[De] Coding Architecture

Open source methods of spatial simulation

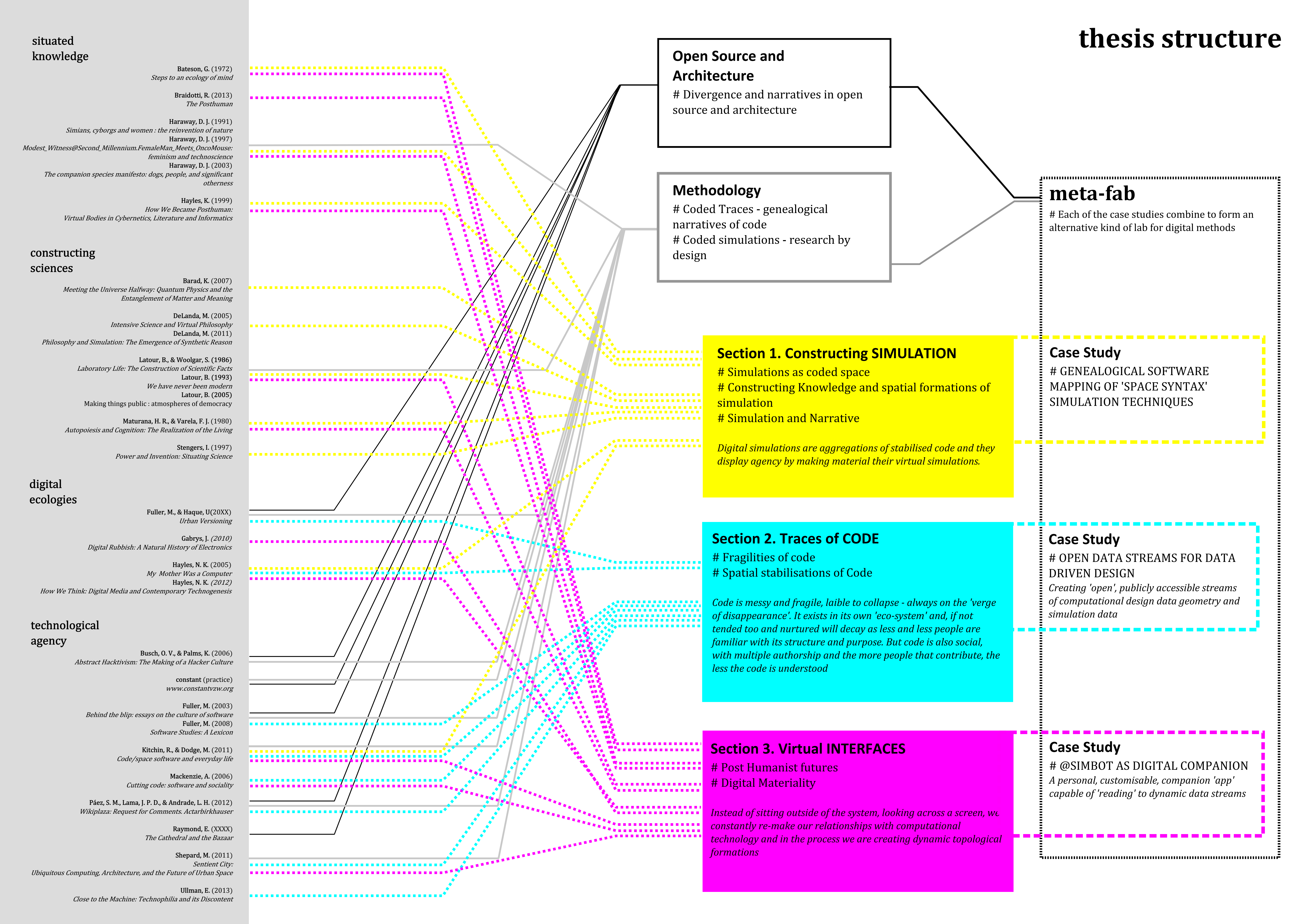
Detailed Report

P. Langley

Sept 2013

Contents

0. Thesis Structure



thesis structure
situated knowledge
Bateson, G. (1972)
Steps to an ecology of mind
Braidotti, R. (2013)
The Posthuman
Haraway, D. J. (1991)
Simians, cyborgs and women : the reinvention of nature
Haraway, D. J. (1997)
Modest_Witness@Second_Millennium.FemaleMan_Meets_OncoMouse: feminism and technoscience
Haraway, D. J. (2003)
The companion species manifesto: dogs, people, and significant otherness
Hayles, K. (1999)
How We Became Posthuman: Virtual Bodies in Cybernetics, Literature and Informatics
constructing sciences
Barad, K. (2007)
Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning
DeLanda, M. (2005)
Intensive Science and Virtual Philosophy
DeLanda, M. (2011)
Philosophy and Simulation: The Emergence of Synthetic Reason
Latour, B., & Woolgar, S. (1986)
Laboratory Life: The Construction of Scientific Facts
Latour, B. (1993)
We have never been modern
Latour, B. (2005)
Making things public : atmospheres of democracy
Maturana, H. R., & Varela, F. J. (1980)
Autopoiesis and Cognition: The Realization of the Living
Stengers, I. (1997)
Power and Invention: Situating Science
digital ecologies
Fuller, M., & Haque, U(20XX)
Urban Versioning
Gabrys, J. (2010)
Digital Rubbish: A Natural History of Electronics
Hayles, N. K. (2005)
My Mother Was a Computer
Hayles, N. K. (2012)
How We Think: Digital Media and Contemporary Technogenesis
technological agency
Busch, O. V., & Palms, K. (2006)
Abstract Hacktivism: The Making of a Hacker Culture
constant (practice)
www.constantvzw.org
Fuller, M. (2003)
Behind the blip: essays on the culture of software
Fuller, M. (2008)
Software Studies: A Lexicon
Kitchin, R., & Dodge, M. (2011)
Code/space software and everyday life
Mackenzie, A. (2006)
Cutting code: software and sociality
Páez, S. M., Lama, J. P. D., & Andrade, L. H. (2012)
Wikiplaza: Request for Comments. Actarbirkhauser
Raymond, E. (XXXX)
The Cathedral and the Bazaar
Shepard, M. (2011)
Sentient City: Ubiquitous Computing, Architecture, and the Future of Urban Space
Ullman, E. (2013)
Close to the Machine: Technophilia and its Discontent
Open Source and Architecture
Divergence and narratives in open source and architecture
Methodology
Coded Traces - genealogical narratives of code
Coded simulations - research by design
meta-fab
Each of the case studies combine to form an alternative kind of lab for digital methods
Section 1. Constructing SIMULATION
Simulations as coded space
Constructing Knowledge and spatial formations of simulation
Simulation and Narrative
Digital simulations are aggregations of stabilised code and they display agency by making material their virtual simulations.
Case Study
GENEALOGICAL SOFTWARE MAPPING OF 'SPACE SYNTAX' SIMULATION TECHNIQUES
Section 2. Traces of CODE
Fragilities of code
Spatial stabilisations of Code
Code is messy and fragile, laible to collapse - always on the 'verge of disappearance'. It exists in its own 'eco-system' and, if not tended too and nurtured will decay as less and less people are familiar with its structure and purpose. But code is also social, with multiple authorship and the more people that contribute, the less the code is understood
Case Study
OPEN DATA STREAMS FOR DATA DRIVEN DESIGN
Creating 'open', publicly accessible streams of computational design data geometry and simulation data
Section 3. Virtual INTERFACES
Post Humanist futures
Digital Materiality
Instead of sitting outside of the system, looking across a screen, we constantly re-make our relationships with computational technology and in the process we are creating dynamic topological formations
Case Study
@SIMBOT AS DIGITAL COMPANION
A personal, customisable, companion 'app' capable of 'reading' to dynamic data streams

1. Open Source Narratives

'Open source' is now a widely understood concept, at least in very broad terms. In software development, at its basic level, in simple terms open source involves the release of a software's 'source code' and allows the user the freedom to;

1. use the software for any purpose
2. study how it works
3. redistribute copies
4. make modifications

(http://www.gnu.org/philosophy/free-sw.html)

'Open source', as a concept has its roots in technological developments that pre-date computing and software. Open standards supporting the exchange of design information for industries such as car manufacturing were commonplace in the first half of the twentieth century. In the 1950s large technology companies such as IBM, then at the forefront of computing, where sharing the 'source-code' of the 'compiling software' for their early computers (Weber 2004) and it is computing that became most synonymous with the idea of open source. The Linux operating system being arguably its most extensive and well-known open source project. By the late 1990s, open source was being 'formalized' in the field of software and programming, with the publication of the source code of the early internet browser 'Netscape' and the publication of Eric Raymond's book *The Cathedral and The Bazaar* (2001), in which he describes two models of open source. In the 'Cathedral', source code is made available only on each stable release of the software, where as in the 'Bazaar', the code is freely accessible throughout a 'public' development process. Raymond used these analogies to indicate the multiple ways in which something can be 'open source' and of the need for a clear understanding of them.

Convergent models of open source

The difficulties that the 'Cathedral' model highlight are twofold. Firstly, the opportunity to contribute is limited as an expert community of coders coalesce around the stable code and become the gatekeepers for modifications proposed by the less skilled contributors. The release of the source code does not in itself widen the pool of potential contributors beyond the expert knowledge that would have been required regardless of whether the code was open source (Fuller 2003). The second, common criticism of such simple models of open source software is that it merely replicates commercial programs rather than providing creative alternatives (Fuller 2003).

In architecture the software tools employed in architectural design practice are typically commercial and closed, therefore already placing certain limits on the ability of non-professionals to adopt or adapt them. The focus has instead been on the release of architectural design as 'source code'. The 'wiki' has been commonly adopted as the method for the release of this 'source code' and as a model of open source, lies somewhere on the road between Raymond's Cathedral and his Bazaar. Whilst the wiki represents an 'always live' resource awaiting editing, wikis are typically heavily regulated, with a relatively small group of editors compared to users.

The 'open source manifesto' for architecture is itself a multi-authored page of wikipedia (http://en.wikipedia.org/wiki/Opensource_Architecture). The manifesto identifies the need for various 'standards' including software and hardware interoperability. Initiatives such as 'wiki house' (www.wikihouse.org) offer 3D models of designed and verified component and assemblies for users to download, manufacture and build their own project. The project Wikiplaza by Hackitectura (Páez, Lama, and Andrade 2012), offers a more diverse resource within its 'source code' and asks the question 'what would it be like to build a wiki?'. Their proposal, conceived as 'copy left' concept, aims to take the methodological approach of the digital wiki and apply it, almost directly, to the physical world.

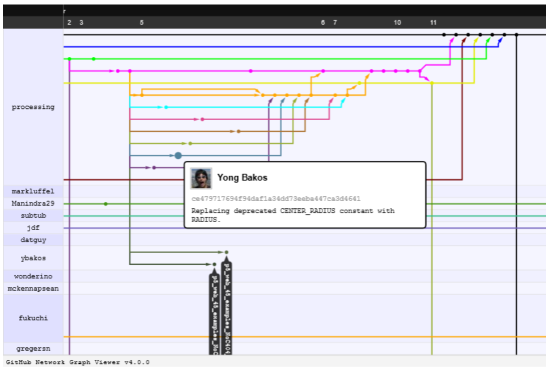
Wiki-like approaches to open source methodologies have their limitations. While they contribute to and support the idea of an open source more generally, they display an underlying 'convergent' characteristic and tend towards a single, stable outcome that is at odds with the spirit of the goals of open source.

Divergent models of open source

Raymond's 'Bazaar' model is about increasing the involvement of non-experts in coding projects in order to create a larger, public group capable of testing and fixing a project in constant development.

A software technology central to this 'divergent' model of open source is 'versioning' which can be used to manage multiple copies of software programs (and parts of programs) that are at various stages of development and stability. 'Versioning' is a method that has been instantiated as a tool in programs such as CVS, SVN and most recently Git (http://git-scm.com).

The core functionality of versioning is the ability for multiple uses to 'branch' individual components of a piece of software assembly, whilst retaining the functionality of the full assembly. This feature allows for multiple, co-authored versions to exist simultaneously. Users can then modify files and 'commit' them back to the assembly, 'merging' them with other branches of the project. This process creates a complete, genealogical trace of each step in the development of a project that can be accessed at any moment during a project's life.



Screen grab of branching project graph from
the popular open source coding software Processing
https://github.com/processing/processing/network

This 'divergent' model of open source makes the writing of code a social act (Yuill 2008). Software such as Github (www.github.com), based on the git versioning technology *Git*, has become a platform for the 'open sourcing' of non-coding or software projects, such as the development of legal documents or the tracking of DIY home improvements *(The GitHub Generation: Why We*'re All in Open Source Now (Wired UK) 2013). The wider significance of versioning, beyond its use in software development has most interestingly been explored by Fuller and Haque in the manifesto for 'Urban Versioning' (2008). They propose a methodology of versioning for architectural and urban design that challenges conventional approaches to design, making and authorship.

Open sourcing knowledge

Open source, of course, has a significance beyond the specifics of this particular narrative. It has been abstracted from its technological origins and applied to ideas of politics, law, authorship and access to knowledge. Lawrence Lessig, through the claim that "Code is Law" (Lessig 2006), identifies the wider importance of the 'open source' movement in challenging the ways in which code is used to exert social control. The issue of 'copyright' is central to this and Lessig advocates a 'Free Culture' approach to authorship and ownership of content (Weber 2004). Such issues are visible in the establishment of 'digital media labs' the more recent phenomenon of 'Fab-labs', that are fuelling the so-called 'makers revolution' (Anderson 2013). Crucial to these movements is the provision of open access to knowledge and skills. Labs such as *Constant* in Brussels (www.constantvzw.org) who adopt a '(cyber) feminist' approach or *Access Space* in Sheffield (access-space.org), represent the new 'Bazaar' and provide a new type of public (digital) space.

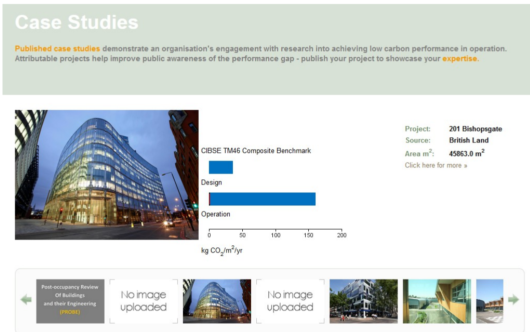
2. Simulation in Spatial Design

The use of simulation in spatial design includes 'well understood' applications such as structural design packages (for sizing steelwork members) and environmental deign software (used for predicting internal temperatures and energy use). In both cases, they are used to provide an evaluation of the predicted performance of the proposed design and in order to do this, the spatial deign model is placed within a simulated world, based on abstractions and simplifications of things such as gravity, wind, weather. In one sense, these kinds of approaches are a development of 'traditional' approaches to the technical aspects of design, where the traditional, by-hand calculations have been replaced by rapid and multiple iterations of complex design scenarios.

There is also another strand of simulation emerging in architectural design. Approaches by groups such as Space Syntax (Hillier et al. 1976; Hillier 2007) represent a break from simulation as augmentation to a pre-existing design operation. Space-syntax analysis investigates characteristics of space such as 'depth of view' – that is how far can someone see – or axial connectivity within geometric models of buildings and urban environments. The results of this analysis are used to evaluate the performance of a design proposal. This approach to spatial analysis is not an augmentation of existing design processes through increased computer processing power, it exists as a direct consequence of that computing power.

These two strands of simulation – augmented processes and invented processes – appear very different and on the surface their only real similarity seems to be that they are both reliant on the availability of computing power to support their use. However, this reliance on computing power goes deeper than a question of 'horse-power'. In deploying such power to create complex simulation models both approaches create a difficulty in evaluating and challenging their outcomes.

In the recent past, research into building performance was translated into rules of thumb guidance that could be applied 'manually', or as laborious hand calculations of things like daylight requirements. Rapid increases in computing power and software sophistication have changed the way in which these calculations are made and conclusions are drawn. It influences not just the types of space that may be constructed, but it shapes the regulatory framework in which the designers operate.



Screen grab of Carbon Buzz website
http://www.carbonbuzz.org/casestudiestab.jsp

The *Carbon Buzz* (www.carbonbuzz.org) initiative creates a platform for building designers and operators to compare simulated and actual environmental performance. While it is currently far from a comprehensive database (the project is still in its early stages), it highlights very clearly three crucial issues. The first, and most obvious, is the discrepancy between the 'designed' and the 'actual' performance. Real energy use is shown to be significantly higher than predicted by the design models. In some cases the difference is so significant that if the 'true' figures were presented at the design phase, the project would not have met the relevant regulatory requirements. Much valuable research has been and continues to be carried out identifying the reasons for such discrepancies, examining such things as end user familiarity with the building's control systems. But there is little focus on designers familiarity with the issues of adopting such computational simulation in the first place. The second aspect is linked to the ease with which this discrepancy can be identified. Both the predicted and measured performance of building is reduced to a single 'grade' - A, B, C etc. The complexity of the simulated model - thousands of triangular faces of geometry, lists of materials, calculations made for every hour of every day - has been simplified to a single character. The final observation is that each design proposal is evaluated and presented in isolation and is not situated in any way. There is no link to its wider 'real-world' context and the simplified environment of the simulation is transposed into the world 'as is'.

Image extract of Elephant & Castle Crossing Project from Space Syntax Website
http://www.spacesyntax.com/project/elephant-castle-crossing/

Although an invented process such as Space Syntax, remains a far from common approach in architectural and urban design and does not (currently) have the same impact on policy. Nevertheless it can be said to have 'left the lab' and has been used on many high profile projects. Even though it is not a widespread approach it is still very difficult to challenge the findings of a study from *Space Syntax*. It is of course, supported by significant and 'robust' research but it is also the nature of the simulation itself that makes contesting it difficult. If I wished to challenge the 'accuracy' of the simulation I would be, by implication, accepting the ontological validity of the approach. I would need a kind of anti 'space syntax' device in order to fully challenge it's usefulness. In returning this idealised virtual model directly into the 'real world', Space Syntax creates its own essentialist ontological position. That is to say that it makes a claim that that there are characteristics of spatial assemblies that can be found, understood and harnessed. The complexity of the computer technology used to generate this position is the means by which it is reinforced.

These two examples of simulation in spatial design sit on a spectrum of approaches that are currently used in architectural practice. At one end, computationally-augmented simulations of environmental performance directly inform the policies than regulate construction. At the other end, computationally-driven simulations of abstracted spatial concepts provide seemingly incontestable analysis of 'good' design. In each case, the ability to engage with the way in which they are produced and the findings they produce are limited to a very narrow professional field.

3. Critical Literature

Section 1. Constructing Simulations

> *"Technologies and scientific discourses can be partially understood as formalizations i.e. frozen moments, of the fluid social interactions constituting them, but they should also be viewed as instruments for enforcing meanings"*
>
> (Haraway 1991:p164)

Scientific Fact(s)

Are there such things as facts, that we can build devices in order to discover or is 'reality' defined by experience? The former, a 'realist' point of view traditionally seen as one adopted by 'hard sciences' such as conventional physics, where objects have a mass that can be measured. The latter, 'constructivist' position, is not concerned with the characteristics or 'essentials' of things, but instead only allow things to be defined by how they are 'interpreted'. This has been seen as the position of 'softer sciences' and humanities. Is their a place for both of these standpoints and if so, how can these two, contradictory positions, be resolved?

Bruno Latour and Steve Woolgar, in their book *Laboratory Life* (1986), describe through ethnographic study, the processes of 'making science' in a biological laboratory, and describe something that is far messier than the 'public face of science'. Latour and Woolgar are interested in 'materiality' as the means of exploring the scientific practices that 'construct' the output of the lab, namely scientific facts. This focus on materiality leads to one of their conclusions, that rather than producing scientific fact, the primary output of the lab is paper. Furthermore, they make a radical step in linking directly the process of experimentation and the production of the printed output. They describes this as a single process, in which the materiality of the experiment if transcribed into the materiality of the document and it is the document, rather than the material of the experiment itself, that become the scientists focus of analysis.

They state that it is the diagrams that record the results of an experiment are the means by which 'facts are constructed'. The shape of a graph's curve, as 'inscribed' is used to indicate the status of a given substance and that the presence of two or more similar statements (or inscriptions) could elevate a substance from subjective statement to ontological 'reality'. The 'inscription', and by extension the 'inscription device', as part of a single material process, are for Latour and Woolgar, critical in the construction of 'fact'.

In presenting their ethnographic account of scientific practice, they do not claim it to be a 'privileged' account. Nevertheless it is a strongly 'constructivist' expression, describing as it does, the information flow from the symbolic, through the operational to the material. Katherine Hayles states the limitations of such a rigid constructivist approach, asserting that it is limited by its adherence [albeit reversed] of the thing which it rejects, namely the realist approach (Hayles 2005). Hayles goes on to claim that Latour himself has since 'softened' his earlier constructivist standpoint and adopt the position that;

> “...the objects of scientific research are at once discursively constructed, socially produced *and* materially real...” (Hayles 2005:209)

Multiple realities

Manuel De Landa attempts to make a description of just such 'objects of scientific research' through an explanation of the philosophical position of Deleuze, both ontologically and epistemologically (De Landa 2005). De Landa makes the assertion that Deleuze is a 'realist' philosopher rather than a 'constructivist' but differs from other realist positions that consider 'things' to have a set of properties that can be used to define and describe it, and that Deleuze contests not only this idea of 'essences', but also the implication that they are unchanging.

Deleuze's ontology, explains De Landa, is about doing away with 'essences' and replacing them with dynamic processes of difference. It is 'morphogenetic' processes that give rise to 'things' that is important, rather than a reliance on properties or resemblance that are typically used to 'classify'. This removal of 'classification' is not only about, say, animals but also things like cells and atoms. Deleuze's use the term 'multiplicity' to make this shift, with the specific aim of leaving fully behind the idea of 'essence' (rather than shifting to a classification of 'process' rather than 'thing').

> "multiplicities specify *the structure of the space of possibilities"* (De Landa 2005:10)

De Landa goes on to address this concept of 'multiplicity', with a technical description of differential geometry, group theory and dynamical systems theory. He links the concept of multiplicity to the mathematical concept of the 'manifold' as a means of providing a suitable description of 'space' that goes beyond the geometric. The manifold is theoretical model which can be used to simulate a system and the ways in which it can change and is part of differential geometry' (rather than 'analytical geometry'). Differential geometry does not rely on the 'thing' being placed 'within a coordinate system', rather the 'thing' becomes a space in itself. Differential geometry is about relations - so much so that it is 'reflexive', creating its own field of measure, rather than relying on an 'external' framework and the manifold is used to develop an approach to 'n-dimensional' spaces. Deleuze's concept of 'multiplicity' is similar to the manifold in that it can address problems at an 'n-dimensional' level, and it doesn't require an 'external' (n+1) framework to be constructed around it in order to analyse it.

De Landa gives an example of a swinging pendulum that has two degrees of freedom - position and momentum - and these can be modelled within a manifold to describe all possible configurations of the pendulum (the system). What is interesting here is that, even within this simple example, there is already the question of 'system definition' – the swinging of a pendulum is one aspect, but, as De Landa goes on to mention, the pendulum could also melt under high temperature – a 'possibility' that is not mapped within the manifold. Further to this, De Landa acknowledges a trade between the 'actual' and modelled complexity of the manifold.

> “As with any model, there is a trade-off here; we exchange the complexity of the object's change of state for the complexity of the modelling space. In other words, an objects instantaneous state, no matter how complex, becomes a single point, a great simplification, but the space in which the object is embedded becomes more complex.” (De Landa 2005:14)

Here, De Landa expands the scientific terminology to include 'the state space' - a single point in the manifold that describes a condition of the system at a given moment. Changes of state are 'observed' through the tracking, the trajectory of this point through the manifold. Trajectories that begin in different points, but 'tend' towards the same point within the manifold describe topological features of the system – *singularities.* These singularities (called 'attractors' in chaos theory), describe long term tendencies within a system (which may also contain multiple attractors). These singularities – topological points – 'guide' (i.e. 'attract') the trajectory of a given point on the system. During this process, our 'point' may adopt changing physical forms. This is crucial – the topological description is not at the 'essential' level, or even the 'systemic' but at the level of process (specifically *morphogenesis*) – and the relationship(s) between the 'state space' and the 'attractors'.

This is called 'progressive differentiation' – De Landa uses an example (as a metaphor) of an egg, undergoing changes in physical form as it goes from 'topological' egg to fully differentiated individual' - and consists of 'symmetry breaking transitions.' De Landa illustrates this with a description (developed by mathematician Felix Klein) of how euclidean geometry can be 'derived' from 'topological geometry' (via other geometric systems) and goes on to describe how this approach is used in physics to derive the four basic forces (gravity, electromagnetism, strong and weak nuclear forces) from a single 'topological' thought to exist under the extreme conditions at the start of the universe.

> "Using these new concepts we can define the sense in which the metric space we in habit emerges from a non-metric continuum through a cascade of broken symmetries." (De Landa 2005:25)

'Progressive differentiation' raises an interesting question and a challenge to the notion of 'physical laws' as a constant ('essential'). It describes how even the most fundamental aspects of physical science can be re-conceptualized as mutable. Furthermore, the 'phase transitions' (say, liquid to gas) that constitute progressive differentiation expose the fact that 'essential' observations made either side of a transition, can result in them being unrelated and therefore 'broken'.

De Landa explains Deleuze's use of the *virtual,* the *real* and the *actual* in the context of the manifold. The actual is a combination of the real (as in a 'realized' state – a trajectory in the manifold) and the virtual is the *structured* space of possibilities within the manifold. The structured aspect is important here – it is not that virtual is simply an array of other possibilities – for Deleuze, you cannot simply jump between the two, as they each exist due to a process of 'individuation' (or 'progressive differentiation') and therefore there cannot be this notion of 'resemblance' between the actual and the virtual. Instead, there is a process of 'individuation' that corresponds with a 'multiplicity'.

> "...similarity and identity are contingent on the process of individuation..." (De Landa 2005:39)

One of the outcomes of this undermining of 'essences' is the disintegration of pre-existing archetypes and therefore any notion of 'normal' (and by association deviations from normal. This is a flat, non-hierarchical ontology.

De Landa goes on to make further moves against the 'essentialist' position, again turning to scientific terminology, looking specifically at 'intensive' properties and 'extensive' qualities (definitions taken from the field of thermodynamics). Extensive qualities – say, mass, or length – and can be understood as 'divisible' (if you cut it a thing' of 'X' length in half, its the value of its extensive property of length is also halved). Intensive properties, on the other hand – say temperature – are unchanged by such an operation. Furthermore, 'extensive' properties can be *summed*, whereas 'intensive' properties *average (or balanced)* - (in adding a volume of cold water to a volume of hot water, the volume is summed, but the temperature is balanced). De Landa states that 'extensive' qualities are those with a 'metric' (rather than exclusively geometric), whereas 'intensive' are topological (in process terms, and not necessarily 'spatial') and 'non-metric' and that presence (and maintenance) of these intensive properties, are what drives the symmetry breaking events – or 'phase transitions' (think of temperature of water determining a phase transition from liquid to gas). It is the *differences of intensity that drive '*progressive differentiation', and is a key element, De Landa states, of Deleuze's ontology.

Moving beyond properties (intensive and extensive), De Landa goes on to describe other aspects of this ontology that influence a theory of assemblage, that replaces the 'essentialist' hierarchies – a 'things' capacity to *affect* and be *affected*. The term capacity can be understood in a similar way to James Gibson's term 'affordance' (1977). The capacities of a 'thing' cannot be known, as they rely on an interaction with an, as yet, unknown other 'thing'. De Landa describes the 'openness' of 'things' to be key to approximating the extent of its capacities, and this quality of 'openness' is an aspect of it's topological state – it's individuation. Again, this is not a question of infinite possibilities, but of virtual states within a *structured* space of possibilities and, therefore, progressive differentiation gives rise to varying capacities for assembly.

Situating simulation

De Landa's is a complex and detailed proposal, blending contemporary philosophy and science in order to make a strident case for a new kind of 'realist' position that does not rely on 'essential-ism' but focuses on dynamic processes. De Landa's explanations of Deleuze's ontology contain a direct proposal for 'simulation' and particularly emergent models as a method for exploration of its epistemological consequences. Although hinted at, in terms of the 'complexity swapping' nature of such models, De Landa does not directly raise any ethical issues of designing such models and simulations.

Donna Haraway, however, does.

In her book Simians, Cyborgs and Women (1991), Haraway traces a history of scientific research into primate behaviour, and social order in order to describe how certain political ideas are first inserted into, and then validated by the mechanics of, the (supposedly objective) 'scientific method'. One of the conclusions that these scientists draw from their early experiments is that 'dominance' and in particular the idea of male dominance are central to 'social order'. Haraway claims that this is not the result of 'objective observation' during the experiments, but is instead due to the existence of 'domination' as socio-political principle. This then enters scientific process at its outset, thereby creating a kind of 'pre-determination' of causality;

> "The political principle of domination has been transformed here into the legitimating scientific principle of dominance as a natural property with a physical-chemical base." (Haraway 1991:19)

Furthermore, what is clear to Haraway is that, for the scientists, the primate colonies that are the subject of their study represent a 'simulation' of human society. The problem of simulation, in the construction of scientific knowledge, is that it risks taking a simplified abstraction of the complexity of the world and returning it as it's underlying composition.

There is a further difficulty in dealing with simulation, that rests not with simulations as an experimental method, but that is related to access to the tools of simulation. In Haraway's 'study of simian study', many of the experiments take place in an 'artificially' constructed primate colony. On a private island. Clearly that is beyond the means of an 'independent researcher' and requires significant institutional funding. Contemporary simulations are more commonly undertaken using computational technology but the same problem occurs.

Haraway sets out the importance of 'science' - 'the game we have to play' - and suggests that it is, in social constructionist terms, the mechanics and terminology of the' scientific discipline' and the search of 'objectivity' (method, epistemology etc.) that have become a way of closing 'the world' and preventing us from viewing it from our own perspective. Haraway goes on to state that it becomes a 'matter of rhetoric;

> "...the persuasion of the relevant social actors that one's manufactured knowledge is a route to a desired form of very objective power." (Haraway 1991:184)

For Haraway - "feminist objectivity means situated knowledge" (1991:188). She uses a notion of 'vision' as the 'sensory system' from which this 'feminist objectivity will be formed. Although taken initially as a metaphor, it is used as a literal challenge to the visualization methods of modern science that Haraway claims are central to the way that objectivity has previously been constructed. Haraway is proposing to change systems of knowledge through (multiple) ways of 'seeing' – or what she calls 'mobile positioning'.

Haraway again charts the transformations that this epistemological shift creates - for example;

> *universal rationality becoming ethnophilosophies*
>
> *world system becoming local knowledge*
>
> *master theory becoming webbed accounts*

This is an acknowledgement of the problem of objectivity (rather than simply an attack on sciences current attempts at creating it). Haraway is not proposing to dis-engage from the pursuit, but instead is suggesting a notion of objectivity that comprises multiple subjectivities - by implication mobile, continuous and open.

Section 2. Traces of Codes

Fragile Code[1]

Code makes software, which then instructs hardware. But code is messy, it is buggy and sometimes it doesn't work and sometimes it causes crashes. Is it still code if it doesn't work as intended? Or doesn't work at all?

Code is a membrane between human and machine. It is 'readable' by both and yet it is not 'understood' by either. The machine is only capable of executing the code, and the human is exposed to the kinds of fragilities just described. So what is the nature of code?

Mackenzie (2006), along with the likes of Fuller (2003), Ullman (2013), Kitchin and Dodge (2011), is influential in the formation of the nascent discipline of Software Studies. He identifies, among four ontological categories of software whose boundaries are blurred, code as *'fragile' material.*[2] Always on the 'verge of disappearance' or collapse- it *barely* works. It is this fragile quality of code, Mackenzie claims, that allows it to become invisible in our everyday lives. It does not exist in isolation and so without nurture and care it will begin to deteriorate and break down - functionality is lost as the 'ecosystem' which it inhabits changes and it becomes less and less intelligible as the code loses contact with those who created and understood its unique idiosyncrasies. Without attention code rots. (Raymond 2001; Ullman 2013).

Ellen Ullman (2013) writes about her experiences as someone who writes code - an 'originator'. She talks about the discursive ways in which code is written and software is created, the diversity of self taught languages she is required to use as well as the tension and nervousness that all of this inspires. Her ethnography is about the nature of code, as well as those who code, and reveals not only the fragility, but also the mess of this process;

> "It has occurred to me that if people really knew how software got written I'm not sure they'd give their money to a bank, or get on an airplane again." (Ullman 2013:2)

1

By the way, I can code.

If you were to ask me, that is what I will tell you. But I cannot actually prove it. I don't have a certificate that proves I completed a course where I learned to code.

Perhaps I could give you a piece of code that I have written. But how do you know I wrote it? Can someone verify it, identify my writing style? I am not sure that I can.

And how would you know that it was all my own work? What if I have used parts of someone else's code in my own? And what if the part of the code that I have taken, whilst not significant in my own, represents the majority of the other persons? My code doesn't really have references or footnotes.

What if we sat down and you watched me code something, from scratch? But maybe I am just regurgitating something that I have remembered? How would you know? Maybe you could come up with something for me to code and then see if I can do it?

That could be tricky though - I never said I could code everything. For example, I can write a genetic algorithm, or an artificial neural network. But to be honest, I am not that good at web design, so it would be best if you didn't ask me about that.

One of the programming languages I know doesn't really exist any more. Sure, there are some survivors still clinging onto it. And you can hear echoes of it in other places - familiar words, phrases. But it is basically dead. I still have the code that I wrote, but I can't show you it. Well, I mean, I can show you the code, but I can't run it, it doesn't do anything.

But I can code. Honestly.

2

The other elements of his software ontology are originators (those who produce it), recipients (people or machines who 'read' and execute code) and prototypes (software as something else, something physical).

Stabilisations of code....

Code becomes algorithm, algorithm becomes software, and, through software, the virtual becomes material.

Code extends and creates agency. It displays not just secondary agency, where, it extends the agency of its originator, or operator, but true agency (Kitchin and Dodge 2011; Mackenzie 2006). It makes the virtual real while moving "...inside and outside the computer." (Mackenzie 2006:39)

This is a process of 'stabilisation' (Kitchin and Dodge 2011; Mackenzie 2006). Code lacks materiality, but this transition starts to give it form. The fragility of code becomes more and more robust and also rigid.

...As Algorithms

Algorithms can be described as procedural steps towards a given task and are crucial to both software engineering and computer science, but in each case, they operate in different ways (Goffey 2008). In computer science, they are seen in abstract, as a description of the tasks to be carried out. Therefore they can be 'transported' outside of their context and still 'function'. In software design the definition of an algorithm is altogether more practical – algorithms make things happen. Through a specific programming language, acting as an interlayer between human and machine, the algorithm is a set of instructions within a set context.

This difference of attitude towards algorithms, while subtle, is significant in understanding what algorithms are. Goffey states that any attempt to abstract algorithms from their context is futile. In both computer science and software design examples the algorithm acts upon 'something' – its context is its data. And while there is a formal separation between the two, to cleave them apart completely leaves the algorithm meaningless and unusable. It is useful here to clarify what I mean by 'data as context'. It is not that any particular set of data itself represents context, it is the nature of the data that defines context. Algorithms, in their attempt to be 'transferable', are designed to anticipate a particular context.

"what is an algorithm if not the conceptual embodiment of instrumental rationality?" (Goffey 2008:19)

In order to be transportable the 'code' of the algorithm is 'stabilised' and this process forms a stable content in which the algorithm can operate. Goffey states that it is not possible to abstract algorithms and 'think algorithmically' (2008).

…As Computer

Code not only instructs hardware, it also makes it. The relationship between software and hardware is dependent and as such neither can easily be separated from the other – the computer is both hardware and software. The computer is formed by the stabilised code which then forms new codes, which require new hardware. This 'new' hardware is typically manifested through increased computing power (although some technological developments result in 'phase changes' that produce new hardware paradigms, such as the mobile phone). Computing speed, is delivered through the processor, the materialization of which is something discussed by Gabrys (2010). But it is the 'power' itself that I wish to discuss here. Computing 'power' is at the heart of the difference between 'consumer' computing and 'elite' computing. You and I are allowed a certain type of device where the power is 'discrete' (I bought my own processor and I am the only one using it) and the memory is shared (a version of this document is currently sitting on a server somewhere in the USA). Models of commercial computing invert this. Distributed computing allows the aggregation of computer power, while computer memory, containing commercially sensitive data, is discrete. Direct links between control of computing power and social control have been made and the situation has been likened to the control of labour in the Marxist sense – congealed computer power rather than labour (Eglash 2008).

Before this current paradigm, in the 'early days' of computing, the relationship between code and its hardware was emerging. Early models of computing were concerned with mechanical automation of otherwise manual processes for example counting, calculating, but also weaving (Slack 2008). The 'Jacquard loom', which automates the process of manufacturing complex textiles, and the 'Difference Engine', a large mechanical calculators, were machines that attempted (with varying degrees of success) to replace human labour in these activities. They can be understood as crude, early computers but they both also represent embryonic prototypes for code and software.

> "...before computers were machines, they were people, usually women..." (Slack 2008:186)

The people Slack is referring to were compiling statistical tables and data cards for use in calculating things like the trajectory or position of Halley's comet. The act of inscribing the data can be seen as a kind of coding for a wholly human process. Slack describes this activity as bureaucratic and he makes a direct link between this and Alan Turing's innovation in creating the 'ticker tape' paper input for early computers. So, a narrative of the origins of computer hardware is entwined with a story of stabilised code.

Slack goes further to claim, citing Sherry Turkle, that not only did the input mechanism for early computers create the need for what we would now call 'computer memory', but that this in turn created a model for understanding 'human memory' and a language by which the field of psychology was able to address it. This 'tight coupling' creates a circular logic - computer memory is seen to be a good model for understanding human memory because the theories human memory is itself modelled on computer memory. It is an extreme, twisted skeumorphism from which the 'cyborg' emerges (Slack 2008).

...As Space

> "Software thus abstracts the world into defined, stable ontologies of capta and sequences of commands that define relations between capta and details how that capta should be processed." (Kitchin and Dodge 2011:p27)

Models of climate change are used by Kitchin and Dodge to illustrate how scientific theories and empirical observations are first formalised into a mathematical model, then translated to code capable of simulating (or, as they say creates narratives about) the earth's future climate behaviour (2011). They make the point that such models have come to underpin the global policy discussions surrounding climate change. The way in which these computational models have such agency is bound to the process of creating the model itself.

Climate change models and simulations represent arguably the greatest collaborative project ever undertaken, with researches sharing data and resources whilst working towards a single, shared goal, in an unprecedented way. This, in a way, is a 'self-validation' of the usefulness of the outcome - if all of the 'scientific community' is behind this project then it must be valuable. Its very scale means that it cannot reasonably be questioned (leaving to one side those who simply do not 'believe' in man made climate change').

If the models are tools for understanding the problems facing the planet due to man-made climate change, they are certainly not interested in offering the solution to them. As Kitchin and Dodge note, the simulations have traction in the policy debate because they are successful at visualising the data - the planet makes an attractive spatial field on to which to project data.

> "The models analyse the world, the world responds to the models." (Kitchin and Dodge 2011:30)

In spite of the colossal shared effort in creating them and the obvious success in communicating the findings, the principle global policy debate around the issue of climate change has centred around what constitutes an acceptable rise in global temperature. The complexity of the model is reduced to a single point in the manifold. The scientific model, simulation changing climate, makes no effort to engage with the political dimensions of either the problem or the solution. It is not a forensic model, it is not able to directly analyse the consequences of climate change at anything other than the macro scale. It is not concerned with the micro level and is not, for example, capable identifying responsibility for more specific impacts of climate change than overall, global temperature rises. This is an example of how code is both stabilised and stabilising – as the science is 'translated' into code, and the code begins to tell a story that is returned as facts.

Kitchin and Dodge (2011) create an ontology around the material formations of code. They are concerned with the symbiotic relationships that exist between code and the 'things' in which it operates;

> **coded objects** require code to perform

> **coded infrastructures** link coded objects
>
> **coded processes** flows across coded infrastructures
>
> **coded assemblages** convergence of several different coded infrastructures

They use the term 'code/ space' to describe space that is enacted through code, giving an example of the security area in an airport terminal. Not only would the space not perform as designed without the software it would also have a different physical form – it is 'designed' to operate with its code.

> "Code/ Space occurs when software and the spatiality of everyday life becomes mutually constituted, that is, produced by one another." (Kitchin and Dodge 2011:16)

The relationship between code and space is therefore not 'deterministic', rather it is symbiotic. They make a clear distinction between 'coded space' and 'code/ space', defining coded spaces as spaces where the presence of code makes a difference but its relationship with space is not 'fundamental'. One example they give of 'coded space' is the use of power point to present to an audience in an auditorium.

Their ontology is only concerned with 'actual' spaces, rather than spaces that could not have been designed without code. A simple example of a code/space would be any building with a reasonably sophisticated digital environmental control mechanism. But there was a process involving code before the building was constructed, and the feedback loops that regulate the day to day performance of the building also operate at a virtual level. Certain aspects of the 'material' design are directly influenced by code. The options for the user to open windows, for example, are linked to the digital technology, it is the use of code in the 'virtual space' (the design simulation) that begins the process of transduction. In this way, the notion of 'code/ space' can be extended beyond the 'performance' of the space, to the process though which it is designed in the first place. This extension of the principles of code/space to include 'simulated space' – a kind of 'virtual code/ space' - is important in developing a strand of software studies that can be applied to spatial design.

Destabilisations of code - hacking

In their book *Abstract Hacktivism - The making of hacker culture*, Otto Von Busch and Karl Palmas (2006) attempt to join the politics of emerging computer technologies and cultures of activism. They do this firstly from the point of view of 'openness' and 'open source', as a consequence of the pervasiveness of computing technology, which many still view as benign or at least a-political, and secondly, in terms of describing contemporary forms of 'activism', and revisiting the activist practices post-1968. They seek to move away from the original use of the term *hacktivist,* coined in 1995 by Jason Sack, as description of a kind of 'resistance' through direct, digital (online) action. Instead, Von Busch and Palmas are concerned with creation rather than destruction - *hacking* rather rather *cracking* in the terms of Eric Raymond.

> "To make a successful hack the hacker needs to keep the power on, keep the flows through the system intact, to keep it functioning as a tool, but reclaiming it, submit it under his will by taming and modulating the flows through the system" (Busch and Palms 2006:60)

This is not done by looking at the use of computers themselves, but the ways in which the mechanics of computing are enacted in abstract, non-computational contexts. This is framed, with reference to Michel Serres, by the symbiotic processes from which conceptual models of knowledge are formed from abstracted concepts of particular technologies;

> "Are the 'traditional' conceptual models of modern thought, inspired by the abstract mechanisms of motors, being replaced by new conceptual models, based on the abstract methods of computers?" (Busch and Palms 2006:23)

It is an attempt to construct a *Hacking Methodology* in which they see hacking as not 'oppositional' but 'dialogical'. And looking at alternative hacking practices – including shop dropping, fan fiction, knitting, they conclude it is more than merely customisation, which suggests only a slight deviation of purpose to suit an individuals requirements. Each of these examples contain some aspect of reconfiguration rather than modification. In the case of fan fiction, elements of an original text may be kept but the context and narrative may be completely re-imagined and characters are retained often in name only, their *characteristics* are altered significantly.

Within all of this, there is a problem of 'abstracting' the practice of hactivisim (and similarly algorithms and simulations). Can hacking, understood solely as a computational activity, be abstracted to create a method applicable in other contexts, or does it become 'merely' a metaphor for a way of working that 'resembles' an unrealisable computational method? Palmas comments that;

> "It is easier to get away with claiming that an artist, activist or author is inspired by computers, rather than claiming that her or his model is actualising the same diagram that is actualised by new media technological contraptions." (Busch and Palms 2006:112)

This then becomes an ontological question. For Deleuze, as explained by Manuel De Landa (2005), the problem of metaphor is linked to 'essentialism'. For Deleuze, 'metaphor' is an unhelpful approach in this process of removing 'essences', and it is of no relevance when that process is complete. Therefore, the removal of metaphor is both a way of working towards, and a result of, the construction of his ontological position. The reticence in Palmas' comment regarding the plausibility of the claims of the artist to be 'actualising the same diagram' is rooted in the nature of 'hacking' itself. He describes hacking as something that existed before computers;

> "....it just so happened that this abstract machine was actualized within, and popularized in the social context of, new media." (Busch and Palms 2006:112)

Hacking could have been about another form of media, in another time. So, rather than choosing a position, his conclusion is that 'hacking' exists as both a 'computer inspired', and as a computer-like model. However, I see this more as a question of tendency – 'tending' toward a computational method that has been successfully abstracted from metaphor, but never quite reaching it (as singularities, or attractors, within a manifold).

Section 2. Virtual INTERFACES

Other ideas of interface

The pervasive nature of contemporary digital technology – coded products, spaces and infrastructures (Kitchin and Dodge 2011) – is such that the question of 'interface' has moved beyond the traditional model of HCI (Human Computer Interface) (Fuller 2003). Instead of sitting outside of the system looking across a screen, we constantly re-make our relationships with computational technology and in the process we are creating dynamic, topological, post-human formations (Haraway 1991; Braidotti 2013). At the same time, there is an apparent 'de-materlialisation' of technology from *our* everyday lives. This is a process driven by the ubiquity of computing which creates environments so rich in this technology that we no longer make the distinctions between the digital and the non-digital (Shepard 2011). Our true interface with the materials of digital technology is only revealed through the waste left behind after its manufacture (Gabrys 2010).

This is the new paradigm for our 'interface' with digital technology. What are the implications of these issues with regard to new types of computational tools for spatial design?

Cyborg formations: Becoming post-human

> "The cyborg is our ontology; it gives us our politics" (Haraway 1991:150)

'A cyborg Manifesto' is Donna Haraway's attempt to first describe and then claim a possible 'cyborg future', identifying the cyborg – a cybernetic organism, a hybrid of the 'organism and machine' - as both a reality and a fiction (Haraway 1991). Haraway states that cyborgs 'exist' in science fiction, in modern medicine, in military hardware, in ideas of reproduction (or 'replication') and they are always coupling between the imaginary and the reality. The manifesto is a call to take ownership of and revel in the confusion created by these new ontological boundaries. For Haraway, the cyborg has no 'origin story' in a western sense and it is a creature in a 'post-gender world'. The manifesto, then, is about transgressed boundaries 'fractured identities' and therefore possible 'fusions'. It contests not only what constitutes 'nature' but, by implication, the ontological basis of Western epistemology. Haraway makes three distinctions in a new 'cyborg ontology';

> 1. Human ≠ Animal
>
> 2. Human ≠ Machine OR Animal ≠ Machine
>
> 3. Physical ≠ non-physical

Haraway goes on to sketch out an alternate epistemological position that is the consequence of this 'cyborg ontology', expressing the transitions from previous, hierarchical dominations to the 'scary new networks' which she names 'informatics of domination'. These are presented as a list that opposes 'natural' with non-natural' terminology, for example reproduction becoming replication.

Whilst the manifesto is certainly 'optimistic', it is not blindly utopian - the transitions described above are not taken as *de facto* positive epistemological 'outcomes'. Haraway is concerned with describing the field in which questions of gender, race, identity will be contested, whether we like it or not. This is a call to occupy these fields and these 'informatics' represent possibilities - they are the potential sites of control and exploitation as much as they are potential spaces of more positive re-configurations;

> "The cyborg is a kind disassembled and reassembled, post-modern collective and personal self. This is the self feminists must code" (Haraway 1991:163)

Haraway's cyborg proposition is a radical move and a precursor to the broader notion of 'post-humanism'. Rosie Braidotti describes post-humanism as a"...generative tool to help us re-think the basic unit of reference for the human..." (Braidotti 2013:6). In forming this definition, Braidotti draws direct links between the (earlier) concept of 'autopoeisis', as described by Maturana and Varela and central to 'second wave cybernetics' and Guattari's 'Three Ecologies' (Braidotti 2013).

The original concept of autopoeisis is about a 'biological' system's closed-ness to external 'input' (i.e. any stimulus acting upon a given system must by definition be part of the system, rather than external to it) (Maturana and Varela 1991). This is a significant departure from 'first wave' cybernetic theory, in which systems are subjected to external input and are capable of regulating their function accordingly (Wiener 1961). Whilst autopoeisis, in its original definition, is concerned with 'biological systems', Braidotti adopts Guattari's extension of its meaning to include a 'machinic-autopoesis'. (Guattari, Pindar, and Sutton 2005). As with Haraway's manifesto, this is a description of post-humanism that produces dynamic formations of the subject and is one that implies a process of 'becoming' and is a strong move against attempts to make a separation between not only human and machine, but also mind and body.

The aspiration to make such a separation was popular within cybernetics, with Norbert Wiener proposing that it would possible to 'telegraph' a human being across a communication network, thereby implying that the mind could be successfully dis-embodied. It is also an idea that is still prevalent in science through disciplines such as molecular biology where the human is treated as code (Hayles 1999). Katherine Hayles makes her definition of post-humanism in opposition to the idea that information can move without change through different material substrates;

> "...my dream is a version of the post human that embraces the possibilities of information technologies without being seduced by fantasies of unlimited power and disembodied immortality, that recognizes and celebrates finitude as a condition of human being, and that understands human life is embedded in a material world of great complexity, one on which we depend for our continued survival." (Hayles 1999:5)

Hayles is seeking to re-figure the cybernetic post-human, and sees the contemporary context as an opportunity to explore new possibilities of 'post-human'. She is attempting to claim a version of 'post-human' that is based on a situated reading of the history of cybernetics and that moves beyond more traditional approaches which privilege informational pattern over material instantiation (Hayles 1999).

Virtuallity and digital (im)materialism

Hayles suggests that developments and changes in cybernetic theories can not only be traced through its artefacts, but also reinterpreted, and that the idea of 'artefact' itself can be expanded to include theories;

> "if we want to contest what these technologies signify, we need histories that show the erasures that went into creating the condition of virtuality, we can de-mystify our progress towards virtuality and see it as the result of historically specific negotiations rather than the of the irresistible force of technological determinism." (Hayles 1999:20)

The current ubiquity of computational technology has contributed to a process of apparent 'de-materialization' of the artefacts themselves. One recent news article about The Pirate Bay, an organisation synonymous with providing peer-to-peer technology that has allowed access to illegal downloading of digital content and which has been the subject of numerous legal challenges, described how they were moving their serves 'to the cloud', in order to avoid further legal disputes by effectively de-territorialising their hardware (BBC News - The Pirate Bay Moves to the Cloud to Avoid Shutdown 2012). Of course, the servers are till somewhere – they are not in an actual cloud. This anecdote is indicative of a wider perception of computational technology as 'virtual'. The material consequences of digital technology are rarely considered within this mood of hardcore 'future-ism'.

Jennifer Gabrys employs a 'natural history' as way to explore and to describe the alternative digital materialities of digital artefacts concerned with waste (2010). In order to create this idea of 'Digital Rubbish' she adopts, as a methodological starting point, Walter Benjamin's own use of 'fossils as the sites of study'. As Gabrys explains, this allows the investigation to be freed from the conventional approaches often associated with technology, such as 'progress' and 'invention' and instead focus on the "...the ways in which electronic technologies fail and decay." (Gabrys 2010:7).

This 'natural history' is built by disassembling the computer into component parts and taking them as site for study. 'The processor' is the seen as result of precise chemical transformations during manufacture, 'the screen' is that of the NASDAQ stock exchange in New York and the plastic case in understood through the waste disposal site in Guangdong, China on which it ends up. Gabrys sees the fossil (and by extension, the fossilised site) not as the (conventional) physical artefact, but instead the intersection between all of the processes by which electronic waste is created.

> “[they are]…. temporal zones that register the speed and volume of production, consumption, and disposal of digital technologies." (Gabrys 2010:14)

Rather than seeing them as 'de-materialized technologies' Gabrys is exploring the consequences of our current condition of “technological transience” (Gabrys 2010) through a process of materialisation through waste.

From the 'fossilized' site of the chip we see unfold, at the chemical scale, the complex interactions and transformations necessary in the production of processor chip (Gabrys 2010). This process is described at the chemical scale of the chips themselves, at the bodily scale for the workers involved and at the global scale of the land contamination from by-products of the process. What is described is the materialisation of the 'Moore's Law' (that the number of transistors on integrated circuits doubles approximately every two years, as stated in 1965 by Gordon Moore, co founder of Intel). In the context of the material processes associated with the mass-scale production of computer chips, this constitutes less a 'Law' or even a forecast and more of an industry growth strategy. It is certainly a long way from the commonly held, almost 'mystical', view that there is some, as yet not fully understood, underlying aspect of computational technology that is driving its development in line with this prediction.

> "New economies, together with new arrangements of labour, altered material and chemical inputs, and spatial distributions, help to create the very conditions through which a technology can take hold, persist, and even become seemingly natural." (Gabrys 2010:31)

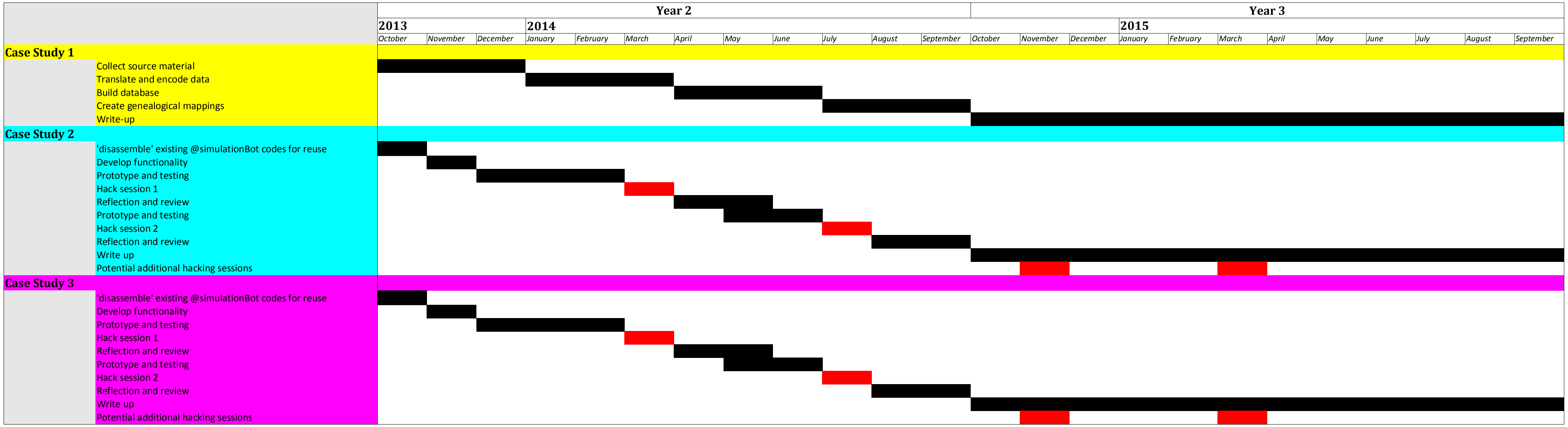
The manufacturing processes described here are far removed from the more broadly understood idea of 'manufacturing'. Instead, abundant silicon is transformed in a process Gabrys describes as 'material alchemy'. It is hard to understand the making of a chip as an assembly in any conventional sense and its elements are perhaps more easily understood as 'ingredients'. This is caused by the size of the artefact, as well as the fact of its apparent 'static' operation with no 'moving parts' visible to the naked eye. Making computer chips represents a *new* form of manufacturing, creating a *new* scale of product, that performed a *new* type of function and the opaqueness of each of these aspects contributes to the problem of a digital technology being widely viewed as 'de-materialized'.

> "Yet the term de-materialised does not necessarily mean “without material” but may, instead, refer to modes of materialization that render infrastructures imperceptible or ephemeral." (Gabrys 2010:51)

Gabrys uses the NASDAQ exchange in New York to explore interface. The building is composed of digital screens that display a distributed network of computed transactions'. The screens on the façade display images to the street while the interior screens, acting as backdrop for 24hr new reports, project to the world. The screens display only an abstracted graphic of the trading activity, rather than the true material fluctuations in commodity ownership or value. Gabrys describes this as our de-materialised and depoliticised “technopresent" (Gabrys 2010:57) and contrasts this mode with the waste sites of the discarded computer screens. She makes reference to media theorist Lisa Parks who describes consumer societies in which the interface for the privileged is the exterior surface of the computer (the screen, the keyboard) whereas for the rest, whose job it has become is to scavenge and recycle the discarded artefacts, it is the inside of the machine that has become their interface.

4. Timetable for Future Work

Timetable



Year 2
Year 3
2013
2014
2015
October
November
December
January
February
March
April
May
June
July
August
September
October
November
December
January
February
March
April
May
June
July
August
September
Case Study 1
Collect source material
Translate and encode data
Build database
Create genealogical mappings
Write-up
Case Study 2
'disassemble' existing @simulationBot codes for reuse
Develop functionality
Prototype and testing
Hack session 1
Reflection and review
Prototype and testing
Hack session 2
Reflection and review
Write up
Potential additional hacking sessions
Case Study 3
'disassemble' existing @simulationBot codes for reuse
Develop functionality
Prototype and testing
Hack session 1
Reflection and review
Prototype and testing
Hack session 2
Reflection and review
Write up
Potential additional hacking sessions

Bibliography

Methodology

Open Source

Simulations

Codes

Interfaces

Adkins, L., & Lury, C. (2012). *Measure and Value*. Wiley-Blackwell.

Alan Irwin. (1995). *Citizen Science: A Study of People, Expertise, and Sustainable Development*. Burns & Oates.

Alexander, C. (1964). *Notes on the Synthesis of Form.* Harvard University Press.

Alexander, C. (1978). *A Pattern Language: Towns, Buildings, Construction*. OUP USA.

Anderson, C. (2013). *Makers: the new industrial revolution*.

Award-winning game 140 tests player's rhythmic ability (Wired UK). (n.d.). *Wired UK*. Retrieved April 4, 2013, from http://www.wired.co.uk/news/archive/2013-04/2/140-game

Barad, K. (2007). *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*. Duke University Press.

Bateson, G. (1972). *Steps to an ecology of mind*. New York: Ballantyne Books.

Bateson, G. (1980). *Mind and nature : a necessary unity*. London: Fontana.

BBC News - The Pirate Bay moves to the cloud to avoid shutdown. (2012, October 17). Retrieved September 29, 2013, from http://www.bbc.co.uk/news/technology-19982440

Beane, S. R., Davoudi, Z., & Savage, M. J. (2012). Constraints on the Universe as a Numerical Simulation. *arXiv:1210.1847*. Retrieved from http://arxiv.org/abs/1210.1847

BeetBox uses Raspberry Pi to turn veggies into drumkit (Wired UK). (n.d.). *Wired UK*. Retrieved December 20, 2012, from http://www.wired.co.uk/news/archive/2012-12/20/beetbox-vegetable-drumkit

Berry, D. M. (n.d.). FCJ-086 A Contribution Towards A Grammar of Code. Retrieved September 13, 2013, from http://thirteen.fibreculturejournal.org/fcj-086-a-contribution-towards-a-grammar-of-code/

Big Ideas: Put digital displays on contact lenses (Wired UK). (n.d.). *Wired UK*. Retrieved April 4, 2013, from http://www.wired.co.uk/magazine/archive/2013/03/big-ideas/put-digital-displays-on-contact-lenses

Blu Dot Real Good Experiment. (2009). Retrieved from http://vimeo.com/8201309

Braidotti, R. (2006). *Transpositions : on nomadic ethics*. Cambridge: Polity.

Braidotti, R. (2013). *The Posthuman*. Polity Press.

Busch, O. V., & Palms, K. (2006). *Abstract Hacktivism: The Making of a Hacker Culture*. Skyscraper Digital Pub Limited.

CarbonBuzz. (n.d.). Retrieved July 24, 2013, from http://www.carbonbuzz.org/index.jsp

Codev2:Lawrence Lessig. (n.d.). Retrieved July 25, 2013, from http://codev2.cc/

Cosmic rays offer clue our universe could be a computer simulation (Wired UK). (n.d.). *Wired UK*. Retrieved October 13, 2012, from http://www.wired.co.uk/news/archive/2012-10/11/universe-computer-simulation

Create personal weather reports with an auto-tweeting weather meter (Wired UK). (n.d.). *Wired UK*. Retrieved October 31, 2012, from http://www.wired.co.uk/news/archive/2012-10/30/monitoring-hurricanes

CTheory.net. (n.d.). Retrieved November 6, 2012, from http://www.ctheory.net/articles.aspx?id=74

Dawkins: Science, "It Works, Bitches!" (2013). Retrieved from http://www.youtube.com/watch?v=Wch4ekOEwaA&feature=youtube_gdata_player

DeLanda, M. (2006). *A New Philosophy of Society: Assemblage Theory and Social Complexity*. Continuum.

DeLanda, M. (2005). *Intensive Science and Virtual Philosophy*. Continuum International Pub Group.

DeLanda, M. (2011). *Philosophy and Simulation: The Emergence of Synthetic Reason*. Continuum.

Deleuze, G. (2004). *A thousand plateaus : capitalism and schizophrenia*. London: Continuum.

Designer publishes collection of developer-written "code poems" (Wired UK). (n.d.). *Wired UK*. Retrieved October 13, 2012, from http://www.wired.co.uk/news/archive/2012-10/12/code-poems-book

Deus Ex and the future of freedom. (2011, August 24). *the Guardian*. Retrieved October 31, 2012, from http://www.guardian.co.uk/technology/gamesblog/2011/aug/24/deus-ex-future-of-freedom

Dick, P. K. (1999). *A Scanner Darkly* (New Ed.). Gollancz.

Dick, P. K. (2007). *Do Androids Dream Of Electric Sheep?* (New Ed.). Gollancz.

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*(3), 431 – 445. doi:10.1068/b32041t

Edifice Maps. (n.d.). Retrieved July 31, 2013, from http://edifice.opencityapps.org/

Eglash, R. (2008). Computing Power. In M. Fuller (Ed.), *Software Studies: A Lexicon*. The MIT Press.

Electronic brain hacks are turning insects into robotic helpers (Wired UK). (n.d.). *Wired UK*. Retrieved December 4, 2012, from http://www.wired.co.uk/news/archive/2012-12/04/insect-mind-control

Emily Monosson – Robot evolution. (n.d.). *Aeon Magazine*. Retrieved June 12, 2013, from http://www.aeonmagazine.com/nature-and-cosmos/emily-monosson-robot-evolution/

Engeli, M. (2001). *Bits and Spaces: CAAD for Physiscal, Virtual and Hybrid Architecture at ETH Zurich* (1st ed.). Birkhäuser.

"E-puck" robot swarm moves objects with minimum of information (Wired UK). (n.d.). *Wired UK*. Retrieved April 4, 2013, from http://www.wired.co.uk/news/archive/2013-04/1/cooperative-robot-swarm

Paulos, E. (n.d.). Retrieved July 25, 2013, from http://www.paulos.net/

Fuller, M. (2003). *Behind the blip: essays on the culture of software*.

Fuller, M. (2007). *Media Ecologies: Materialist Energies in Art and Technoculture*. MIT Press (MA).

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

Gabrys, J. (2010). *Digital Rubbish: A Natural History of Electronics*. The University of Michigan Press.

Gibson, J. (1977). The Theory of Affordances. In R. Shaw & J. Bransford (Eds.), *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*. Lawrence Erlbaum Associates.

Gibson, W. (1993). *Neuromancer*. London: Harper Collins Science Fiction & Fantasy.

Gibson, W., & Sterling, B. (1996). *The Difference Engine*.

GitHub Data | NetWords. (n.d.). Retrieved May 24, 2013, from http://labs.sapo.pt/networds/2012/05/09/github-data/

Github explorer. (n.d.). Retrieved May 24, 2013, from http://lumberjaph.net/graph/2010/03/25/github-explorer.html

Goffey, A. (2008). Algorithm. In M. Fuller (Ed.), *Software Studies: A Lexicon*. The MIT Press.

the great game, raouf haj yihya in liverpool. (n.d.). *notes from a fruit store*. Retrieved April 4, 2013, from http://notesfromafruitstore.net/2010/09/18/the-great-game-raouf-haj-yihya-in-liverpool/

Guattari, F. (1995). *Chaosmosis: an ethico-aesthetic paradigm*. Sydney: Power Publ.

Guattari, F., Pindar, I., & Sutton, P. (2005). *The Three Ecologies*. Continuum International Pub Group.

Hacking, I. (1983). *Representing and intervening : introductory topics in the philosophy of natural science*. Cambridge: Cambridge University Press.

Hacking, I. (1999). *The social construction of what?* Cambridge, Mass; London: Harvard University Press.

Hanna, S., Hesselgren, L., Gonzalez, V., & Vargas, I. (2010). Beyond simulation (p. 1). ACM Press. doi:10.1145/1878537.1878727

Haraway, Donna J. (2003). *The Companion Species Manifesto: Dogs, People and Significant Otherness* (2nd ed.). University of Chicago Press.

Haraway, Donna J. (2007). *When Species Meet*. University of Minnesota Press.

Haraway, Donna Jeanne. (1989). *Primate visions : gender, race, and nature in the world of modern science*. New York ; London: Routledge.

Haraway, Donna Jeanne. (1991). *Simians, cyborgs and women : the reinvention of nature*. London: Free Association Books.

Haraway, Donna Jeanne. (1997). *Modest_Witness@Second_Millennium.FemaleMan_Meets_OncoMouse : femin ism and technoscience*. New York ; London: Routledge.

Harding, S. G. (1991). *Whose science? Whose knowledge? : thinking from women's lives*. Ithaca, NY: Cornell University Press.

Harding, S. G. (1998). *Is science multicultural? : postcolonialisms, feminisms, and epistemologies*. Bloomington, Ind: Indiana University Press.

Harding, S. G. (2008). *Sciences from below : feminisms, postcolonialisms, and modernities*. Durham, NC; London: Duke University Press.

Hatnote Listen to Wikipedia. (n.d.). Retrieved August 21, 2013, from http://listen.hatnote.com/#nowelcomes,en

Hayles, K. (1999). *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature and Informatics* (74th ed.). University of Chicago Press.

Hayles, N. K. (2002). *Writing machines [...] [...]*. Cambridge, Mass [u.a.: MIT Press.

Hayles, N. K. (2005). *My Mother Was a Computer: Digital Subjects and Literary Texts*. University of Chicago Press.

Hayles, N. K. (2012). *How We Think: Digital Media and Contemporary Technogenesis*. University of Chicago Press.

Hillier, B. (2007). *Space is the machine: a configurational theory of architecture*. London, UK: Space Syntax. Retrieved from http://discovery.ucl.ac.uk/3881/

Hillier, B., Leaman, A., Stansall, P., & Bedford, M. (1976). Space syntax. *Environment and Planning B: Planning and Design*, *3*(2), 147–185.

How to make balloon maps (Wired UK). (n.d.). *Wired UK*. Retrieved April 4, 2013, from http://www.wired.co.uk/magazine/archive/2013/03/how-to/how-to-make-balloon-maps

Hyde, R. (2012). *Future Practice: Conversations from the Edge of Architecture* (1st ed.). Routledge.

If Things Can Talk, What Do They Say? If We Can Talk to Things, What Do We Say? | Electronic Book Review. (n.d.). Retrieved July 25, 2013, from http://www.electronicbookreview.com/thread/firstperson/voicechip

Issue #2: Bio/Hardware Hacking » Journal of Peer Production. (n.d.). Retrieved July 24, 2013, from http://peerproduction.net/issues/issue-2/

Istanbul Design Biennial - Adhocracy. (n.d.). Retrieved April 24, 2013, from http://istanbuldesignbiennial.iksv.org/adhocracy/

Jasanoff, S. (2004). *States of Knowledge: The Co-Production of Science and the Social Order*. Routledge.

Johnson, R., Gamma, E., Vlissides, J., & Helm, R. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Kitchin, R., & Dodge, M. (2011). *Code/space software and everyday life*. Cambridge, Mass.: MIT Press. Retrieved from http://site.ebrary.com/id/10479192

Lash, S., & Lury, C. (2007). *Global Culture Industry: The Mediation of Things* (1st ed.). Polity Press.

Latour, B. (1987). *Science in action: how to follow scientists and engineers through society*. Cambridge, Mass.: Harvard University Press.

Latour, B. (1993). *We have never been modern*. London: Harvester Wheatsheaf.

Latour, B. (2004). *Politics of nature : how to bring the sciences into democracy*. Cambridge, Mass: Harvard University Press.

Latour, B., Weibel, P., (Exhibition), M. T. P. A. of D., & Karlsruhe, Z. für K. und M. (2005). *Making things public : atmospheres of democracy*. Cambridge, Mass; London: MIT.

Latour, B., & Woolgar, S. (1986). *Laboratory Life: The Construction of Scientific Facts*. (J. Salk, Ed.) (Reprint.). Princeton University Press.

Law, J. (2004). *After Method: Mess in Social Science Research*. Psychology Press.

Lord of the Files: How GitHub Tamed Free Software (And More) | Wired Enterprise | Wired.com. (n.d.). *Wired Enterprise*. Retrieved March 11, 2013, from http://www.wired.com/wiredenterprise/2012/02/github/

Lury, C., & Wakeford, N. (Eds.). (2012). *Inventive Methods: The Happening of the Social*. Routledge.

Mackenzie, A. (2006). *Cutting code: software and sociality*. New York: Peter Lang.

MacKenzie, D. A., & Wajcman, J. (1999). *The social shaping of technology [...] [...]*. Buckingham [u.a.: Open University Press.

Magical Maths « How Japanese Kids Learn To Multiply – Amazing, No Need to Learn Japanese. (n.d.). *magicalmaths*. Retrieved April 10, 2013, from http://www.magicalmaths.org/how-japanese-kids-learn-to-multiply/

Manifesto for Agile Software Development. (n.d.). Retrieved August 7, 2013, from http://agilemanifesto.org/

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

Marres, N. (2012). *Material Participation: Technology, the Environment and Everyday Publics*. Palgrave Macmillan.

Maturana, H. R., & Varela, F. J. (1991). *Autopoiesis and Cognition: The Realization of the Living* (Softcover reprint of the original 1st ed. 1980.). Springer.

McCullough, M. (2007). New media urbanism: grounding ambient information technology. *Environment and Planning B: Planning and Design*, *34*(3), 383 – 395. doi:10.1068/b32038

Mieville, C. (2011). *The City & The City*. Pan.

Mieville, C. (2012). *Embassytown*. Pan.

Miller, C. A., & Edwards, P. N. (2001). *Changing the Atmosphere: Expert Knowledge and Environmental Governance*. MIT Press.

Mitchell, W. J. (1995). *City of bits : space, place and the infobahn*. Cambridge, Mass; London: MIT Press.

Mitchell, W. J. (2003). *Me[plus plus] : the cyborg self and the networked city*. Cambridge, Mass; London: MIT.

Myers, D. (2010). *Play redux : the form of computer games*. Retrieved from http://hdl.handle.net/2027/spo.7933339.0001.001

Nash, A. (n.d.). FCJ-148 Affect and the Medium of Digital Data. Retrieved September 13, 2013, from http://twentyone.fibreculturejournal.org/fcj-148-affect-and-the-medium-of-digital-data/

Opensource Architecture. (2013, April 20). In *Wikipedia, the free encyclopedia*. Retrieved from http://en.wikipedia.org/w/index.php?title=Opensource_Architecture&oldid=550529455

Páez, S. M., Lama, J. P. D., & Andrade, L. H. (2012). *Wikiplaza: Request for Comments*. Actarbirkhauser.

Pask, G. (1961). *An approach to cybernetics.* London: Hutchinson.

Pask, G. (1975). *The cybernetics of human learning and performance : a guide to theory and research*. London: Hutchinson Educational.

Paulos, E. (n.d.). Retrieved July 25, 2013, from http://www.paulos.net/

Penn, A., & Turner, A. (2001). Space syntax based agent simulation. In *Presented at: 1st International Conference on Pedestrian and Evacuation Dynamics, University of Duisburg, Germany. (2001)*. Presented at the 1st International Conference on Pedestrian and Evacuation Dynamics, University of Duisburg, Germany. Retrieved from http://discovery.ucl.ac.uk/2027/

Penn, Alan. (2003). Space Syntax And Spatial Cognition Or Why the Axial Line? *Environment and Behavior*, *35*(1), 30–65. doi:10.1177/0013916502238864

Pickering, A. (2011). *The Cybernetic Brain: Sketches of Another Future*. University of Chicago Press.

Plant, S. (1998). *Zeros + ones : digital women + the new technoculture*. London: Fourth Estate.

Prigogine, I., & Stengers, I. (1985). *Order out of chaos: man's new dialogue with nature*.

Public Lab: a DIY environmental science community. (n.d.). Retrieved July 31, 2013, from http://publiclab.org/

QUELSOLAAR:COM : Love - Online Procedural Adventiure Game. (n.d.). Retrieved October 31, 2012, from http://www.quelsolaar.com/love/index.html

Raymond, E. S. (2001). *The cathedral & the bazaar: musings on Linux and open source by an accidental revolutionary*. Cambridge, Mass.: O'Reilly.

Resnick, M. (1997). *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. The MIT Press.

Robinson, R., Rittenbruch, M., Foth, M., Filonik, D., & Viller, S. (2012). Street Computing: Towards an Integrated Open Data Application Programming Interface (API) for Cities. *Journal of Urban Technology*, *19*(2), 1–23. doi:10.1080/10630732.2012.698064

Sign at Bar Reads "Google Glass Is Banned On These Premises." (n.d.). *Laughing Squid*. Retrieved July 24, 2013, from http://laughingsquid.com/sign-at-bar-reads-google-glass-is-banned-on-these-premises/

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

Situated Technologies |. (n.d.). Retrieved November 6, 2012, from http://www.situatedtechnologies.net/

Slack, W. (2008). Memory. In M. Fuller (Ed.), *Software Studies: A Lexicon*. The MIT Press.

Spuybroek, L., & Lang, B. (2000). *Weight of the Image: Masterclass NAi 1999*. NAI Publishers.

Stengers, I. (1997). *Power and Invention: Situating Science*. (P. Bains, Trans.). University of Minnesota Press.

Stengers, I. (2000). *The Invention of Modern Science*. (D. W. Smith, Trans.). University of Minnesota Press.

Stengers, I. (2010). *Cosmopolitics I : I. the science wars, II. the invention of mechanics, III. thermodynamics*. Minneapolis, Minn: University of Minnesota Press ; Bristol.

Stengers, I. (2011). *Cosmopolitics II*. Minneapolis, Minn; London: University of Minnesota Press.

Sterling, B. (2005). *Shaping Things*. MIT Press.

Study: brainless robots swarm just like animals (Wired UK). (n.d.). *Wired UK*. Retrieved March 15, 2013, from http://www.wired.co.uk/news/archive/2013-03/15/robot-swarms

Terry, J. (1997). *Processed Lives: Gender and Technology in Everyday Life*. Routledge.

The dumbing-down of programming. (n.d.). Retrieved March 20, 2013, from http://www.salon.com/1998/05/13/feature_320/

The GitHub generation: why we're all in open source now (Wired UK). (n.d.). *Wired UK*. Retrieved March 11, 2013, from http://www.wired.co.uk/news/archive/2013-03/11/github-democracy

The virtual adventures of a blogging ship propelled by weather data (Wired UK). (n.d.). *Wired UK*. Retrieved February 6, 2013, from http://www.wired.co.uk/news/archive/2013-02/05/ship-adrift

This Old (Open Source) House: Man Renovates Home on GitHub | Wired Enterprise | Wired.com. (n.d.). *Wired Enterprise*. Retrieved March 11, 2013, from http://www.wired.com/wiredenterprise/2013/01/this-old-house/

Thought Experiment: Build a Supercomputer Replica of the Human Brain | Wired Science | Wired.com. (n.d.). *Wired Science*. Retrieved May 15, 2013, from http://www.wired.com/wiredscience/2013/05/neurologist-markam-human-brain/

Ullman, E. (2013). *Close to the Machine: Technophilia and its Discontents* (First.). Pushkin Press.

Urban Computing and its Discontents | Situated Technologies. (n.d.). Retrieved November 6, 2012, from http://www.situatedtechnologies.net/?q=node/77

Urban Probes. (n.d.). Retrieved July 25, 2013, from http://www.urban-atmospheres.net/UrbanProbes/index.htm

Urban Versioning System 1.0 | Situated Technologies. (n.d.). Retrieved May 28, 2013, from http://www.situatedtechnologies.net/?q=node/85

Verran, H. (2001). *Science and an African Logic* (2nd ed.). University of Chicago Press.

Virtual Reality Allows Adults to See World Through a Child's Eyes - Wired Science. (n.d.). *Wired Science*. Retrieved July 17, 2013, from http://www.wired.com/wiredscience/2013/07/virtual-reality-perspective/

Wall, L. (2000). *Programming Perl* (3rd ed.). Beijing ; Cambridge, Mass: O'Reilly.

Wall, L. (n.d.). Perl, the first postmodern computer language. Retrieved March 20, 2013, from http://www.wall.org/~larry/pm.html

Watch: A Hidden Camera Captures A Package's Hidden Journey. (n.d.). *Co.Design*. Retrieved April 4, 2013, from http://www.fastcodesign.com/1672262/watch-a-hidden-camera-captures-a-packages-hidden-journey

Weber, S. (2004). *The success of open source [...] [...]*. Cambridge, Mass. [u.a.: Harvard Univ. Press.

What is free software? - GNU Project - Free Software Foundation. (n.d.). Retrieved September 29, 2013, from http://www.gnu.org/philosophy/free-sw.html

Whitehead, A. N. (1929). *Process and reality : an essay in cosmology.* Cambridge: Cambridge University Press.

Wiener, N. (1961). *Cybernetics, or, Control and communication in the animal and the machine* (2nd ed.). Cambridge Mass: MITPress.

WikiHouse. (n.d.). Retrieved September 29, 2013, from http://www.wikihouse.cc/

Woolgar, S. (1988). *Science : the very idea*. Chichester: Ellis Horwood.

Woolgar, S. (2002). *Virtual society? : technology, cyberbole, reality*. Oxford: Oxford University Press.

Yaneva, A. (2012). *Mapping Controversies in Architecture*. Ashgate.

"*Yeah Science Bitch!!" {MEME}*. (2013). Retrieved from http://www.youtube.com/watch?v=Dbt-OfbQTpw&feature=youtube_gdata_player

Yeah Science, Bitch | Know Your Meme. (n.d.). *Know Your Meme*. Retrieved July 18, 2013, from http://knowyourmeme.com/memes/yeah-science-bitch

Your Startup's Legal Docs: Now on GitHub | Wired Enterprise | Wired.com. (n.d.). *Wired Enterprise*. Retrieved March 11, 2013, from http://www.wired.com/wiredenterprise/2013/03/series-seed/

Yuill, S. (2008). Concurrent Versioning System. In M. Fuller (Ed.), *Software Studies: A Lexicon*. The MIT Press.

(n.d.).