Environment and Planning A 2009, volume 41, pages 1283 – 1293

doi:10.1068/a42133

Guest editorial

How does software make space? Exploring some geographical dimensions of pervasive computing and software studies

Introduction: why software matters

> "There is no reason anyone would want a computer in their home."
>
> Ken Olson, founder of Digital Equipment Corp. (1977)

> "While the last computer revolution placed a single computer in front of a vast majority of our population, the next revolution is poised to place many computers into our environment and onto us."
>
> T Scott Saponas et al, "Devices that tell on you" (2006)

Computers are widely recognised as powerful tools in many aspects of contemporary society. Significantly their agency is now changing as the social and spatial disposition of computers diffuses further into almost all aspects of everyday life. Computers, that increasingly don't look like computers, are permeating domestic spaces, built into appliances like washing machines for example, and accompany us throughout the day (energising our mobile phones, PDAs, and MP3 players), mediating our interactions and facilitating a myriad of mundane activities. Many argue that this is just the beginning of the next wave of digital technological development, the so-called pervasive computing revolution, which according to Anne Galloway (2004, pages 384 – 385), "seeks to embed computers into our everyday lives in such ways as to render them invisible and allow them to be taken for granted." Such computing, that is active-in-absence heralds much more subtle forms of software mediation and automated decision making in the world. It is this code work that is the focus of this theme issue.[1]

The extent to which software is actively engaged in a multitude of tasks, ranging from the profound to the mundane, is easily overlooked yet readily apparent when thoughtfully considered. Close to home, for example, many typical academic work practices increasingly depend on code (cf Borgman, 2007). The practices surrounding the production, dissemination, and consumption of this paper are in many respects prosaic but also powerfully demonstrative of the mediating agency of code. This text was composed and edited through the mediation of the code of Microsoft Word and shared between its authors via e-mail programs, firewalls, routers, and the TCP/IP protocol. It is quite likely that you are reading this paper on a computer display (or printed from a pdf file) having downloaded it from the Envplan website. The text representation on your screen or printed page is, in some senses, just the visible surface of a large realm of software, a complex amalgam of data structures, algorithms, packages, protocols and capta that make the space of reading possible [cf Livingston (2005) on other aspects of the spatiality of reading].

Software matters because of the ways it changes the nature of everyday practices. In many areas of scholarship today there is a reliance on analytical applications (cf Uprichard et al, 2008) and computer databases (cf Hine, 2006), and others on

[1] The genesis of this theme issue was a double session, organised by the editors, at the 2007 Association of American Geographers conference, where preliminary versions of the papers by Dodge and Kitchin, Shaw and Warf, and Sheller were presented. We are grateful to the other participants and the audience at this session for stimulating a number of ideas. The other three papers by Budd and Adey, Leyshon, and Mackenzie were solicited after the AAG meeting specifically for this theme issue.

software simulations (eg Lane et al, 2006), most particularly in the 'hard' sciences but not exclusively so as computational social science grows in importance (Lazer et al, 2009). The life blood of scholarship in terms of journal publishing is thoroughly software mediated, from electronic documents, to manuscript reviewing and distribution as e-journals; teaching practices built upon PowerPoint are common and are increasingly being folded into so-called virtual learning environments[2] with automated quizzes, electronic submission of coursework, and algorithmic plagiarism detection software [which has questionable effectiveness and dubious politics (cf Hayes and Introna, 2005)]. This reliance, and in some cases dependence, on software in scholarship stretches beyond the confines of offices, lecture theatres, and labs as many elements of fieldwork are now also code work (cf Fraser, 2007). Geographers, therefore, are very much software workers, like many other occupations and professions, even though most do not write code and probably could not do so even if they desired.

It could be argued that this inability to programme software (held generally across humanities and social sciences) is a problem as there is a real danger of becoming thoroughly alienated from a key source of creative power in our working practices. As Rob van Kranenburg (2008, page 23) put it, in an analogous context:

> "If as a citizen you can no longer fix your own car—which is a quite recent phenomenon—because it is software driven, you have lost more than your ability to fix your own car, you have lost the very belief in a situation in which there are no professional garages, no just in time logistics, no independent mechanics, no small initiative.... [Citizens] become helpless very soon, as they have no clue how to operate what is 'running in the background', let alone fix things if they go wrong. As such, Ambient Intelligence presumes a totalising, anti-democratic logic."

As computers grow to outnumber people and become invisibly embedded in the environment, their agency clearly poses issues for accepted notions of individual privacy, social autonomy, and democracy. The growing calculative role of code is of greatest concern with its ability to render all kinds of spaces 'machine readable' through identification technologies and fine-scale sensors (Dennis, 2008; Dodge and Kitchin, 2005a) enabling the generation of 'data shadows' and so-called 'lifelogs' that record the minutiae of everyday life with great granularity and, potentially, to never forget what has been captured (Allen, 2008; Dodge and Kitchin, 2007). These extensive and readable spaces can then be interpreted by code which makes decisions automatically including socially significant ones, in terms of access control and anticipatory governance (Adey, 2009; Graham, 2005).

Software also matters because it is big business. The creation, maintenance, and marketing of software products is a major industry in many parts of the world. Microsoft is the largest software firm, by a significant margin, with revenues in 2006 of $44.3 billion and the code it creates has made many of its founding programmers very wealthy (OECD, 2008). This is in part because Microsoft, and the software industry as a whole, is also very profitable, despite the high risks and large R&D expenditures required (Microsoft's R&D investment in 2006 was $6.6 billion). According to OECD analysis (2008, page 37) the average profit margin for the top-ten largest software firms was 23% in 2006, compared with the average for the top-250 ICT firms of 7.7%.

Large sectors of entertainment media production are also, at a fundamental level, now part of a broadly conceived software industry. The media output of television, film, and music is rapidly moving to complete end-to-end digital manufacture, distribution, and consumption with no physical manifestation. (The move of books from physical print

[2] At the University of Manchester all academics are being strongly encouraged to 'Blackboard' their courses. This involves much reconfiguration of teaching materials to fit the parameters and structures of a proprietary software system sold by US corporation Blackboard, Inc.

to digital delivery is slower but happening.) Pure software-based entertainment in the form of video games is also a growing element of mass market entertainment. In 2007 the video game market was estimated to be worth $9.5 billion in revenues in the US (ESA, 2009), equal to the US cinema box office take (MPAA, 2008). The media format of the age, the World Wide Web, is of course wholly a matter of code, quite literally, existing only in software and never having had an analogue existence.

As a product, software also enjoys some unusual qualities. How software is licensed means that liabilities for its failures are limited. Although most people do not read the terms and conditions they agree to when using licensed software, they are willingly accepting an imperfect product whilst abrogating the supplier from responsibility for damage caused. These imperfections in terms of bugs, glitches, and crashes are at once notorious and yet also largely accepted as a routine part of the 'conveniences' of computers. The imperfections are also instrumental evidence that software is a hard thing to produce (cf Charette, 2005).

The imperfections in software matter also because it facilitates new criminal activity along with petty digital vandalism. Illegal hacking, software viruses, and network attacks are an ever present threat in the spaces of software. Many people are unwittingly using computers that are compromised with software elements infected and potentially being utilised by outside forces when networked. The scale of corrupted code and deliberate virus infections is hard to gauge with any reliability, but John Markoff (2007) reports that some 11% of computers on the Internet are infected. Software to secure other software is now itself a significant business opportunity!

Software and space

Software matters today, and it will matter more so in the decade to come, as various aspects of pervasive computing play out. How do we begin to make sense of what this might mean? One way is to analyse the way in which software can, quite literally, make space. Code beckons into being sociospatial relations that are dependent on the effective operation of software; what Dodge and Kitchin (2005b) have called 'code/space'. As such, geographers can potentially contribute valuable new perspectives to the emerging field of 'software studies'.

Software studies seeks to create an expanded understanding of code that extends significantly beyond the technical. It offers cultural and theoretical critiques to how the world itself is captured within code in terms of algorithmic potential and formal data descriptions. A leading theorist in the field, Lev Manovich (2008, page 6), asserts: "I think that Software Studies has to investigate both the role of software in forming contemporary culture, and cultural, social, and economic forces that are shaping development of software itself." The emerging research is transdisciplinary, driven by scholars and intellectual hackers in media theory and new media art. Matthew Fuller (2008, page 2) argues that it "proposes that software can be seen as an object of study and an area of practice for kinds of thinking and areas of work that have not historically 'owned' software, or indeed often had much of use to say about it." There is much, we believe, that needs to be said by geographers who have traditionally not had much to say about the spatiality of software beyond critiques of GIS and cyberspace (see Dodge and Kitchin, 2001; Pickles, 1995).

Thus, the goal of this theme issue is to bring geographical work on code and pervasive computing into alignment with emerging themes of software studies. To do this the six papers in this theme issue offer up a range of conceptual ideas and practical strategies to understand how the diversity of software agency contributes to the production of space. The papers draw on diverse contexts and empirical evidence to make their case, and amply demonstrate the validity of focusing analytical attention on the significance of software as a vigorous agent in the making of spaces.

Of course these papers do stand alone. The theme issue builds on a number of significant research articles in the past five years or so by geographers that have begun the task of describing and explaining the geographies of software with methods and concepts from the social sciences (eg Adey, 2004; Crang and Graham, 2007; Dodge and Kitchin, 2005b; Graham, 2005; Thrift and French, 2002; Zook and Graham, 2007). Allied to this work there are also the noteworthy contributions from economic geographers that have begun to chart the geographical significance of software as products and an industry, here looking primarily at production chains and locational clusters (eg Coe, 1999; Johns, 2006; O'Riain, 2004).

Burgeoning contemporary work in human geography on different aspects of software also has a number of important intellectual roots back, at least to the early 1990s, in terms of privileging the social power of computing, networks, and telecommunications in the understanding of changing geographical landscapes. The significance of the spatial virtualism and new Internet geographies was flagged first by Mike Batty (1993, page 616) when he asserted, in a prescient editorial, that

> "now we have the emergence of cyberspace.... It is largely invisible to conventional methods of observation and measurement....We need to begin to map this space, to visualize its architecture, and to show how it connects to and transforms our traditional geographies. The task before us is urgent, baffling, and exciting... ."

Since Batty's early intervention, to take one avenue of scholarship, there have been a number of theme issues published across human geography journals that have usefully interrogated various aspects of computing, digital technologies, and the nature of emerging virtual spaces. For example, Adams and Warf (1997) edited a foundational special issue in the *Geographical Review* that presented an opening salvo of papers which took seriously the Internet as a phenomenon with important geographical implications, while Dodge's (2001) issue of *Environment and Planning B* and Aoyama and Sheppard's (2003) issue of *Environment and Planning A* gave further consideration to other dimensions of 'cybergeography' and virtual space. More recently Gordon's (2008) theme issue in *Space and Culture* focused on the geography of virtual worlds; Adams's (2007) collection in *Ethics, Place and Environment* presented six papers covering technological change, focusing on digital media and computing; Dixon and Whitehead's (2008) theme issue in *Social and Cultural Geography* drew on ideas from STS to consider the diverse spatiality of technology; lastly Dave (2007) edited an informative theme issue on space, sociality, and pervasive computing in *Environment and Planning B*.

We now sketch out briefly the key conceptual tools that each of the six papers in the theme issue uses to highlight how software makes space. Firstly, and most obviously, software can make space through the capacity of calculation at a scale and speed far beyond human abilities. Code has the ability to gather together data and automatically determine an appropriate action in given contexts that give the appearance of quite complex behaviours. Adrian Mackenzie's paper in this issue discusses an esoteric but essential exemplar of the automatic calculative ability of code in terms of the Viterbi algorithm that provides the 'guts' of digital signal processing which is essential to the operation of mobile phones, along with a growing array of other wireless devices that are at the vanguard of pervasive computing. Importantly, Mackenzie argues that the calculative conduct of software algorithms is qualitatively different from analogue ones and thus requires us to step theoretically beyond 'centres of calculation' (Latour, 1987) to a Deleuzian notion of 'envelopment'. Such a shift is significant because it acknowledges that algorithms create spaces characterised by change that is always changing, what Mackenzie (2009) calls 'intensive movement'. In other words, these are types of spaces in flux that cannot be mapped in certain terms, but can only be guessed at in probabilistic ways; they are spaces existing more in an unseen quantum universe than the experienced fixity of a Newtonian one. The enveloping radio spaces swirling around, between, and through the spaces populated by our mobile phones, exhibit this intensive movement with ever-changing patterns of congestion and contestation between signals that inevitably overlap, disrupt, and inhibit each other. Yet the Viterbi algorithm is able to make sense of the intensive movements in real time because it "assume[s] that we can only hope to determine the most probable series of sent signals" (page 1303) which is at odds with "the images of strict determinism sometimes associated with digital technologies" (page 1304). The result, which we now take for granted, means that the phone in our pocket receives only one clear call despite it being in the midst of cacophonous tumult of competing, continuously changing signals. In short, code is creating the fundamental space in which communicative practices (eg a phone call or text) takes place.

Mackenzie's paper also highlights the necessity of decoding the workings of obtuse algorithms at the heart of software systems, like the mobile phone, but also the real difficulty of doing this in ways that produce meaningful knowledge in a social science sense. This is partly a problem of 'black-boxing'—what he (page 1299) describes as the "submersion of algorithms into commodity hardware". Such obscurity of software's presence and operational logics is apparent in many offline and online settings [cf Zook and Graham's (2007) critique of search engine rankings]. Yet this is not the only issue, because, as Mackenzie (page 1295) notes:

> "the algorithmic processes ... offer a strong challenge for research .... in their somewhat stunning complexity, they seem to bear only a tangential relation to the powerful dynamics of belonging, participation, separation, and exclusion typical of contemporary network cultures."

The result is that "algorithms for wireless [digital signals processing] offer few recognisable social attributes or properties as handholds for critical analysis" (page 1305).

One route to foreground the social attributes in code is to consider how software has augmented, and continues to change and enhance, the creative practices of different economic sectors. As Andrew Leyshon's (2009, page 1309) analysis of the music industry, focused on the practices of recording studios, shows, "code [has] ushered in a regime of distributed musical creativity." Software is important, then, in how it can distribute creative practice both through time and across space, enabling the required consistency and repeatability of practice at much lower costs. Within studios themselves, the engineer's control consoles for mixing and recording the sound are crucial sites of creative work and Leyshon (page 1324) notes that, by "integrating software and memory into the operating of desks, producers and engineers were able to easily reestablish the settings between recording sessions." Leyshon's work also shows the degree to which code can automate some aspects of the embodied and tacit knowledge held by specialist sound engineers, although it is not able to replicate the emotional labour of studio staff who are able to formulate the sense of congeniality deemed significant for the highest calibre of creative musicianship.

The combination of efficient memorialisation and automation of creative practices made possible by code, coupled with a steep drop in equipment costs, has led to a decline in the barriers to entry into professional quality music recording. As Leyshon (page 1309) puts it: "[t]he rise of more affordable digital recording rigs and easier programming protocols represents a democratisation of technology." The competition from cheaper 'bedroom' recording facilities has denuded the exclusive preserve of purpose-built recording studios and contributed to deteriorating employment conditions for skilled sound engineers. More competition has driven down costs across the sector.

As such, software, like many other rounds of technological investment, needs to be understood as a critical agent in economic restructuring with geographical consequences. To understand this large-scale structural change flowing out of the technicity of software, Leyshon (2009) argues, one needs to develop rich historico-geographical accounts of the contexts in which code is embedded into workplaces and labour practices over the time and the new kinds of future trajectories this enables. Otherwise, code is simply seen as an abstract, exogenous factor rather than a socially embedded variable.

Besides a focus on changed labour practices, the work that code does in the world also needs to be analysed critically in terms of how it can affect people's states of being. As Ian Shaw and Barney Warf (2009) note in relation to the potency of the sensory experiences conjured forth by video games, software can be designed to engage and manipulate the emotional registers of players. Indeed, "game spaces", they argue, are "constellations of affect" (page 1333), in which 'affect' is "a pre-cognitive force which disrupts and delights the player with reactions ranging from fear to joy" (page 1332). Increasingly sophisticated and hyperrealistic graphic representations in video games are able to beckon into being believable environments that possess a genuine sense of spatiality, and often intense sociality, that grips players and pulls them into a compelling ludic realm 'beyond' the screen display. Shaw and Warf (page 1340) deploy Deleuzian notions of 'contraction' to account for this interweaving of the material display and affect so that "as the player navigates virtual space, the representations contemplated become embodied, felt, experienced, and lived."

Such contraction is readily evident as you watch the intensity of mental and bodily concentration of game players, which can be at such a pitch that it is unclear "whether we are seeing bodies controlling machines or machines controlling bodies" (Shaw and Warf, 2009, page 1340). Contraction is not at all surprising either, given the immense amount of interactive pleasure that the best designed computer games clearly provide to players. The intensity of enjoyment that code can generate needs to be acknowledged, not least because it explains the popularity of video games as recreation (and not just for children) and their increasing cultural significance (Kerr, 2006; Wark, 2006). Clearly there is much that software studies can gain by analysing how people play with, and through, code; but despite the popularity and inherent sociospatiality of video games and virtual worlds, there is little theoretical work by geographers on their nature and meanings (although see the work of Schwartz, 2006). Moreover, the role of code in creating new spaces of recreation and play also lies beyond the virtual worlds of video games, imbuing toys with new capacities which, as Nigel Thrift (2003, page 400) notes, are "rapidly becoming something else: something between a lumpen object onto which all manner of fantasies and all kinds of play could be projected and a kind of alternative life form, participating in the world on at least some terms of its own choosing."

While software is immaterial in its essential executive state, its agency derives from the ways it can change material things. This theme is the foci of Martin Dodge and Rob Kitchin's (2009a) paper, where they assimilate ideas from pervasive computing research and home geographies, to conceptualise how code is changing the nature of material objects. Their chosen empirical context is important as well, in that they consciously focus on 'messy' home spaces and the typical constellations of everyday objects, tools, and gadgets that are enrolled in daily domestic routines.

Their analytical approach is one of taxonomy building around the notion of 'codejects', where they seek to classify how domestic objects are gaining capacities that extend their technicity and enable them to do additional work in the world. These capacities in turn, they argue, are helping to reshape the home and its spatialities, by augmenting and supplementing domestic tasks and also plugging the home into new, extended networks of service and surveillance. Their empirical discussion shows how a range of technologies and an increasing number of coded objects are used in the daily production of our homes, creating a series of overlapping coded space and code/space. For many technologists, the deployment of a growing number and range of 'codejects' into (Western) domestic spaces is evidence that we are moving to the era of the 'smart home' and widespread pervasive computing. However, Dodge and Kitchin (2009a) conclude by trying to unpack pervasive computing ideology as it plays out through imagined 'smart home' scenarios and utopian predictions of better living. Their conclusion is that the 'smart home' conceived as a totalising code/space environment of fully automated convenience and comfort will not come to pass. Software infusion into homes is significant but partial, and will remain haphazard and incomplete over the coming decade.

The notion of simulation is another conceptual tool to understand how software has the capacity to effect the shape of present spaces by the ways it can predict events and spaces that have yet to occur. Such "simulation models", Lucy Budd and Peter Adey (2009, page 1370) argue, are significant socially because they "enable predictions to be made about uncertain futures and allow users to *run* 'what if' scenarios" (emphasis in original). One result is the production of software simulated space in the now that are premised on a calculation of the state of spatiality in some future time. Perhaps the most obvious exemplar, and one of daily significance to many people, are meteorological software models, where complex algorithms process voluminous data to generate weather forecasts that informs the present. Budd and Adey (page 1367) focus their attention on how "software systems virtualise potential aerial mobilities in an effort to prepare for, prevent, or preempt some future event."

Budd and Adey's (2009) paper also alerts us to the scope for code to work in an anticipatory fashion, particularly in the domain of surveillance and governmentality. Here the actual predictions, created algorithmically and automatically by code, do work in the world to *prevent* that future scenario from coming into being. Such preemptive mechanisms have much appeal in the risk-conscious and real-time world of global mobility, but clearly raise serious issues of ethics and power. In space – times where anticipatory governance using software simulations is active, how can people be sure of the social equity in the design of the code that affects, very materially, their life chances? Partly this is a 'disappearance' issue, as Budd and Adey (2009, page 1370) point out that when simulations "move into the public domain their inherent uncertainties and qualifications may be forgotten and the public seduced into accepting their 'crystal-ball'-like assumptions." It is also that people often have little, if any knowledge, that they have been subjected to anticipatory governance and consequently have no recourse to challenge the validity of the simulated future that they were never able to see into actuality. (When you arrive at the airport to board the flight you have no idea what risk and threat calculations have been applied to you since you booked your ticket.) Similar issues have been raised in terms of consumption, where geodemographic and algorithmic risk models potentially curtail people's life chances without their knowledge or consent (cf Graham, 2005; Parker et al, 2007).

In common with Shaw and Warf's (2009) stress on the role of affect for a nuanced reading of the potency of code to do work in the world, Budd and Adey (2009) also make the case that to understand software simulations in the airworld one needs to incorporate the corporeal feelings and visceral sensations in people generated by code. Here they flag a specific strand of computer science research known as 'affective computing' that seeks to extend and enhance "human computer interaction by including emotional communication together with appropriate means of handling affective information" (Picard, 1999, page 1). In their discussion of evermore realistic cockpit simulations used to train airline pilots, for example, they argue the subconscious and emotional coupling of living bodies and digital code is vital to how they work. This is because "the simulation is not only being used to create a situation, it is used to condition a pilot's response to it" (page 1376). Such responses are deeply affectual and vital to their effectiveness. As they note, "[f]light simulators ... deliberately *create* potential emergencies in order *to* create surprise, confusion, fear, and shock, so that the feeling of a feeling may become known. This kind of preparedness tests the mental resilience and ability of pilots to cope with such a future" (page 1380, original emphasis).

Software can also be analysed in terms of its conceptual, and also realised, capacities to effect scale and sovereignty. This is the focus of Mimi Sheller's (2009) examination of the roles of code in the construction of elite leisure spaces in the Caribbean. Here a swath of coded infrastructures works in concert to rescale territorial relations, to manage the mobilities of tourists and workers, and in crucial place-making and promotion activities necessary for such ambitious schemes. Software is critical to create exclusive resort spaces that, in some senses, deploy code to float free from the constraining jurisdictions of their geographical position. "Colonial and postcolonial spaces within the Caribbean", Sheller (page 1388) argues, "may actually be at the forefront of contemporary processes of cyberspatial state restructuring insofar as they lead the way in unfurling new forms of national territory, in both real and virtual spaces." Her analysis also highlights the degree to which software codes collide with legal codes resulting in strategic refashioning of spaces in new ways. These refashioning actions are often exclusionary in form; thus highlighting that code calculations can create social difference. As Sheller explains (page 1399), "[n]ew forms of infrastructural exclusivity, computer-aided design, media-savvy web-based property marketing, and uneven forms of software-sorted mobility underwrite proprietary regimes that assist in channeling who has access (or does not) to various kinds of real estate and residency rights." [See also Graham (2005) and Parker et al (2007) for further consideration of inequities created by code.]

Sheller's (2009) cogent analysis also demonstrates the purposeful nature of code in the processes of spatial planning, landscape design, and, most especially, architectural practices to beckon into being virtual properties and imagined islands that appear so real they can affect future decisions. Her case study of the impact of off-shore software in imagining Dellis Cay in the Turks and Caicos Islands also highlights how code does work in the world in remote, peripheral, and nonmetropolitan places—these are typically overlooked in the analysis of pervasive computing that tend to focus on urban spaces and core sites (cf Dodge and Kitchin, 2009b).

Towards a geographical agenda for software studies

It is readily apparent that software is special. Equally significant for geographers is the degree to which software is also spatial; an issue that will become only more important as pervasive computing unfolds in the social world. Moreover, we believe that geographical approaches have the potential to contribute significantly to software studies. For example, empirically informed analysis that unpacks the 'automatic production of space' (Thrift and French, 2002) in terms of the everyday experience of living within (and increasingly living through) coded environments will also be important to scholarship that is itself being effected by software in evermore significant ways. Likewise, there are real opportunities for research examining the power of code enacted spatially in the processes of governmentality by states and sophisticated management by corporations, along with the need to understand the productive capability of software to reformulate collective life and enhance individual's spatiality in creative, playful, empowering ways.

Among the many questions to be asked are: To what extent is software acting with its own agency in the automatic production of space? Where is the division between the agency of software programmers, the agency of the software, and the agency of the user? Who controls code and how does this control change over time? How do the intentional and unintentional sorting of software affect the prospects of different classes of people and places? How do concepts of near and distant, codified and tacit, evolve in concert with software? Does software exhibit particular kinds of scalar effects? What inequalities are exacerbated by, and what new digital differences or lags are created by, the uneven distribution of software use? To what extent are these divides planned as a part of the software writing and deployment? To what extent are they simply unintentional outcomes of how code encounters with the world? Where is code made particularly visible and when does it hide its role? How does the increasing use of mobile devices impact the distribution, use, and role of code? How will the growing significance of spatially oriented code (eg, satnav and Google Maps) impact the role of software in everyday lives?

Overall, the set of papers in this issue provide a useful contemporary review of different developments under what can be labelled, broadly, as 'the spatiality of pervasive computing'. They provide a range of conceptual tools around calculation, simulation, affect, and anticipation. These papers, however, are just a starting point in teasing out larger theoretical themes into a convincing narrative of the spatiality of software. As such the lasting value of this set of papers is, in our opinion, going to be significant. Yet, there remains a large number of further questions to be researched on the socialtechnical dimensions of pervasive computing and the changing landscape of 'code/space'. Building upon these papers and pursuing the questions outlined above will provide us with better descriptions, explanations, and responses to the subtle agency of software in every space.

Acknowledgements. We would like to thank Michael Brown as editor of *Social and Cultural Geography* who organised the initial referring of some of the articles in this theme issue. We also gratefully acknowledge Nigel Thrift who supported this theme issue, Ros Whitehead for helping it come together logistically, the contributors, and the anonymous reviewers whose useful comments and suggestions contributed to the quality of the papers.

Martin Dodge, Department of Geography, University of Manchester

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

Matthew Zook, Department of Geography, University of Kentucky

References

Adams P, 2007, "Introduction to technological change special issue" *Ethics, Place and Environment* **10**(1) 1 – 6

Adams P C, Warf B, 1997, "Introduction: Cyberspace and geographical space" *Geographical Review* **87** 139 – 145

Adey P, 2004, "Surveillance at the airport: surveilling mobility/mobilising surveillance" *Environment and Planning A* **36** 1365 – 1380

Adey P, 2009, "Facing airport security: affect, biopolitics, and the preemptive securitisation of the mobile body" *Environment and Planning D: Society and Space* **27** 274 – 295

Allen A L, 2008, "Dredging up the past: lifelogging, memory, and surveillance" *The University of Chicago Law Review* **75**(1) 47 – 74

Aoyama Y, Sheppard E, 2003, "Guest editorial: The dialectics of geographic and virtual space" *Environment and Planning A* **35** 1151 – 1156

Batty M, 1993, "Editorial: The geography of cyberspace" *Environment and Planning B: Planning and Design* **20** 615 – 616

Borgman C L, 2007 *Scholarship in the Digital Age* (MIT Press, Cambridge, MA)

Budd L, Adey P, 2009, "The software-simulated airworld: anticipatory code and affective aeromobilities" *Environment and Planning A* **41** 1366 – 1385

Charette R N, 2005, "Why software fails" *IEEE Spectrum* September, pp 42 – 49

Coe N M, 1999, "Emulating the celtic tiger? A comparison of the software industries of Singapore and Ireland" *Singapore Journal of Tropical Geography* **20**(1) 36 – 55

Crang M, Graham S, 2007, "Sentient cities: ambient intelligence and the politics of urban space" *Information, Communication and Society* **10** 789 – 817

Dave B, 2007, "Guest editorial: Space, sociality, and pervasive computing" *Environment and Planning B: Planning and Design* **34** 381 – 382

Dennis K, 2008, "Sensoring the future: complex geographies of connectivity and communication" *World Futures* **64** 22 – 33

Dixon D P, Whitehead M, 2008, "Editorial: Technological trajectories: old and new dialogues in geography and technology studies" *Social and Cultural Geography* **9** 601 – 611

Dodge M, 2001, "Guest editorial: Cybergeography" *Environment and Planning B: Planning and Design* **28** 1 – 2

Dodge M, Kitchin R, 2001 *Mapping Cyberspace* (Routledge, London)

Dodge M, Kitchin R, 2004, "Flying through code/space: the real virtuality of air travel" *Environment and Planning A* **36**(2)195 – 211

Dodge M, Kitchin R, 2005a, "Codes of life: identification codes and the machine-readable world" *Environment and Planning D: Society and Space* **23** 851 – 881

Dodge M, Kitchin R, 2005b, "Code and the transduction of space" *Annals of the Association of American Geographers* **95** 162 – 180

Dodge M, Kitchin R, 2007, " 'Outlines of a world coming into existence': pervasive computing and the ethics of forgetting" *Environment and Planning B: Planning and Design* **34** 431 – 445

Dodge M, Kitchin R, 2009a, "Software, objects, and home space" *Environment and Planning A* **41** 1344 – 1365

Dodge M, Kitchin R, 2009b, "Software surveillance of farms, farmers and food", paper presented at the Association of American Geographers Meeting, 27 March, Las Vegas

ESA, 2009, Entertainment Software Association: Industry Facts, http://www.theesa.com/facts/index.asp

Fraser A, 2007, "Coded spatialities of fieldwork" *Area* **39** 242 – 245

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

Galloway A, 2004, "Intimations of everyday life: ubiquitous computing and the city" *Cultural Studies* **18** 384 – 408

Gordon E, 2008, "The geography of virtual worlds: an introduction" *Space and Culture* **11** 200 – 203

Graham S D N, 2005, "Software-sorted geographies" *Progress in Human Geography* **29** 562 – 580

Hayes N, Introna L, 2005, "Systems for the production of plagiarists? The implications arising from the use of plagiarism detection systems in UK universities for Asian learners" *Journal of Academic Ethics* **3**(1) 55 – 73

Hine C, 2006, "Databases as scientific instruments and their role in the ordering of scientific work" *Social Studies of Science* **36** 269 – 298

Johns J, 2006, "Video games production networks: value capture, power relations and embeddedness" *Journal of Economic Geography* **6** 151 – 180

Kerr A, 2006 *The Business and Culture of Digital Games: Gamework/Gameplay* (Sage, London)

Lane S N, Brookes C J, Heathwaite A L, Reaney S, 2006, "Surveillant science: challenges for the management of rural environments emerging from the new generation diffuse pollution models" *Journal of Agricultural Economics* **57** 239 – 257

Latour B, 1987 *Science in Action* (Harvard University Press, Cambridge, MA)

Lazer D, et al, 2009, "Computational social science" *Science* **323** 721 – 723

Leyshon A, 2009, "The Software Slump?: digital music, the democratisation of technology, and the decline of the recording studio sector within the musical economy" *Environment and Planning A* **41** 1309 – 1331

Livingston D N, 2005, "Science, text and space: thoughts on the geography of reading" *Transactions of the Institute of British Geographers, New Series* **30** 391 – 401

Mackenzie A, 2009, "Intensive movement in wireless digital signal processing: from calculation to envelopment" *Environment and Planning A* **41** 1294 – 1308

Manovich L, 2008 *Software Takes Command* 20 November version, http://lab.softwarestudies.com/2008/11/softbook.html

Markoff J, 2007, "Attack of the zombie computers is growing threat" *The New York Times* 7 January, http://www.nytimes.com/2007/01/07/technology/07net.html

MPAA, 2008 Motion Picture Association of America: Research & Statistics, http://www.mpaa.org/researchStatistics.asp

OECD, 2008 *OECD Information Technology Outlook 2008* (Organisation for Economic Co-operation and Development, Paris)

O'Riain S, 2004, "The politics of mobility in technology driven commodity chains: developmental coalitions in the Irish software industry" *International Journal of Urban and Regional Research* **28** 642 – 663

Parker S, Uprichard E, Burrows R, 2007, "Class places and place classes: geodemographics and the spatialization of class" *Information, Communication and Society* **11** 901 – 920

Picard R W, 1999, "Affective computing for HCI" mimeo, http://affect.media.mit.edu/pdfs/99.picard-hci.pdf

Pickles J, 1995 *Ground Truth: The Social Implications of Geographic Information Systems* (Guilford, New York)

Saponas T S, Lester J, Hartung C, Kohno T, 2006, "Devices that tell on you: The Nike+iPod Sport Kit", University of Washington Computer Science and Engineering Technical Report UW-CSE-06-12-06. ftp://ftp.cs.washington.edu/tr/2006/12/UW-CSE-06-12-06.pdf

Schwartz L, 2006, "Fantasy, realism, and the other in recent video games" *Space and Culture* **9** 313 – 325

Shaw I G R, Warf B, 2009, "Worlds of affect: virtual geographies of video games" *Environment and Planning A* **41** 1332 – 1343

Sheller M, 2009, "Infrastructures of the imagined island: software, mobilities, and the architecture of Caribbean paradise" *Environment and Planning A* **41** 1386 – 14043

Thrift N, 2003, "Closer to the machine? Intelligent environments, new forms of possession and the rise of the supertoy" *Cultural Geographies* **10** 389 – 407

Thrift N, French S, 2002, "The automatic production of space" *Transactions of the Institute of British Geographers, New Series* **27** 309 – 335

Uprichard E, Burrows R, Byrne D, 2008, "SPSS as an 'inscription device': from causality to description?" *The Sociological Review* **56** 606 – 622

van Kranenburg R, 2008 *The Internet of Things: A Critique of Ambient Technology and the All-seeing Network of RFID* Network Notebooks, http://networkcultures.org/wpmu/portal/publications/network-notebooks/the-internet-of-things/

Wark M, 2006, "GAM3R 7H30RY" *Cultural Politics* **2** 213 – 223

Zook, M, Graham M, 2007, "The creative reconstruction of the internet: Google and the privatization of cyberspace and digiplace" *Geoforum* **38** 1322 – 1343

© 2009 Pion Ltd and its Licensors