From a single line of code to an entire city: reframing thinking on code and the city

Rob Kitchin, National Institute for Regional and Spatial Analysis, National University of Ireland Maynooth, County Kildare, Ireland.

The Programmable City Working Paper 4

http://www.nuim.ie/progcity/

7th November 2014

Presented at the *Code and the City* workshop, 3rd-4th September 2014, Programmable City, NIRSA, National University of Ireland Maynooth

Abstract

Cities are rapidly becoming composed of digitally-mediated components and infrastructures, their systems augmented and mediated by software, with widespread consequences for how they are managed, governed and experienced. This transformation has been accompanied by critical scholarship that has sought to understand the relationship between code and the city. Whilst this work has produced many useful insights, in this paper I argue that it also has a number of shortcomings. Principal amongst these is that the literatures concerning code and the city have remained quite divided. Studies that focus on code are often narrow in remit, fading out the city, and tend to fetishize and potentially decontextualises code at the expense of the wider socio-technical assemblage within which it is embedded. Studies that focus on the city tend to examine the effects of code, but rarely unpack the constitution and mechanics of the code producing those effects. To provide a more holistic account of the relationship between code and the city I forward two interlinked conceptual frameworks. The first places code within a wider socio-technical assemblage. The second conceives the city as being composed of millions of such assemblages. In so doing, the latter seeks to provide a means of productively building a conceptual and empirical understanding of programmable urbanism that scales from individual lines of code to the complexity of an entire urban system.

Key words: code, city, programmable urbanism, software studies, smart city, urban studies

I Introduction

> 'The modern city exists as a haze of software instructions. Nearly every urban practice is becoming mediated by code' (Amin and Thrift 2002: 125)

Over the past few decades software has become essential to the functioning of cities. It is deeply and pervasively embedded into the systems and infrastructure of the built environment and in the management and governance of urban societies. Digital technologies and services augment and facilitate how we understand and plan cities, how we manage urban services and utilities, and how we live urban lives. Software is used to produce, mediate, augment, and regulate systems and tasks. In so doing, networked digital technologies are helping to produce what has been termed 'smart cities': densely instrumented urban systems that can be monitored, managed and regulated in real-time (see Townsend 2013; Kitchin 2014) and whose data can be used to better depict, model and predict urban processes and simulate future urban development (Batty *et al*., 2012).

Thousands of papers and reports document the development of new digital technologies and their potential impact on cities and citizens or have examined the role software plays in managing urban infrastructures and practices. The vast majority of studies, however, focus on the development of new innovations and the production, deployment and effects of software from a non-critical, technological, engineering and governance perspective. A relatively small proportion take a more critical perspective, detailing how certain digital technologies produce new socio-spatial practices and effects (such as spatial sorting, algorithmic regulation, anticipatory governance, and control creep) and forms of networked urbanism and their wider social, political and economic consequences to urban life (e.g., Mitchell 1995; Graham and Marvin 2001; Graham 2005; Foth 2008; Shepard 2011). Only in a handful of cases, however, has critical and conceptual attention been focused on the nature of software itself, its underlying code, and its relationship to urban management, governance and practices (e.g., Thrift and French 2002; Kitchin and Dodge 2011; Kelley 2014).

Drawing inspiration from software studies -- a new field that takes software, and its production and deployment, as its object of critical analysis (see Fuller 2008; Berry 2011; Manovich 2013) -- these critical interventions consider the ways in which cities and citizens

are translated into code and how this code is then used to reshape cities and mediate the lives of their inhabitants. The principle argument forwarded it is that:

1. code is an actant that possesses 'secondary agency' (Mackenzie 2006), that is, it is ceded the power to process data and to make automated, automatic and autonomous decision-making and action, thus making aspects of the city sentient (Dodge and Kitchin 2007; Shepard 2011);
2. code transduces space, that is, it alters the unfolding production of space through its deployment (Dodge and Kitchin 2005);
3. the city becomes programmable, that is, open to recoding and remediation, but also to being buggy and hackable (Kitchin 2011; Townsend 2013).

Code, it is thus argued, through its work as an actant produces forms of coded space, wherein code augments or mediates the production of space but is not essential to its production, and code/space, wherein code is essential to a space being produced as intended. Much of the city is now produced as code/space, wherein if the code fails the space is not transduced as desired (e.g., if checkout software crashes then a space is transduced as a warehouse not a supermarket, or if check-in software crashes then the space is transduced as a large waiting room -- in both cases there is no longer any manual way to process transactions; code and space are mutually constituted). Moreover, code and forms of automated management are actively and extensively employed in the management and governance of urban systems, especially with respect to critical infrastructure and utilities (e.g., transport, energy, water) and policing, security and surveillance.

Despite the rapid development and deployment of digital technologies for augmenting city management and urban life, and the creation and rollout of new forms of networked urbanism, it is fair to say that critical analyses of the relationship between code and cities is small in number, underdeveloped conceptually, and lacking detailed empirical case material (the same can be said for software studies more generally). The speed of technological innovation and material deployment, and the power of the discursive regimes driving their adoption, is outpacing and outflanking critical reflection and intervention. Moreover, critical social scientists and humanities scholars are still struggling to get to grips conceptually with a series of interrelated phenomena -- code, ubiquitous computing, big data, networked urbanism, and smart cities -- at the same time as trying to map out and dissect their consequences and implications. My book with Martin Dodge, *Code/Space: Software and*

Everyday Life (2011), was an attempt to provide such an overarching, holistic conceptual framework and to make sense of the changes digital technologies were making to the urban condition. As with all such texts it was provisional -- a staging post rather than definitive guide.

In this paper I want to revisit some of the conceptual ideas we developed and to rework and extend them, focusing particularly on deepening and widening our conceptualisation of code and software. The rest of the paper is divided into two sections. The first focuses on code itself and the importance of delving into the nuts and bolts and mechanics of its constitution and operation, whilst at the same time not overly fetishizing code at the expense of the wider socio-technical assemblage within which it is embedded. The second focuses on how these socio-technical assemblages are framed within the wider discursive and material technological terrain and urban landscape, and interact and scale to produce densely instrumented cities consisting of millions of coded objects/systems all in dynamic flux. In this sense, the two sections are trying to find a way of dealing with the issue of productively building a conceptual understanding that scales from individual lines of code to the complexity of an entire urban system; of building a conceptual edifice that moves beyond marrying software studies to urban studies. This is no easy challenge, and I would see the arguments I make as another provisional step that others will hopefully help develop and make more robust.

II Thinking about *code* and the city

In *Code/Space* we argued that software needed to be understood as being both a product of the world and a producer of the world. Code -- the lines of declarations, procedures, commands and algorithms, expressed in different languages (assembly, scripting, procedural, etc) -- that when compiled create software are not simply the result of a neutral, technical exercise. Rather coding needs to be understood as a complex and contingent process, shaped by the abilities and worldviews of programmers and engineers, working in companies or on their own time, situated in social, political and economic contexts (Rosenburg 2007). Software development occurs in a collaborative framework, with individuals performing as part of a team or re-appropriating code from libraries or ideas from websites, books and magazines. Often several teams will work on different aspects of the same programme which are then stitched together. Teams can have different visions about what they are trying to achieve, and have different skill levels to tackle the job at hand. Software then is not an immaterial, stable, value-free product, it is a complex, multifaceted, mutable set of relations

created through diverse sets of discursive, economic and material practices rooted in particular locales. Moreover, this software does not simply represent the world, but actively participates in it, transducing space, reshaping work, transforming practices, and so on (Dourish 2001).

We argued for a need to, on the one hand, delve further into the nature of code itself, and in particular to start to unpick how coding is actively practised and code created in context, and on the other to examine the work that code does in the world. Here, I want to focus on the former. In trying to make sense of code and coding with respect to urban systems we advocated: (1) a focus on the code itself, deconstructing the lines of code and examining the ways in which elements of the world, and ways to think about and process them, are captured and formalised in sets of interlinked algorithms, and excavating how the code and algorithms evolve through revisions and editions as they incorporate new ideas, ambitions, policy and law; (2) ethnographies of coders and coding projects, including their wider social, political and economic framing. In other words, we posited a very software studies approach to making sense of code and cities.

I am still of the view that an in-depth focus on code and coding would be an enormously profitable endeavour. Given the huge growth in forms of algorithmic governance -- everything from recommendation systems, to automated forms of surveillance, to profiling and sorting -- it is becoming increasingly important to understand the aetiology of code (how algorithms are constructed and operate), how they are utilised, and to tease apart their inherent politics (see Gillespie 2014; Kitchin 2014b). This is evident in two recent, excellent software studies texts: Nick Montfort et al's (2012) *10 Print*, a detailed analysis of a single, but iconic, line of code; and Lev Manovich's (2013) *Software Takes Command*, in which he provides an in-depth genealogy of the 'softwarization' of cultural media -- art, photos, film, television, music -- that has taken place since the 1970s. That said, I have a major concern with this approach in and of itself: it adopts an analytical lens that over-fetishizes and potential decontextualises code at the expense of its wider assemblage of production and use.

Since the publication of *Code/Space* I have written another monograph -- *The Data Revolution* (2014c) -- which I loosely thought of as the third book in a trilogy of sorts (*Mapping Cyberspace*: infrastructure; *Code/Space*: software; *The Data Revolution*: data) and started a large, five year European Research Council funded project, *The Programmable City*, than involves ten subprojects focused on the intersections of ubiquitous computing, software, big data and the creation of smart cities. Both projects have highlighted that the relationship

between code and the city is complex and diverse. Code/software are critical to networked urbanism, but so too are data, platforms, hardware, interfaces, and users. And none of these can be fully understood without being considered in relation to one another, nor outside of their wider context. This has been bought home to me in two ways, which when combined provide a path forward.

First, in *The Data Revolution* I develop the argument that to fully comprehend an open data system, or a big data product, or a research data infrastructure, one needs to examine its entire data assemblage (see Table 1). The apparatuses and elements detailed in Table 1 interact with and shape each other through a contingent and complex web of multifaceted relations. And just as data are a product of the assemblage, the assemblage is structured and managed to produce those data (Ribes and Jackson 2013). Data and their assemblage are thus mutually constituted, bound together in a set of contingent, relational and contextual discursive and material practices and relations. This argument can be equally extended to code/software (indeed, this is an extension of a discussion first expressed in *Code/Space* and also at the start of this section). For example, an app like Foursquare or a city GIS system consist of a large amalgam of apparatuses and elements that shape how they are conceived, developed, administered, operated, and interactions with them deployed. A GIS is underpinned by a realist system of thought; it pulls together and combines hundreds of analytic and visualisation algorithms and dozens of datasets and has to be able to handle lots of different data formats, standards, and protocols; it has a diverse set of accompanying forms of supporting documentation, trade and academic journals; the system and its data are maintained, updated and used by many collaborating stakeholders, through a diverse set of practices, undertaken by many workers, using a range of materials and infrastructures; its operational costs are a source of contention; its use is shaped by legal frameworks and regulations; it is one part of a multi-billion dollar industry and community of practice; and so on. And GISs continue to evolve and mutate as "new ideas and knowledges emerge, technologies are invented, organisations change, business models are created, the political economy alters, regulations and laws are introduced and repealed, skill sets develop, debates take place, and markets grow or shrink" (Kitchin and Lauriault 2014). They are thus always in a state of becoming. One cannot fully grasp the constitution, operation and work of a GIS by concentrating attention on its code, despite the fact that without code a GIS could not exist. It has to be framed as a socio-technical assemblage.

Table 1: The apparatus and elements of a data assemblage

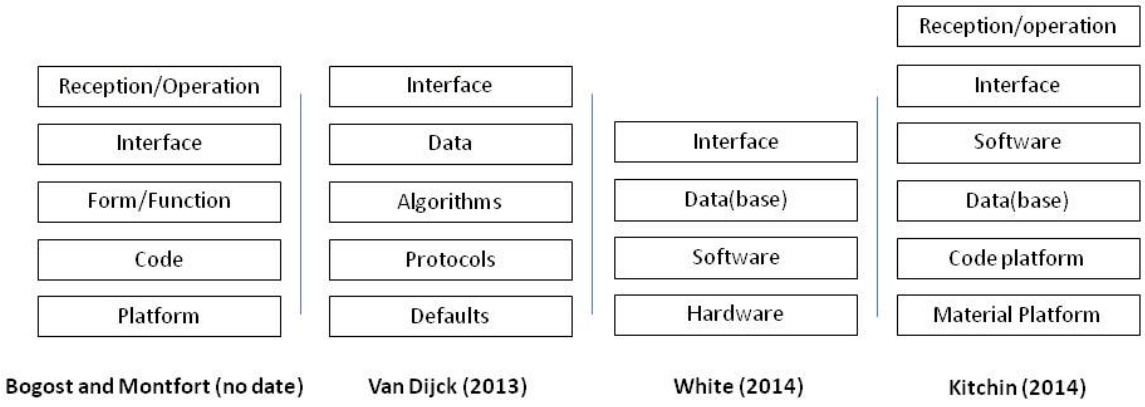
Apparatus	Elements
Systems of thought	Modes of thinking, philosophies, theories, models, ideologies, rationalities, etc.
Forms of knowledge	Research texts, manuals, magazines, websites, experience, word of mouth, chat forums, etc.
Finance	Business models, investment, venture capital, grants, philanthropy, profit, etc.
Political economy	Policy, tax regimes, incentive instruments, public and political opinion, etc.
Governmentalities and legalities	Data standards, file formats, system requirements, protocols, regulations, laws, licensing, intellectual property regimes, ethical considerations, etc.
Materialities and infrastructures	Paper/pens, computers, digital devices, sensors, scanners, databases, networks, servers, buildings, etc.
Practices	Techniques, ways of doing, learned behaviours, scientific conventions, etc.
Organisations and institutions	Archives, corporations, consultants, manufacturers, retailers, government agencies, universities, conferences, clubs and societies, committees and boards, communities of practice, etc.
Subjectivities and communities	Of data producers, experts, curators, managers, analysts, scientists, politicians, users, citizens, etc.
Places	Labs, offices, field sites, data centres, server farms, business parks, etc, and their agglomerations
Marketplace	For data, its derivatives (e.g., text, tables, graphs, maps), analysts, analytic software, interpretations, etc.

Source: Kitchin (2014a: 25)

Second, I have been trying to assemble my thoughts with respect to making conceptual sense of algorithms (Kitchin 2014b) and interfaces (Kitchin et al., 2014) that draws on related, but distinctly labelled literatures (e.g., critical code studies, HCI, new media studies), thus adding to my existing ideas with respect to infrastructure, code and data. This has led to a consideration, drawing on the discussion and conceptual diagrams of Montfort et al. (2012), Bogost and Montfort (no date), Van Dijik (2013, detailed in White 2014) and White (2014) (see Figure 1), of the make-up of the digital technology stack (the elements that work *together*) underpinning particular digital innovations/products/services that are deployed in cities. In my version of the stack there are six elements: material platform (infrastructure - hardware), code platform (operating system), data(base), code/algorithms (software), interface, and reception/operation (user/usage). Each layer has effects with regards to the others. For example, the hardware influences the choice of operating system, which shapes the choice of programming environment; the form and extent of the data influences how algorithms are constructed, as do user expectations and patterns of use; the interface is constrained by the hardware and shapes user experience of a technology, and so on (Montfort et al. 2012 has a nice discussion about how a single line of code and its output is effected by what language it is expressed in, what parameters are selected, and the hardware it is run on). Prioritising code, at the expense of the rest of the stack, places a constraint on developing a holistic, socio-technical understanding of how a digital technology is conceived

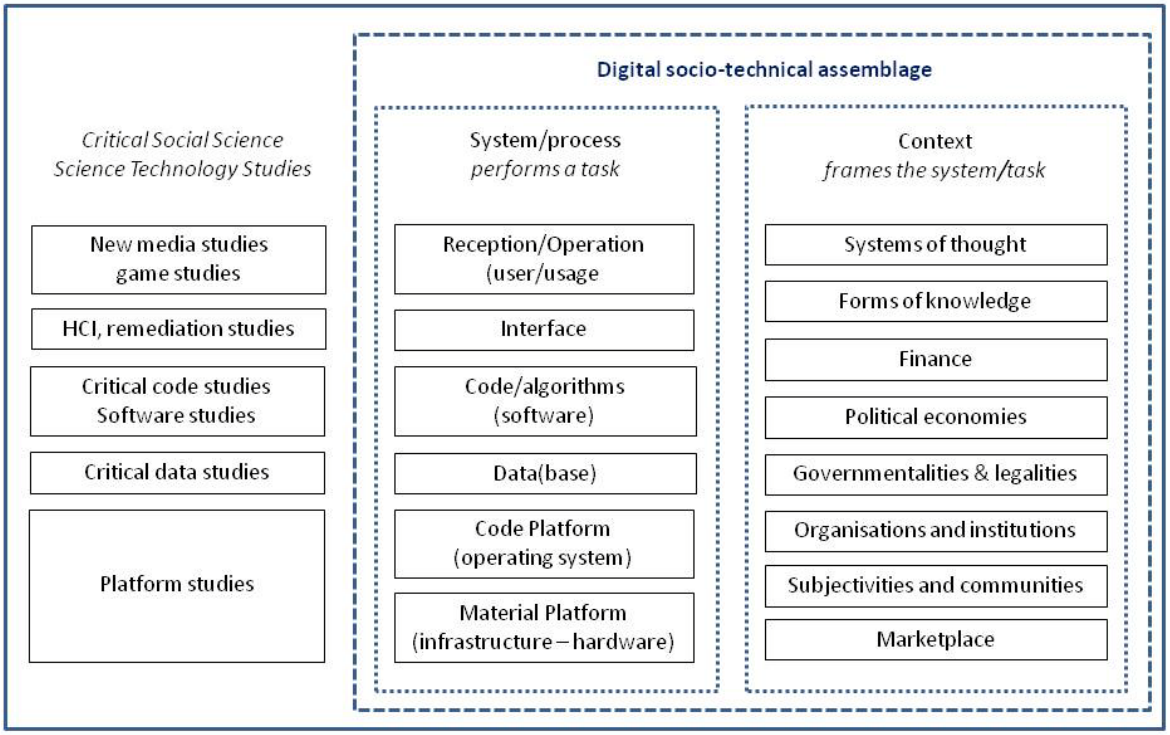
and works in practice (White 2014). This holistic approach is also presently limited by each layer in the stack being the focus of a particular field of study -- new media studies, HCI, software studies, critical data studies, platform studies (see Figure 1).

Figure 1: Digital technology stacks



Taken together, the notion of a data assemblage and technology stack, has led to the creation of an initial wider conceptual framing for *The Programmable City* project (from my perspective -- whether the other ten researchers working on the project subscribe to it is an open question) that intertwines these ideas into an overarching notion of a digital socio-technical assemblage (see Figure 2). Within this perspective, code/software is just one element, albeit a critical one, in a much wide assemblage that frames the interrelationship between code and the city. And making sense of a socio-technical assemblage needs to draw on ideas and empirical insights from a range of fields within critical social science and science and technology studies, including new media studies, game studies, human computer interaction, software studies, critical code studies, critical data studies, platform studies, as well as anthropology, sociology, political science, economics and human geography. Unpacking a digital socio-technical assemblage then is no easy task, but it is manageable as a large case study given it is focused on a single assemblage, such as an program/app/system. The city, however, consists of millions of interconnected socio-technical assemblages, working in concert and contest to transduce the urban condition. A key question then is how to make sense of this dense, interconnected web of assemblages that are constantly working in dynamic flux? It is to this conundrum I now turn.

Figure 2: Conceptualising the constitution of a digital socio-technical system



III Thinking about code and the *city*

The problem with examining in detail individual socio-technical assemblages is that the city largely disappears from view. Certain elements get examined, but in isolation, meaning that a more holistic understanding of how various systems combine and interact to produce the whole is never formulated. Clearly cities are large, complex, multifaceted, open systems and it is all but impossible to fully comprehend all their interlocking systems. Nevertheless, it is possible to map out the ways in which socio-technical assemblages (mis)align, work together, compete, coalesce to form larger assemblages, and so on. To date, very little detailed empirical research has been conducted on how socio-technical assemblages are framed within the wider discursive and material technological terrain and urban landscape, and interact and scale to produce densely instrumented cities. Yet such research would usefully illustrate how networked urbanism is being built and functions in practice.

In contrast, urban studies suffers from the converse problem. Since the early 1990s, as noted in the introduction, a fairly substantial literature on the development of networked urbanism and smart cities has emerged. These studies have focused on examining the effects of networked, digital infrastructure on the management and regulation various urban systems, and urban governance and economy more broadly, providing useful insights into how software-enabled technologies are transforming cities and urban life. However, there is a

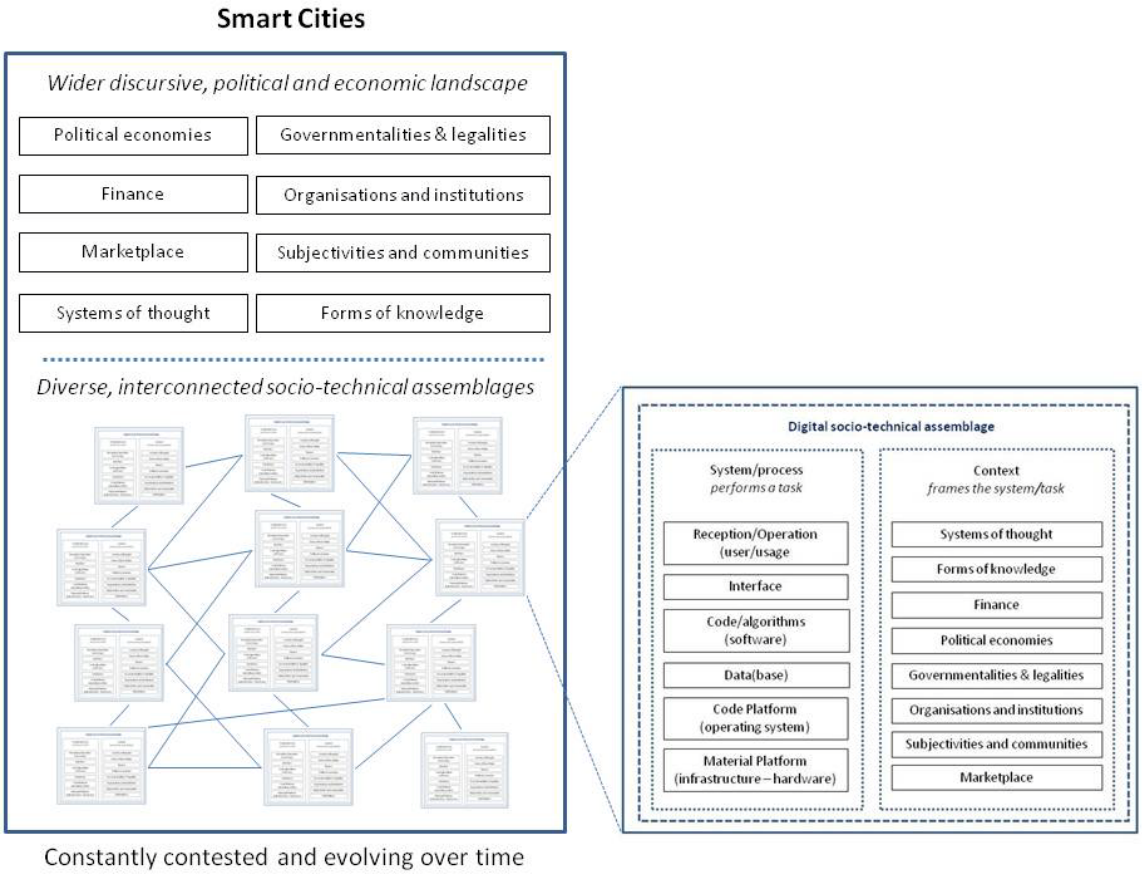
major omission in such work: it discusses the effects of digital socio-technical assemblages, but rarely unpacks the constitution and mechanics of those assemblages. For example, a paper might discuss anticipatory governance and its effects on civil liberties, or the security vulnerabilities of the internet of things and its consequences with respect to privacy, without explicating the specific ways in which systems are configured, code and algorithms work, data are parsed and analyzed, users interface, engage, resist, and so on. In part, this is because the socio-technical assemblages are black-boxed and it takes a bit more effort to leverage access or to undertake approaches that would shine a light into the box (see Kitchin 2014b), but it is mainly to do with adopting a viewpoint that examines effects rather than the causes. In *Code/Space* we illustrated this by comparing approaches that examine the underlying epidemiology of ill-health and the effects of ill-health on the world. Our argument was that whilst one can gain an understanding of the relationship between health and society by studying how ill-health affects social relations, one can gain deeper insights by also considering the specifics of different diseases, their aetiology, and how these manifest themselves in shaping social relations. Similarly, one could examine how telematic networks shape traffic management without studying how such effects are manifestly the result of how the telematic assemblage constituted and configured, with rules and procedures formalised within algorithms and code.

It seems to me, therefore, that we have a major disconnect occurring in the literature. Science and technology scholars are focused on the nature of specific elements of socio-technical systems. Urban scholars are focused on the embedding of digital technologies into urban environments and their social, political and economic effects. Occasionally these perspectives meet, but largely remain apart. A key question, for me at least, is how to marry them into a conceptual whole, or at least place them in productive tension. The solution seems to be to scale the socio-technical perspective up, and drill the urban studies focus down so that they overlap in view and epistemology.

Figure 3 provides an initial attempt at setting out a conceptual framework for what I term ‘programmable urbanism’ -- the instrumented, mutable form of smart cities -- that scales between individual socio-technical assemblages and their components to the city and their dense interconnection and embedding within a wider discursive, political and economic landscape. The framework thus seeks to promote and support research that attempts to simultaneously unpack socio-technical assemblages *and* chart their interconnections and interdependencies and how they scale to frame and create city life. It thus aims to produce a holistic analysis, examining how programmable urbanism is framed within a wider

discursive, political and economic landscape (the rhetoric of smart cities, for example) and how it is built, functions and has effects in practice. The apparatus of 'political economies', 'finance', 'governmentalities & legalities', etc. appear in each socio-technical assemblage and the wider landscape of smart cities to denote that there are a multitude of discursive and material elements at play, some supporting individual assemblages and others the broader terrain of city policy, that often align but can also be in conflict. For example, smart city policy within a city might generally support technocratic forms of governance, but preclude some forms due to legal interventions. Yet there could be active discursive field supporting the rollout of precluded socio-technical assemblages.

Figure 3: A conceptual framework for programmable urbanism



Enacting this framework through empirical study would be an arduous task for an individual, but it is certainly not beyond the bounds of a research team or network of collaborators. It would also be possible to draw insights by stitching together the findings and ideas from across the literature to create a synoptic analysis. It therefore seems plausible

that its vision could be realised, enabling us to gain an enhanced understanding of the relationship between code and the city that scales from lines of code to the city in action.

IV Conclusion

Cities are rapidly becoming composed of digitally-mediated components and infrastructures, their systems augmented and mediated by software, with widespread consequences for how they are managed, governed and experienced. A smart city is not a vision of a future city, as often depicted in the media; it already exists in practice through the millions of interconnected, digital socio-technical assemblages embedded into the fabric of cities that frame how people travel, communicate, manage, play, consume, work, and so on. The challenge for critical scholars is to understand the tightening bonds between code and the city: how such bonds are configured and work in practice, and what they mean for how cities operate and citizen's lives.

My argument in this paper has been that whilst there has been much progress in examining programmable urbanism there is much conceptual and empirical work to be done. To date, the literatures concerning code and the city have remained quite divided, and both have shortcomings. On the one hand, studies that focus on code are narrow in remit, fading out the city, and tend to fetishize code at the expense of the wider socio-technical assemblage within which it is embedded. On the other, studies that focus on the city tend to examine the effects of code but rarely unpacks the constitution and mechanics of the code producing those effects.

My contention has been that we need to marry the ideas within these two literatures to provide a more holistic account of the relationship between code and the city. Building on ideas initially developed in *Code/Space* (Kitchin and Dodge 2011), I have forwarded two, interlinked conceptual frameworks. The first places code within a wider socio-technical assemblage. The second conceives the city as being composed of millions of such assemblages. In so doing, the latter seeks to provide a means of productively building a conceptual and empirical understanding of programmable urbanism that scales from individual lines of code to the complexity of an entire urban system. It is certainly not comprehensive in scope or captures the complex processes and interdependencies at play. But it does, I believe, provide an initial scaffold for seeking to scale software studies up towards the city and to drill urban studies down towards code.

Acknowledgements

The research for this paper was funded by a European Research Council Advanced Investigator Award to Rob Kitchin, entitled 'The Programmable City'. ERC-2012-AdG-323636-SOFTCITY

References

Amin, A. and Thrift, N. (2002) *Cities: Reimagining the Urban.* Polity, London.

Batty, M., Axhausen, K.W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G. and Portugali, Y. (2012) Smart cities of the future. *European Physical Journal Special Topics* 214: 481–518

Berry, D. (2011) The *Philosophy of Software: Code and Mediation in the Digital Age.* Palgrave, London.

Bogost, I. and Montfort, N. (n.d.) Levels. *Platform Studies* http://platformstudies.com/levels.html (last accessed 7 August 2014)

Dodge, M. and Kitchin, R. (2005) Code and the transduction of space. *Annals of the Association of American Geographers* 95(1): 162-80

Dodge, M. and Kitchin, R. (2007) The automatic management of drivers and driving spaces. *Geoforum* 38(2): 264-75

Dourish, P. (2001) *Where The Action Is.* MIT Press, Cambridge, MA.

Foth, M. (2008) *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-time City*. Information Science Reference, New York.

Fuller, M. (2008) *Software Studies: A Lexicon.* MIT Press, Cambridge, MA

Graham, S.D.N. (2005) Software-sorted geographies. *Progress in Human Geography* 29(5): 562-80

Graham, S. and Marvin, S. (2001) *Splintering Urbanism: Networked Infrastructures, Technological Mobilities and the Urban Condition.* Routledge, London.

Kelley, M.J. (2014) The semantic production of space: pervasive computing and the urban landscape. *Environment and Planning A* 46(4): 837–851

Kitchin, R. (2011) The programmable city. *Environment and Planning B* 38: 945-951

Kitchin, R. (2014a) The real-time city? Big data and smart urbanism. *GeoJournal* 79(1): 1-14

Kitchin, R. (2014b) Thinking critically about algorithms. *Programmable City Working Paper 5*. SSRN http://ssrn.com/abstract=2515786

Kitchin, R. and Dodge, M. (2011) *Code/Space: Software and Everyday Life*. MIT Press, Cambridge, MA.

Kitchin, R. and Lauriault, T. (2014) Towards critical data studies: Charting and unpacking data assemblages and their work. Programmable City Working Paper 2, *Social Science Research Network* http://ssrn.com/abstract=2474112

Kitchin, R., Lauriault, T. and McArdle, G. (in press, 2014) Knowing and governing cities through urban indicators, city benchmarking and real-time dashboards. *Regional Studies, Regional Science*

Mackenzie, A. (2006) *Cutting Code: Software and Sociality*. Peter Lang, New York.

Manovich, L. (2013) *Software Takes Control*. Bloomsbury, London.

Mitchell, W.J. (1995) *City of Bits: Space, Place and the Infobahn.* MIT Press, Cambridge, MA.

Montfort, N., Baudoin, P., Bell, J., Bogost, I., Douglass, J., Marino, M.C., Mateas, M., Reas, C., Sample, M. and Vawter, N. (2012) *10 PRINT CHR$ (205.5 + RND (1)); : GOTO 10*. MIT Press, Cambridge, MA.

Shepard, M. (2011) *Sentient City: Ubiquitous Computing, Architecture, and the Future of Urban Space*. MIT Press, Cambridge, MA.

Ribes, D. and Jackson, S.J. (2013) Data bite man: The work of sustaining long-term study. In Gitelman, L. (ed) *"Raw Data" is an Oxymoron*. MIT Press, Cambridge, pp 147-166.

Rosenberg, S. (2007) *Dreaming in Code: Two Dozen Programmers, Three Years, 4,732 Bugs, and One Quest for Transcendent Software.* Three Rivers Press, New York.

Thrift, N. and French, S. (2002) The automatic production of space. *Transactions of the Institute of British Geographers* 27(3): 309-35

Townsend, A. (2013) *Smart Cities: Big Data, Civic Hackers, and the Quest for a New Utopia*. W.W. Norton & Co, New York.

Van Dijck, J. (2013). *The Culture of Connectivity: A Critical History of Social Media*. Oxford: Oxford University Press.