Digital Design Methods:

Open Computing For Critical Spatial Simulations In Architecture

Research Question

My research is about defining an alternative open-source architectural practice of computational methods. rather than tools? How can such a practice address the question of access to complex computer technology within architectural design processes and what new kinds of design process can emerge from this?

Research Aims

- To define an ecology of digital techniques, that includes the physical consequences of virtual technologies as well as access to the knowledge by which it is constructed
- To engage with computational design as a democratic and participatory tool
- To develop alternative open source computer techniques to capable of countering existing closed models of spatial simulation in architectural design

Theoretical Context

My work is framed by contemporary feminist theories of situated knowledge and specifically the construction of scientific knowledge through practice [Haraway, Braidotti,Latour] as well as a historical narrative of computing that has lead to alternate formations such as 'cyborgs' [Haraway] and 'post-humans' [Hayles]. Within this context it is my intention to 'appropriate' specific ideas, technologies and code that are entangled with these aspects computational design and to then explore the alternative uses within architecture that they suggest.

I not only want 'recycle' these concepts and algorithms, but to then incorporate them into a alternative, modified idea of 'open source' [FLOSS], that retains a focus on access to computational technology, but that also includes ways of democratising the means of constructing scientific knowledge.

More broadly, the question of computational science is addressed though the ontological and epistemological significance of the concept of emergence [Deleuze, De Landa] and the science of quantum physics [Prigogine, Barad]. Terms such as 'measure' and 'value', so common in computing, are re-conceptualized to reflect the significance of these new strands of scientific research, both in terms of what and how we can measure.

I will place this within a new form of ‘Digital Ecologies’ [Guattari, Bateson], formed from the consequences – knowledge, material, environment, bodies, politics, technologies, economies - of the current paradigm of ‘technological transience’ [Gabrys, Haraway].

Principle Themes

Using these theoretical strands, my PhD focuses more specifically on three particular aspects of computing that are linked through the historical construction of contemporary understanding of what constitutes computational technology – algorithms, interface and simulation – and I am currently undertaking early case studies and projects related to each theme.

Algorithms – as an explicit records of critical processes

Algorithms are crucial to both software engineering and computer science. But in each case, they operate in different ways. In computer science, they are seen in abstract, as a description of the tasks to be carried out, whereas in software design they are more practical - through a programming language, algorithms do things. But this conflict is in fact a conflation. Whilst there is a formal separation between an algorithm and the data that it requires to execute its tasks, the relationship between the two is such that an algorithm stripped of its data – i.e. it's context – is no longer an algorithm. Therefore, the algorithm as an abstract concept is problematic – you cannot simply 'think algorithmically' [Goffey]. But you can 'act' algorithmically. Algorithms are ways of recording and distributing actions as well as communicating them.

Interfaces – as social models of involvement

Interface, and specifically the GUI, is what commonly marks the difference between 'code' and 'software'. However, the knowledge of the user blurs this apparently clear distinction. Furthermore, 'Human Computer Interaction' (HCI) operates through an over simplification of feedback mechanisms of 'stimulus' and 'response' and as such, designing with a (mythical) 'typical' user in mind that is both empowered and made invisible [Fuller]. Software interface is more usefully understood as social models of involvement in the complex processes and technologies that sit 'behind the screen'.

Simulations – as speculative open experiments

Simulation is intertwined with both the historical development of computational technology and science and a wider understanding of its potential uses. Cybernetics, and the subsequent research into Artificial Intelligence is indicative of this duality. The architecture of computer memory is itself a simulation of the human brain that the computer then seeks to imitate through (algorithmic) techniques such as neural networks [Slack]. Simulation represents both the potentials and perils of computational technology. They risk taking the simplified abstraction of the complexity of the world and return it as it's underlying structure [Hayles, Harraway, Latour]. Furthermore, our ability to contest the conclusions of complex simulation is limited by access to both hardware, through the difference between elite and personal computing power [Eglash] and software, through commercial dominance and control.

Ongoing case studies & projects

Algorithms - Tactile programming: 'Material Computing' For Architectural Design

'Tactile programming' is the title of a workshop that I have taught at the School of Architecture, University of Sheffield, over the past 6 years. The purpose of the workshop has been to challenge the common narratives of computational design in architecture and specifically that of 'parametrics'. This was done by adopting physical modelling techniques in place of the actual parametric software tools. The workshop was undertaken during the mid-point of Masters level student projects and marked the transitory point from data collection and diagramming, to the exploration of it's spatial consequences. Students were asked to make physical models of their project data, and use the models as a way of exploring alternate design possibilities and scenarios.

'Tactile programming' is about the extent to which you engage with algorithms as both a means of recording and communicating action.

> This was presented at *'Facts and Fictions'* 10th Annual AHRA Research Student Symposium, Sweden, May 2013

Interfaces - Branch, Merge, Commit: New forms of Open Source for Designing With BIM

This is a case study of a piece of versioning software - *GitHub* - that is commonly used the open source software development community and offers potential for developing a critical method of open source design for architects. I have taken three modes that are characteristic of BIM work flows - *collaborative, concurrent and continuous* – and explore them using the terminology and actions associated with versioning software – *commit, branch and merge*.

Collaborative Behaviour (Commit) - parts, components, assemblies

Concurrent Behaviour (Branch) - Many versions

Continuous Behaviour (Merge) – Authorship

As part of this case study, I have published my ongoing PhD work on *GitHub* and will use this as platform to investigate the above ideas, particularly that of authorship.

> This work will be presented at *'Pedagogy meets Big Data and BIM' Conference,* The Bartlett, UCL, London, June 2013.

Simulations - environmental 'simBot' – mechanisms of feedback for design and simulation

This project is about building a prototype tool for critical engagement in spatial simulation. The initial purpose is to make a critical commentary of AI and cybernetic feedback by creating a 'bot' that is able to 'read', 'understand' and 'comment' on the work of students participating in a (separate) environmental simulation workshop. They will be using *Energy Plus* to solve a specific design problem and the 'bot' will provide 'live' feedback on the progress of the students. The 'bot' will provide a subversive commentary

and an alternative representation of the data provided by the simulation software with intention of encouraging the students to challenge their preconceived ideas around precision and interpretation in simulation.

My intention is to take this prototype and develop into a series different 'cyborg-instruments', at different scales, as a way mediating architectural designers interactions with existing computational technologies of simulation. These 'cyborg-instruments' will form the basis of my ongoing PhD research.

'simBot' prototype to be tested at *Environmental Simulation Workshop,* School of Architecture, University of Sheffield, July 2013.