TITLE: TACTILE PROGRAMMING: 'Material Computing' For Architectural Design

AUTHOR:
Phillip Langley
School of Architecture
University of Sheffield
arp12ppl@sheffield.ac.uk

KEYWORDS: technology, programming, emergence, resilience, pedagogy

ABSTRACT:

Architects are typically consumers of software as products, rather than users of software as technology and the relationship between computers and architecture can often be fractious - viewed with either complete suspicion or total devotion. In either case the role of the computer is often seen in the same way - to supplement an pre-existing design process rather than to fundamentally change it .

This paper investigates how critical approaches to computational methods can inform a practice of architecture that is defined *by* computational methods, rather than simply composed *of* computational methods. To that end, this paper proposes a specific method for data modelling that replaces the computer [and its software] with a physical modelling technique. This method has been developed during a series of generative and parametric workshops, entitled “Digital Design Tactics” with students of the School of Architecture at the University of Sheffield .

Central to these workshops were two ideas - the basic of a computer algorithm understood through the syntax of a programming language - 'loops'...'ifs'...'ands'...'ors' – as well as the use of simulation to explore emergent conditions. These are then re-conceptualized as physical, almost 'mechanical' processes that can be recreated through making physical models. The models created by students were based on data collected for their projects could be spatial or abstract, at the scale of the city or a the site, but were always *relational*. The models were used to develop strategies within complex projects that combined multiple scales, actors and changing over significant periods of time. The models became a kind of manifold for the discovery of the projects possibilities that may not otherwise have been visible.

As well as playing' with the tradition of architectural model making, this is a kind of *tactile programming,* where the syntax and semantics are replaced with physical operations, and represents both the software and hardware of the conventional computing paradigm. Within this method, there is no longer a computer and user. The conventional notion of interface is outdated.

This proposal of *material computing* offers a means of democratising technology and provides a pedagogical frameworks for teaching a critical approach to such technology. The use of computers is often restricted due to the high entry requirements in terms of cost and specialist skill. By reconfiguring what software and hardware represents, the 'user/ designer' of this new form of computer, no longer constrained, is able to create, maintain and modify their own 'machine'. This is also then a means of technological resilience – the purpose, portability and functionality of this method are determined by the user and not other forces [i.e. software designers or hardware manufacturers] and the 'science' of computational technology can be appropriated separately from its physical manifestation.