it is something of a distraction from developing relational understandings of the work that software does, that is, its apparent ‘agency’.

First, let us consider the status of software as ‘a material’. This term is drawn from Shove, Pantzar, and Watson (2012) and is more properly described as ‘a material element’. It refers to a category of component brought together in the performance of practices and necessary to them, alongside skill and meaning. This framework draws on Reckwitz (2002, 208) who notes that “‘artefacts’ or ‘things’... necessarily participate in social practices just as human beings do”. In emphasising a practical engagement with the material world over an exclusively interpretive one, Reckwitz’s account of materiality within practice theory highlights “things” or “objects” that are “handled” in practices (2002, 212, 210). Given this, and given a lingering predilection in western thinking to conceptualise materiality in terms of discrete, fixed objects (Ingold 2010), it is not surprising that many who have worked with practice theories empirically have also tended to limit consideration of materiality to objects or things that are used or operated upon as part of a practice.

Yet software stands in stark contrast to this image of a thing or object. Firstly, the ways that it can respond flexibly to predefined data and ‘learn’ patterns of activity, including how to recreate human-like driving performances, resembles the characteristics of living organisms more than it does of objects (Thrift and French 2002, Thrift 2004). Secondly, software does not exist physically in the way that an object does. It is not physically tangible, yet it does have a physical substrate: it is stored and operates on disks or circuits of some kind. But these physical properties neither define it, nor do they shape the contours of possible actions that can be performed with or by it: which is the case for physical objects. Rather, software “exists in the way that speech or music exist... software when merely written as lines of code loses its essential essence – it’s executability” (Kitchin and Dodge 2011, 24). In other words, the essence of software is as a process, a doing: it “does *something* to *something*... it performs functions and processing” (Berry 2008, para. 1).

However, whilst the adaptivity and process-like nature of software are at odds with the image of an object, neither of these characteristics is actually incompatible with understandings of materiality within practice theories. Indeed, Schatzki (2010) describes materiality as arrangements composed of physical, biological or natural “somethings” that include “objects, things, properties, *events*, or *processes*” (133, emphasis added). In this account, one way in which materiality forms part of society arises from the biological or physical flows that pass through arrangements and the practices with which they are interlinked (practice-arrangement nexuses). Thus, to Schatzki material arrangements are not as fixed or static as the name suggests: they “are in some sense crystallizations of matter-energy flows” (Schatzki 2010, 137). This resonates with Ingold’s (2007, 2010) account of materials as immersed in processes of formation and transformation, as opposed to states of matter (see also Shove, Walker, and Brown 2014). In other words, it is not incompatible with practice theories for a material phenomenon like software to be dynamic, changeable, and responsive to situations, or even to *do* something.

With the possibility for a dynamic materiality that goes beyond concepts of objects or artefacts established, the next step in arguing that software can largely be accommodated within the existing conceptual resources of practice theories is to note that material entities or processes need not be direct parts of the performance of practices in order to matter for, or relate to, them. This is to recognise a variety of roles for materiality beyond simple or direct, tool-like ‘elements’ of practice. Indeed, there are already a number of moves in this direction, for instance, in thinking through how infrastructures relate to practices (Shove, Watson, and Spurling 2015) or the different roles that the same material entities can play as relative to practices (Shove 2017). To Schatzki (2010, 2012) material arrangements and practices are inter-linked in many different ways: “practices effect, use, give meaning to, and are inseparable from arrangements while ... arrangements channel, prefigure, facilitate, and are essential to practices” (2012, 4) through relations of causality, prefiguration, intelligibility, intentionality and constitution. In this light, software does not need to figure as a direct constitutive part of practices in order to feature in the way that practices and arrangements hang together, and thus how social life transpires.

More specifically, I have argued elsewhere (Morley 2017) that automated machines may be conceptualised as part of processes, like manufacturing or heating, that are organised by, and cut across, many practices. Even if they do not perform a central device-like, resource-like or even infrastructural role (Shove 2017) in any single practice, automated machines can usually be understood to feature as a necessary component in a complex or system of practices. Like automated machines, software connects to multiple practices even those in which it is not directly

handled or 'used'. Thus, the automated work that software allows for can also be accounted for, and studied, as part of complexes of practice. In fact, there is broader movement away from the analysis of single practices to that of multiplicities or complexes of practice and how they change (Schatzki 2011, Shove, Pantzar, and Watson 2012, Blue and Spurling 2017, Blue 2017).

From this perspective, the work that software does, particularly in automating complex technologies, needs to be understood as a relational outcome with reference to a range of practices. The very ability of software to do work in the world, to "carry, fold, frame and redistribute actions in different environments" (Mackenzie 2006, 423) might be seen as an extension of human agency, perhaps even human bodies (Wallenborn 2013). Mackenzie (2006) refers to this as secondary agency: the capacity to act 'autonomously' yet *on behalf of* a range of other actors, such as the companies, groups and programmers that build and deploy code, and the people in myriad situations who use it.

The challenge of thinking about what software does, or its 'agency', is therefore one of thinking through relationships with and between practices. In echoing a number of other authors (Barad 2003, Ingold 2007), I would suggest that the apparent 'performative agency' of software, such as when a Tesla car takes over the control of driving along a motorway, is not an inherent attribute of the software, or the car, itself: it does not *have* agency that then allows it to act. Rather, much like human agency, this action is realised and made possible only through the wider practice-arrangement nexuses within which such cars are embedded. Whatever this work or action is, it has been established, is understood and has effect through practices, and the material arrangements to which they connect. In other words, if we want to better understand and conceptualise what self-driving cars, and the software that 'drives' them, actually *do*, it is paramount to pay close attention to how software relates to practices and how it mediates relationships between them.

Connective tissue: Software actively connects practices

This section briefly explores the idea that at least some of the work that software does in relation to practices can be understood as a kind of connective tissue within practice complexes. The concept of connective tissue is used by Shove, Pantzar, and Watson (2012, 36) as that which "holds complex social arrangements in place, and potentially pulls them apart." It is developed by Blue and Spurling (2017) in order to analyse how complexes of practice in hospitals change. In particular, they suggest that material-spatial qualities of practices, alongside jurisdictional and temporal qualities form interconnections within practice complexes.

Based on the case of connected and self-driving cars, outlined above, it is certainly possible to appreciate how software contributes to making, and re-making, relationships between practices in each of these ways. Firstly, by including new data flows within the practice of driving, such as live navigational advice, and by sending new flows of data to car makers and emergency services, connected car software extends the material-spatial co-ordinates of driving. As Dodge and Kitchin (2009, 1359) observe for networked logjects in the home:

"homes are being embedded in real-time, two-way networks so that the everyday nature of domestic practices... can be monitored by service providers with the attendant data being used to profile and social sort customers. The consequence of networked logjects is that homes are increasingly being stretched out across space in networks of greater and shorter length... and they are subject to increasing levels of (corporate) surveillance."

With the arrival of connected cars, the same will be increasingly true for driving and its relation to other practices such as those of emergency services, insurers and the car industry. Secondly, this could have jurisdictional implications. Commentators note that regulating software-based driving is a very different prospect to that of human driving (Stilgoe 2017). Thirdly, software that enables live sharing of data with third parties and which automates driving has the potential to re-make the temporal connections that hold driving and other practices together. At the simplest level, traffic control measures or emergency services may be able to respond to incoming data over different (faster) time scales whilst other activities such as working might become possible when commuting by car. In other words, the ability of software to 'rewire' relationships between practices is testament to the way that it 'performs' these connections (or may do in the future).

Shove, Pantzar, and Watson (2012, 17) propose that such connections within and between practices might also be understood as "circuits of reproduction through which one performance relates to the next and... through which practices shape each other". Between practices, this shaping is referred to as 'cross-referencing'. This denotes how multiplicities of practices (bundles and complexes) are held together, and might include temporal co-ordination, narrative-based organisation, or the ways that elements co-ordinate and link practices when they are shared, passed between practices or competed for. They also refer to forms of 'monitoring', recording and feedback that have constitutive roles in performances and feed forward from one performance to the next. If, and when, such forms of feedback cross-over with the circuits formed between practices, it is possible to envisage how circuits of reproduction may work across complexes as a whole. Understood like this, connective tissue is not just about how practices cohere, but also how complexes reproduce and change.

In the first section of this chapter, I suggested that the dynamic characteristics of software (that situate technologies within extended data-based networks, that adapt to past performances, and that can be circulated near-instantaneously and globally) might be understood as forms of feedback within complexes of practice. In general, information technologies may already be implicated in 'rewiring' the circuits of reproduction. In commenting on the concept of global microstructures developed by Knorr Cetina (2005), Shove, Pantzar, and Watson (2012, 117) suggest that:

"the emergence of such modes may be symptomatic of a more pervasive transformation in the circuitry of society itself. Whether or not this is so, it is clear that technologies and instruments of feedback, broadly defined, are of direct relevance in configuring circuits of reproduction that are, in turn, of equally direct consequence for the survival and transformation of relations between practices and the elements of which they are composed."

Through its roles in communication technologies software already matters in this way. The extending role of software in everyday networked logjects prospectively takes this even further.

In the case of connected and self-driving cars, software enables new possibilities for 'feedback' in the relations between practices described above. These software-enabled circuits of feedback go beyond the relationships between car 'users' and those who make, insure, regulate, maintain them or provide emergency assistance. They include circuits of feedback within practices, and practice complexes, from past to present or future performances. This can be seen in the local machine learning that Audi's personal assistant software will depend upon, and in the aggregated, mass 'learning' by which Tesla's Autopilot software is evolving. Analytics performed on past performances are combined with analysis of incoming and live data, either to produce contextually specific 'advice' or to affect control of a car. These circuits extend both across time, connecting past to present

performances, and space, bringing to bear traces of data about the many other performances of driving that generate traffic conditions. In other words, software does indeed appear to simultaneously 'perform' multiple circuits of reproduction across 'automobility' complexes or systems (Urry 2004).

Taking this idea further, we might suggest that it is the very re-making of connective tissues of the complexes surrounding driving, through which automated driving is becoming an increasingly real prospect. On the one hand, Stilgoe (2017) argues that the emphasis on machine learning makes self-driving cars a much closer prospect by removing some of the previously assumed need to overhaul infrastructures to accommodate these new kinds of vehicles. The development of autonomous vehicles through machine-learning based on data of actual driving performances reduces the field of necessary change to one of technological substitution (or delegation) between human and non-human driver. Yet, on the other hand, this apparent substitution, limited to the task of driving itself, can still be seen as a process of transformation in the connective tissue of automobility complexes. This is because it depends on making new relations of feedback between past performances of driving and the development of software, and in turn this depends upon new and different relations between drivers and car-makers as observed in the case of Tesla and their access to vast sets of data on what drivers do.

In summary, when it comes to accounting for the software embedded in networked logjects, this section has argued that it is helpful to trace the modes of 'feedback' – between practices and between performances – that forge connections across complexes of practice. Indeed, I have suggested that this could be a useful approach for analysing the very development of self-driving technologies. These modes of feedback are an important aspect of the work that software does in the world, and it is worth stressing that this is not passively achieved, but actively 'performed' through the flows of data, software updates, analytics and algorithmic processing that make software such a dynamic phenomenon.

Conclusion

In drawing on self-driving and connected cars as an example of how software can matter in new ways, my aim has been two-fold. First, I have hoped to set-out and respond to some of the qualities of software that make it such an intriguing 'material', particularly for theories of practice that have hitherto been relatively limited in their conceptual and empirical treatment of material elements or arrangements as primarily consisting of a set of discrete, well-bounded, tangible and static objects. I have argued that this understanding of materiality is not inherent or necessary to practice theories. Rather, alternative and more promising directions are already present and emerging: in Schatzki's (2010) description of material arrangements including processes and events, in Shove, Watson, and Spurling's (2015) discussion of how infrastructures relate to practices, in Shove's (2017) argument that materials occupy roles relative to the practices in question, and in Blue and Spurling's (2017) account of materiality as an aspect of the connective tissue by which practice complexes hang together. In adding to these developments, I argue that the work that software does in the world can be defined in relation to complexes (or systems) of practice.

Therefore, if we have an interest in understanding the distinctively dynamic qualities of software, and the work this makes possible - the 'performative' qualities of software - then the challenge is to understand the multiple relationships within practice complexes through which this work is done. To

reduce this to the question of whether vehicles or non-humans can perform social practices without humans, risks prematurely cutting out of view the richness of relationships, material and otherwise, in which driving is embedded. In itself, this is a fascinating topic to explore, as in other chapters in this volume (see Strengers, Kuijer). But, I would suggest that it is not the first question to ask of increasingly sophisticated and 'smart' technological developments like the self-driving car. Indeed, to be occupied only with the practice of driving and the re-distribution of performative work within the driver-car assemblage (Dant 2004) is to miss the work that software is doing to make this possible. This can only be understood through transformations in relations across wider systems of practice in which driving is embedded. To this end, it is helpful to conceptualise software as part of the connective tissue through which practices hang together. More specifically, software is plausibly involved in circuits of reproduction at the level of complexes.

This remains a limited and speculative account of how software matters; it is also likely to matter through many other contributions to regulating, sustaining and reproducing practices. But I hope it indicates a direction of travel. Indeed, there is much more to do in thinking through the conceptualisation of connective tissue, and in developing empirically-founded analyses of how technologies, and not only software, are implicated in it. This marks a move away from a dominant focus on the direct roles that non-human materiality plays within single practices, towards consideration of indirect roles in stitching together practice complexes and the ways in which they reproduce, extend and change over time. This could help in developing practice theoretical accounts of 'large' social phenomena (Schatzki 2011) and systemic societal change.

I have also noted the parallel that other authors have drawn between software and organic life. Even though software is a technology, and is crafted to carry on human intentionality and thus differs markedly from organic life or geo-physical processes, the approach to exploring indirect relationships in complexes or looser systems of practice, as sketched out in this chapter, may be helpful for analysing a range of other 'non-humans' and their contributions to social life, such as the weather, light, health or heat (see Opperman and Walker, this volume). Yet I also hope it is clear that the study of software in cases like the self-driving car, may serve as fruitful ground on which to further develop and substantiate the concept of connective tissue, through empirical investigation of circuits of feedback and reproduction.

References

- Barad, Karen. 2003. "Posthumanist performativity: Toward an understanding of how matter comes to matter." *Signs: Journal of women in culture and society* no. 28 (3):801-831.
- Beer, David. 2017. "The social power of algorithms." *Information, Communication & Society* no. 20 (1):1-13.
- Berry, David. 2008. "A Contribution Towards A Grammar of Code." *Fibreculture Journal* no. 1 (13):1-1.
- Berry, David. 2011. *The philosophy of software: Code and mediation in the digital age*. Basingstoke: Palgrave Macmillan.
- Blue, Stanley. 2017. "Institutional rhythms: Combining practice theory and rhythm analysis to conceptualise processes of institutionalisation." *Time & Society*.
- Blue, Stanley, and Nicola Spurling. 2017. "Qualities of connective tissue in hospital life: How complexes of practice change." In *The Nexus of Practices: Connections, Constellations, Practitioners*, edited by Allison Hui, Theodore Schatzki and Elizabeth Shove, 24-37. Abingdon, Oxon: Routledge.

- Dant, Tim. 2004. "The driver-car." *Theory, Culture & Society* no. 21 (4-5):61-79.
- de Decker, Kris. 2017. *Rebooting energy demand: Understanding software updates* [Accessed 17.08.17]. Available from <http://www.demand.ac.uk/rebooting-energy-demand-automatic-software-updates/>.
- Dodge, Martin, and Rob Kitchin. 2009. "Software, objects, and home space." *Environment and Planning A* no. 41 (6):1344-1365.
- European Commission. 2015. *eCall in all new cars from April 2018* [Accessed 09.09.2017]. Available from <https://ec.europa.eu/digital-single-market/news/ecall-all-new-cars-april-2018>.
- Fehrenbacher, Katie. 2015. *How Tesla is ushering in the age of the learning car*. Fortune, 16 October 2015 [Accessed 04.09.2017]. Available from <http://fortune.com/2015/10/16/how-tesla-autopilot-learns/>.
- Gillespie, Tarleton, Pablo Boczkowski, and Kirsten Foot. 2014. "The relevance of algorithms." In *Media technologies: Essays on communication, materiality, and society*. MIT Press.
- Gitlin, Johnathan M. 2016. *Audi wants its connected cars to improve the breed* [Accessed 09.08.2017]. Available from <https://arstechnica.co.uk/cars/2016/09/personal-assistants-and-data-analytics-the-future-of-audis-car-ux/>.
- Ingold, Tim. 2007. "Materials against materiality." *Archaeological dialogues* no. 14 (01):1-16.
- Ingold, Tim. 2010. Bringing things back to life: Creative entanglements in a world of materials NCRM Working Paper. Realities / Morgan Centre, University of Manchester, Available from <http://www.manchester.ac.uk/realities/publications/workingpapers/>.
- Kitchin, Rob, and Martin Dodge. 2011. *Code/space: Software and everyday life*: MIT Press.
- Kitchin, Rob, and Sung-Yueh Perng. 2016. *Code and the City*: Routledge.
- Knorr Cetina, Karin. 2005. "Complex Global Microstructures The New Terrorist Societies." *Theory, Culture & Society* no. 22 (5):213-234.
- Leonardi, Paul M. 2010. "Digital materiality? How artifacts without matter, matter." *First Monday*.
- Lin, Patrick. 2016. "Why Ethics Matters for Autonomous Cars." In *Autonomous Driving*, edited by M Maurer, J Gerdes, B Lenz and H Winner. Berlin: Springer.
- Mackenzie, Adrian. 2006. *Cutting code: Software and sociality*. Vol. 30: Peter Lang.
- Macrorie, Rachel, Mette Kragh-Furbo, and Janine Morley. 2016. Cracking the Code: How algorithms and software are shaping everyday life. In *Daily life, digital technologies and energy demand. Working Paper Collection*, edited by Rachel Macrorie and Faye Wade: Balance Network and Urban Institute.
- McDonald, Rob. 2016. *Why driverless cars may make cities sprawl even more* [Accessed 04.09.2017]. Available from <https://blog.nature.org/science/2016/04/20/why-driverless-cars-cities-sprawl-urban-suburban-development/>.
- McHugh, Molly. 2015. *Tesla's cars now drive themselves, kinda*. Wired [Accessed 20.04.2017]. Available from <https://www.wired.com/2015/10/tesla-self-driving-over-air-update-live/>.
- Morley, Janine. 2017. "Technologies within and beyond practices." In *The Nexus of Practices: Connections, constellations, practitioners*, edited by Allison Hui, Theodore Schatzki and Elizabeth Shove, 81-97. Abingdon, Oxon: Routledge.
- Pasquale, Frank. 2015. *The black box society: The secret algorithms that control money and information*: Harvard University Press.
- Pollock, Neil, and Robin Williams. 2008. *Software and organisations: The biography of the enterprise-wide system or how SAP conquered the world*. Abingdon, Oxon: Routledge.
- Reckwitz, Andreas. 2002. "The Status of the "Material" in Theories of Culture: From "Social Structure" to "Artefacts"." *Journal for the Theory of Social Behaviour*:195-217.
- Schatzki, Theodore R. 2011. Where the action is (on large social phenomena such as sociotechnical regimes) Sustainable Practices Research Group, Working Paper 1, Available from <http://www.sprg.ac.uk/uploads/schatzki-wp1.pdf>.
- Schatzki, Theodore R. 2002. *The Site of the Social: A Philosophical Account of the Constitution of Social Life and Change*. University Park, Pa: The Pennsylvania State University Press.

- Schatzki, Theodore R. 2010. "Materiality and Social Life." *Nature and Culture* no. 5:123-149.
- Schatzki, Theodore R. 2012. A Primer on Practices. In *Practice-Based Education: Perspectives and Strategies*, edited by Joy Higgs, Ronald Barnett, Stephen Billett, Maggie Hutchings and Franziska Trede. Rotterdam, The Netherlands: Sense Publishers.
- Shove, E., G. Walker, and S. Brown. 2014. "Material culture, room temperature and the social organisation of thermal energy." *Journal of Material Culture* no. 19 (2):113-124.
- Shove, Elizabeth. 2017. "Matters of practice." In *The Nexus of Practices: Connections, Constellations, Practitioners*, edited by Allison Hui, Theodore Schatzki and Elizabeth Shove, 155-168. Abingdon, Oxon: Routledge.
- Shove, Elizabeth, Mika Pantzar, and Matt Watson. 2012. *The dynamics of social practice: Everyday life and how it changes*. London: SAGE Publications Ltd.
- Shove, Elizabeth, Matt Watson, and Nicola Spurling. 2015. "Conceptualizing connections: Energy demand, infrastructures and social practices." *European Journal of Social Theory* no. 18 (3):274-287.
- Stilgoe, Jack. 2017. Machine Learning, Social Learning and the Governance of Self-Driving Cars SSRN Research Paper, Available from <https://ssrn.com/abstract=2937316>.
- Thielman, Sam. 2017. *The customer is always wrong: Tesla lets out self-driving car data – when it suits* [Accessed 04.09.2017]. Available from <https://www.theguardian.com/technology/2017/apr/03/the-customer-is-always-wrong-tesla-lets-out-self-driving-car-data-when-it-suits>.
- Thrift, Nigel. 2004. "Electric animals: New models of everyday life?" *Cultural Studies* no. 18 (2-3):461-482.
- Thrift, Nigel, and Shaun French. 2002. "The automatic production of space." *Transactions of the Institute of British Geographers* no. 27 (3):309-335.
- Urry, John. 2004. "The 'System' of Automobility." *Theory, Culture & Society* no. 21 (4-5):25-39.
- Wallace, Nick. 2017. *EU's eCall regulation sacrifices safety for privacy* [Accessed 09.08.2017]. Available from <https://www.euractiv.com/section/digital/opinion/eus-ecall-regulation-sacrifices-safety-for-privacy/>.
- Wallenborn, Grégoire. 2013. "Extended bodies and the geometry of practices." In *Sustainable practices: social theory and climate change*, edited by Elizabeth Shove and Nicola Spurling, 146-164. London: Routledge.
- Watson, Matt. 2012. "How theories of practice can inform transition to a decarbonised transport system." *Journal of Transport Geography* no. 24:488-496.
- Zarsky, Tal. 2015. "The Trouble with Algorithmic Decisions: An Analytic Road Map to Examine Efficiency and Fairness in Automated and Opaque Decision Making." *Science, Technology & Human Values* no. 41 (1):118-132.