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

## Phil Langley

### School of Architecture, University of Sheffield, https://github.com/phiLangley arp12ppl@sheffield.ac.uk

## Abstract

This paper proposes a new method of designing with BIM technology firstly with the application of two contemporary methods of computing – open source and object orientated - and secondly the adoption of the versioning software platform 'GitHub' as a means of harnessing them and to encourage wider participation in the processes of designing with BIM.

BIM software represents a significant move from the traditional skeuomorphic paradigm of CAD as drawing board to something new, that is in and of itself. Whilst this offers the designer huge potential, it also presents a difficulty. BIM software requires alternate, and unfamiliar, design processes and work flows – collaborative, concurrent, continuous.

I propose to re-configure two existing methods of computing; open source (from software design) and object- orientated (from approaches to programming and code).

Briefly, open source is traditionally understood as the release of the source code for computer programmes. Typically, however the environment of open software is one where the users and developers are basically the same and in which the 'extended negotiation' between user and programmer becomes a closed loop. This is analogous to existing BIM working practices – closed loops of expert users. In the method proposed here, the loop must be prised open, to allow access and contribution to a wider range of of participants. In programming terms, an object orientated approach is one whereby elements of the code are organised in such a way that they may be more easily understood and edited as well as quickly reused and re-structured – they are assembled components in the same way that BIM can be described as a technology of 'assembled objects'.

By viewing BIM design methods as component assemblies, and then opening access to both the components and the rules of assembly, BIM practitioners have the possibility to create new methods of designing with BIM, through wider participation in the processes associated with it.

## Section(s)

1. Introduction – Defining a BIM method

**2. Collaborative (commit) - Part, component, assembly**

**3. Concurrent (branch) – Many versions**

**4. C**ontinuous (merge) - **Authorship**

**5. Conclusion - My open source PhD**

## References

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

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

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

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

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 Fuller, M. (Ed), *Software Studies: A Lexicon* (pp. 15-19). The MIT Press.

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

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

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/>

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.

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>

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/>

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

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

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

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). Interupt. In Fuller, M. (Ed), *Software Studies: A Lexicon* (pp. 153-167). The MIT Press.

# Presentation Outline

## Speaker(s)

Phil Langley

## Abstract

*As above.*

## Summary

1. Introduction - Defining a BIM method

This will set the context for a speculative description of an open source BIM design method as something separate from the tools and technology that the industry use to define BIM (e.g. Revit, IFC....). I identify 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*.

**2. Collaborative Behaviour (Commit) - part, component, assembly**

Outside of industry norms, collaboration is not about 'open' technological platforms, to access complex model assemblies. It is about a different idea of system access. Where can I make a contribution? BIM software packages (for reasons of industry productivity) mediate our interaction with the elements of our design – features such as wall components, for example, are very strict about what constitutes a wall. Object-oriented design processes give me the possibility to access and re-assemble content in new ways, but they never allow me to redefine what the 'thing'(or *component*). On the other hand, the 'procedural' algorithms that make up these components allow me much more scope to change the *parts* of *my* *component* – I now have other ways of entering and collaborating in the process. Collaborative methods of BIM must support both of these behaviours.

**2. Concurrent Behaviour (Branch) - Many versions**

What does open source really mean, beyond 'open access'? Versioning technology, such as GitHub offer decentralized, collaborative and genuinely open work flows that are most commonly used in software development. Concurrence, as a design behaviour, is used to develop the features and functionality of a piece of software over time, without upsetting that core stability of the software itself. It also means that I can take my own 'branch' of a piece of open source software and modify only one small aspect (a part of the user interface for example), while retaining the functionality of the software assembly. Github and similar software technologies are being used in more novel ways than just software development and offer a powerful modification to the centralised (and therefore closed) industry models.

**3. C**ontinuous **Behaviour** (**Merge) – Authorship**

In architecture we still have the idea of *the* designer but continuous design processes change completely the idea of authorship. In the industry context, this question is most commonly about the responsibility for a 'shared' model. In an educational context, it is a bit more complicated. We need new ways to evaluate the contribution of individuals to a continuous design model.

**4. Conclusion – My open source PhD**

My ongoing PhD, including the notes and slides of this presentation, is available for use on GitHub https://github.com/phiLangley