A Study of Process in Design:
Curatorship, cloud intelligence and applied research

Philip Plowright, Lawrence Technological University, synchRG
James Stevens, Lawrence Technological University, synchRG
Dr. Anirban Adhya, Lawrence Technological University, synchRG

Abstract

The architectural design community is faced with a shift in focus from an object-centric model of design creating discrete aesthetic objects in the landscape to addressing valid and relevant process-oriented solutions involving complex problems within complex systems. In this evolution, traditional boundaries of ownership, ego and control of knowledge or product become insignificant and flow and accessibility issues of the design process become critical for examination. The issue is how to adjust an old model of project methodology which is not structured to support a systems-based approach. The investigation addresses the idea of systems, curatorship, invited experts, open source standards and focused crowd-sourcing as a core operational structure for knowledge creation and dissemination, research, and practice.

This paper presents the organizational aspect of an research and design structure developed through analysis of open source processes. In addressing a tradition research/design framework, the role of primary investigator was interpreted as a role of curatorship rather than ownership. The proposed organizational structure focuses on catch-and-release data framework, point-of-release timings, problem-set organization and co-existing top-down/bottom-up authority structures. Inherent in the structure, the curator defined the catch-and-release strategies for informational feedback loops and point-of-release of defined segments of research (problem-sets) to the interested community. The two functions then engaged the larger project by tying problem-set knowledge and solutions into larger lines of enquiry. In addition, by linking open source contribution to curatorship, the project addresses the possibility of a parallel top-down/bottom-up authority structure. A result is a potential asset for maintaining a position as a dedicated generalist within a culture of increasing specialization while addressing the benefits of systems process which supports a culture of innovation.

Introduction

We are at a point of a societal event horizon with the intersection of social media, increased specialization, deep disciplinary knowledge, proprietary research and computer processing power (Stevens et al 2009). This point is an opportunity to address what is increasingly becoming an introverted design culture within architecture (Till 2009), and to address the integration of research, both internal and external to the discipline of architecture, into that design process as an examination of methodology. Previously, the open source model of project methodology was highlighted as a possible adaption to the architectural design process in order to address the solution finding in a creative problem-solving process. In addition, the potential to use a lateral and self-organizing system may address a design process in architecture has been reduced to being self-referential and mainly internally focused. Architecture has been accused of often ignoring everyday problems and disregarding the importance of non-formal functions, events and interactions within complex real-world systems (Hillier 1996) .

This paper first identifies some existing issues in the use of research, process and design within architecture. The concept of openness and the connection to innovation are then explored. Next, the potential of alternative processes by examining organizational systems based on social media and open source (permeable barrier) systems are identified. Extracting potentials from both existing case studies and from original experiments in design process, an alternative organization structure for an architectural design (formal/technical/social) is detailed.

1.0 Design, Research and Process

Traditionally, design has achieved "success" through a process of trial and error which creates an iterative but undefinable methodology, labelled 'intuition' and 'practice'. However, this position is uncritical, difficult to transfer in an educational setting due to its esoteric and defensive posture, and limiting in vision. Architectural design has maintained a narrow focus in terms of detailing both interest and success (Till 2009), a focus which often ignores relevant extra-disciplinary knowledge and even internal research (social, technical, environmental) from being applied. The iterative process in architecture has been reduced to self-referential process. What is needed instead is iterative processes that takes *any* relevant knowledge from *any* discipline and applies it to a solution focused model (i.e. It has to do this, I don't care how we get there). This is fundamentally the core of a creative problem solving process, the heart of any design ecology.

The traditional design process is generally a top-down and compartmentalized process, due to both business organizational models and the complexity of the product to be delivered. This division can be internal or external, dividing tasks such as the design of a curtain wall to a department of experts or contracted to a specialist. The general rule is to divide into categories based on disciplinary knowledge, generating discrete teams for schematic design team, HVAC, electrical, construction documentation, construction administration, site integration, and landscape, to detail a few. Failures often occur when there is a conflict between the components and overall aesthetic or formal composition, or significant lack of organization between the siloed specialist teams. However, the architect does not often think of this failure in terms of a conflict between the intentions of the parts as discretely considered and the total system, of which they are an element. In an article examining notions of architectural innovation of product technologies, where architecture is used not in the disciplinary way to mean all that is held in the discourse of the design of buildings, but as it is used in Organizational Science to mean the distribution and composition of the whole, Rebecca Henderson and Kim Clark note;

> *"This distinction between the product as a whole —the system—and the product in its parts—the components—has a long history in the design literature (Marples, 1961; Alexander, 1964). [...] A component is defined here as a physically distinct portion of the product that embodies a core design concept (Clark, 1985) and performs a well-defined function. When the two distinct portions are separated (components/architecture) there lies an opening in the process of architecture and design, one that allows for components to be distributed in a designed system for maximum productivity. Furthermore, the distinction between the product as a system and the product as a set of components underscores the idea that successful product development requires two types of knowledge. First, it requires component knowledge, or knowledge about each of the core design concepts and the way in which they are implemented in a particular component. Second, it requires architectural knowledge or knowledge about the ways in which the components are integrated and linked together into a coherent whole." (Henderson, Clark, 1990).*

Taking knowledge from Organizational Science, organization and communication channels have been shown to be critical to both task completion and effectiveness in design (Galbraith, 1973; Arrow, 1974). As researched and described by Henderson and Clark (1990), the communication channels of an organization embodies its architectural knowledge as a relationship of the parts (structure) to the whole (limits of content). This means the communication structure *pre-determines possibilities of design solutions based on component knowledge*. Communication channels create a set of filters, so that each link focuses on only key relationships and masks any adjacent or minor element, locations were solutions to the whole might be found but are unintentionally suppressed as they don;t solve the local. Most structured organizations are, by definition, not able to be fully innovative as their own rigid structural shortcomings limit the range of possible answers before the question is even asked. This includes proprietary knowledge bases as closed systems which again limit the potential of an innovative solution and the use of tangential research or access to extra-disciplinary knowledge.

2.0 Openness and Innovation

Traditional disciplinary barriers often inhibit the acquisition of knowledge from beyond the boundaries of that discipline by defining terriotorial control however openness is important. A study completed by Karim Lakhani, Lars Bo Jeppesen, Peter A. Lohse, and Jill A. Panetta titled *The value of openness in scientific problem solving* (2007) present the case using existing literature and studies from Mulkay, Stephan, Hagstrom, Campbell and Grushcow that there is often even difficult transfer of knowledge within the boundaries due to issues of "career, publication priority, intellectual property, and financial concerns often trump openness to the potential detriment of overall scientific advancement". The study recognizes a problem solving approach called "broadcast search" based on the premise that "knowledge is unequally and widely distributed in society" (Lakhani et al 2007, 4).

Another key finding in the Lakhani et al study addresses predictors for innovation. The study found a 29.5% resolution for previously unsolved scientific problems which had remained within the industrial proprietary centers, such as R&D labs, by broadcasting for a solution *outside* of its disciplinary boundaries. That is innovation has been proven to occur at a higher ratio when a closed group which speaks the same specialized language (a discipline) must translate the problem they are attempting to solve into a general language to discuss in an interdisciplinary manner. Another way of saying this is a broadcast search is performed which introduces knowledge provided by 'outsiders' into a closed system (Lakhani et al 2007). For example, one solution provided through an inter-disciplnary knowledge transfer website (www.innnovate.com) provided a solution to a previously unsolved problem within the boundaries of toxicology by by applying standard processes in protein crystallography, processes that were unknown for toxicology. It was the transfer of knowledge from one discipline to another which was the key to innovation and problem solution. The type of knowledge was particular as well, it was *deep knowledge well encoded in disciplinary syntax.*

What is important for architecture to realize is the core of innovation is found in knowledge transfer, not product, material or manufacturing. This knowledge transfer is essentially one of the central elements in the creative problem solving process which is the essence of design. Instead of defending disciplinary boundaries, the critical activity is to make those boundaries permeable so to allow external deep knowledge to filter into architectural design through an organizational structure. It is the potential of broadcast social media which opens up the ability to present information and request solutions from outside the discipline.

3.0 Permeable Barrier Organizational Systems

Looking outside of the discipline of architecture, the Open Source Software (OSS) industry provides a relevant example for the organization of communication, tasks and evaluation of success and failure. In terms of broadcast social media and cross-disciplinary knowledge transfer, the OSS community has shown how a large pool of eyes and contributors, often not from within the core of the process or discipline, can significantly improve quality and innovation (Raymond 2001).

There are strong parallels between software development and architectural design processes. In both disciplines, varied and numerous design variables must be synthesized into a cohesive whole. Open Source Software philosophy uses a completely different model than the proprietary system of commercial firms. The style that was pioneered by Linus Torvalds, the founder of Linux, was "release early and often, delegate everything you can, be open to the point of promiscuity" (Raymond 2001, 3). The organizational system is further described:

> *"In traditional, commercial software projects, the response to the Brooksian critique of large teams is to divide and conquer, by fiat. The system is deliberately divided into smaller components, and the developer pool grouped into manageable teams which are then assigned to those components . By contrast, Open Source Software (OSS) projects are not formally organized, and have*

> *no pre-assigned command and control structure. Not one is forced to work on a particular portion of the project. Team members contribute as they wish in any number of ways.“* [Bird et al 2008, 1].

As the project forms, interested parties participate based on their interest and abilities based on project vision and not on a rigid management structure focused on a predetermined outcome. This doesn't mean the project is anarchistic, it instead relies on knowledge content holders to rise into positions rather than to be placed there. It is self-regulating, efficient and organized on competencies rather than political structures. Anyone can contribute *if* they make their contribution relevant to the process. If it is not, the system will eliminate that contributor and/or contribution by a process of natural selection.

Open Source Software is predicated under the assumption that when a program is developed, the source code will be openly distributed and redistributed. Open exchange of ideas and knowledge and an accessible public development process is required. For the open source organizational alignment to work, participants must commit to a common set of rules.

- All software is created with the foundation called source code. This code is made available for free distribution.
- When using the source code developers are then expected to make the new software available to the originator and future developers. This is a critical step to maintain the circular process of development.
- If programmers modify the source code then the new software will be renamed or given a version number. A small modification or a “patch” is often an exception.
- New software that embeds the source code can not place further licensing restrictions that would prohibit future development.
- Distribution of the source code can not be restricted to exclude specific professions, person or groups.
- Software innovations are not proprietary but mutually beneficial.

The Open Source Software community identified early that not one individual could solve all of the problems facing the profession and that it was inefficient for programmers to replicate existing efforts, and even considered unethical for anyone to 'hoard' code. This understanding has streamlined the development process and has allowed the community of programmers to respond to a rapidly changing market (Stevens et al, 2009). The process of open decision making can appear foreign to the academic and professional architect. However, the fluidity in the OSS allows efficient management and organization of large amount of data along with quick modification of software components to adjust to the evolving outcomes and needs. At the core of an open source philosophy, and one that should also be considered as the foundation architectural design, is an iterative process. However, unlike strong hierarchical structures , the iterative process uses broadcast media and a fluid, open structure to be innovative and solution finding rather than self-definitive. More to the point for architecture, the Open Source community identifies its core competency to be a design-focused process, not a code-focused industry (Raymond 2001, 19).

3.1 Curatorship

In terms of open source processes (management organization), one of the important factors is not that projects are designed by committee or emerge as part of a general anarchistic crowd effort, but that they are guided by an individual or small team in a process that is *open* to external influences. This is a very important distinction. Eric Raymond, one of the originators of the open source label, uses the terms *coordinator* and *leader* to describe this position. Due to the requirements of selecting, editing and merging elements during the design process, we use the terms *curator* and *curatorship*.

4.0 Alternative Organizational Flow

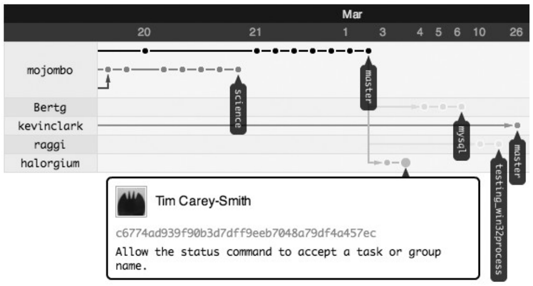
Using knowledge gained from the analysis of existing processes which encourage openness and innovation, possible alternatives to the existing organizational structure in architecture design were examined by running a series of test processes through social networking applications. The intent of the systems tests was to established points of alignment with current research in other fields regarding organizational structure and the potential for a open systems which will allow diverse input while still maintaining curator overview. The goal is to allow for an environment of innovation by transferring research and knowledge from the periphery, where knowledge traditionally is housed in architecture, in a non-rigid structure. Curatorship is a key term in developing a successful open source system for architectural research.

4.1 System Tests

The system tests were developed using existing media, and analysis for potential based on factors listed above. The tests were executed and analyzed with the intention to extract potential elements to be reintegrated into a proposed model for architectural thought production. Currently, general categories within emerging social media are blogs (share personal views and experiences with others in a formal setting), wikis (group contribution building knowledge with collective intelligence), micro-blogging (quick bursts of information which are informal & asynchronous, for example Twitter), RSS feeds (constant connection to relevant sources of news/information), social bookmarking (extension of social network, the sharing of discovered information), instant messaging (quick and direct synchronous communication with others), and social networks (used to find and connect with people like common interests or pasts, for example Facebook and Linkedin). This list can be further organized by 1) connections and context (social networks, IM, social bookmarking), 2) personal broadcasting (blogs, micro-blogging) and 3) Collective Intelligence (wikis).

The first test involved connections and context, using Facebook as a broadcast media. It was thought that due to the high visibility and high population of this social network site, the potential to reach a large audience was possible. A contact database of possible professional and academic contributors were identified and provided with a group invitation. A curator originated the problem and requested a solution. The sourced interested parties where encouraged to pass the problem set to others they thought might be interested in a viral process. Respondents were asked to return possible solutions with the promise of a reward (notoriety in this case). Once the information was returned, it was distributed back out to the social network for critique and modification. This system is a *catch-and-release* information loop; challenges released and solutions caught and then released again. The systems theory approach lead to the isolation of various operational factors into lines of enquiry to be solved individually and at various scales.

A second test involved the ability to transfer hard information in both upload and download directions. This is not supported by most social networking sites. A open digital repository was set up in another location and linked into the broadcast media for visibility. Due to the attempt to maintain openness, the repository was not secured in any way, in order to allow for maximum access.



Tim Carey-Smith

c6774ad939f90b3d7dff9eeb7048a79df4a457ec

Allow the status command to accept a task or group name.

Figure 1: Versioning notation in Github

A third test was performed with an alternative social networking site based in the computer software industry. Part of early analysis illustrated that one of the major components necessary to realize an open source research project was version control and the tracking of the catch and release process. Version control has the ability to track changes, catalogue differences and most importantly allow for parallel development with the functionality to merge information (see Figure 1). This third test was hosted by the social coding site Github (github.com) Git has multiple important attributes not available on the standard web 2.0 social networking site. First, it provides a secure public repository that is open to all other git members. Furthermore the site manages your data through file versioning located on your local machine. Each contributor works independently and tracks changes through versioning. At key points (decided by the contributor) the files are “committed” to the repository with notations. The commit is a researchers way of providing their data to the community. If the resource was developed by modifying or improving an existing asset in the repository (as is normal) then the contributor will conduct a “pull request”, this request notifies the original author that modifications have been submitted. The original author then has the ability to “pull” the changes into the original or leave them parallel.

4.2 Analysis

The system tests illustrated several issues and opportunities. The intention was to develop an understanding on how to operationalize weak connections in the design process and to pull adjacent or extra-disciplinary knowledge into a core process by using a catch-and-release data framework, point-of-release timings, and examining problem-set organization. It was hypothesized that a hybrid culture of co-existing top-down/bottom-up authority structures could develop.

Transparency of information
Problem-set organization needs to address the degree of transparency of information. As part of the tests, the problem was assigned and critical information was masked by the curator. It was believed that through masking background information, the widest possible set of solutions would occur. However, this bias became a possible issue in pre-determining outcomes. As well, in a lateral design process, the lack of parameters was shown to inhibit design options rather than encourage them. The balance between transparency and masking has to be well considered.

Version control
One of the key elements in an open source project is the ability to document changes and 'versions' of the current project. The tests revealed that most existing broadcast media does provide any tools for versioning, with the exception of Github. Openness of information is also necessary in versioning in order to allow the potential of peer-to-peer curatorship. Version control also allows for branching to occur, points of potential of a project when it chooses to go in one direction of development but there is an alternative and valid second course of action. Current social media, and organizational structures do not lend themselves well to versioning, where a line of enquiry can be shared across proprietary boundaries.

Iterations
One of the key factors in an innovative design process is an aggressive iterative process which is *fully connected* to outside influences. The issue that was raised by the system tests was the degree of control the curator should and does have in the process. This relates to the issue of transparency, wherein the curator would have the option of closing the challenge loop or re-issuing a new challenge focused on synthesizing each suggestion. There is a large degree of pressure of judgement, akin to a top-down proprietary system, if 1) the curator alone is responsible for determining the most viable solution, 2) whether it was reissued in an iterative process, 3) whether it would be used internally only, 4) how the information provided is evaluated and 5) when/how the process is continued.

Role of curator
The need for curator participation exposed a flaw in the adaptation of the open source process as it relates to architecture research. As noted above, in order to effectively produce multiple iterations, the curator must stay fully engaged. The problem exists when the sourced community overwhelms the curator, or the curator is too dominant in the process, thus losing the power of open-system dynamics. This can cause confusion, lack of direction and loss of interest on part of the community, project and curator.

4.3 Proposed Elements of Organizational Flow

These tests revealed potentials and shortcomings in developing an alternative model of communication and development focused on maximizing innovation. Extracted from the systems tests and analysis, four primary elements were identified in order to be reintegrated into an organizational structure. These are 1) the release, 2) the catch,3) the pull request and 4) the fork. It is proposed that these primary components form the basis for an open source and circular architectural design research methodology.

The release (Figure 2) leverages the open source value of publishing early and often. The initial broadcast of research intention should be an investment into vision of the project. The researcher is motivated by the prospect of further development by others and developing knowledge that can be later incorporated back into the original proposal (or not). The release also does not prohibit the initial researcher from developing the project as long as each subsequent version is committed to community knowledge. This is done by versioning, logging and distributing relevant intellectual knowledge. The point of release, particularly the initial commit is a significant shift in the standard mode of research by making the findings available for all and providing the potential for cross-disciplinary contributions. Releasing findings from the "academic silo" will provide feedback and development ordinarily not received.

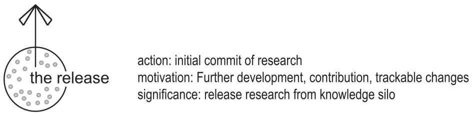

the release
action: initial commit of research
motivation: Further development, contribution, trackable changes
significance: release research from knowledge silo

Figure 2: The Release

The catch (Figure 3) is the process of searching, finding and obtaining research findings. The catcher is not only the beneficiary of transferable and repeatable knowledge but also a contributor to the original findings. The catch is the first step in community benefit and contribution. Exclusion of participants does not occur at this step in the process, anyone can catch and contribute. The idea of full and open participation is critical to insure cross-disciplinary contributions and it also allows for multiple interpretations and vectors of enquiry. The catch represents a shift in point-of-release timing, that is to say, the point in time that academia limits participation of others – curatorship. The action of curatorship is moved to the end point of the contribution, allowing for the participation to occur freely with the significance of the findings being judged only occurring when requested by the catcher.

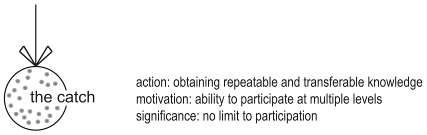

the catch
action: obtaining repeatable and transferable knowledge
motivation: ability to participate at multiple levels
significance: no limit to participation

Figure 3: The Catch

The pull request (Diagram 3) is the point of curatorship. When a researcher has caught findings and developed them further, new findings only run in parallel lines. It takes a pull request to merge the new findings into the original release. When a pull request is sent the catch researcher is asking the researcher that initially committed the findings to consider the pull to be incorporated into the original, thus furthering the project as a whole. At this point the initial commit researcher becomes a curator and decides relevance. If the pull is accepted, the findings have completed the circular open-source process and have benefited from contributors ordinarily excluded. If the pull is rejected or ignored, the findings remain parallel. It is important to recognize that by rejecting or ignoring a pull request does not remove the content to others but only prevents incorporation. The rejected project then forks and becomes a new version (1.1, 1.2, 1.3) and is available to be caught by others, improved and developed independently.

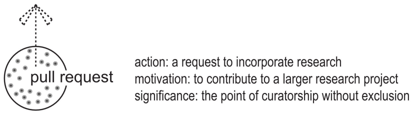

pull request
action: a request to incorporate research
motivation: to contribute to a larger research project
significance: the point of curatorship without exclusion

Figure 4: The Pull Request

The fork (Diagram 4) occurs when findings are either rejected or ignored by the initial commit researcher but also when the catch researcher intends to fork the project from the onset with the intent of using the findings as foundational information. The forked research re-enters the reiterative cycle of research and will proceed forward in the same way as initial commit research. The pull request is an action of contribution and curatorship, the fork is a mechanism of inclusion, insuring that *all* research is given the opportunity to be found, used and developed.

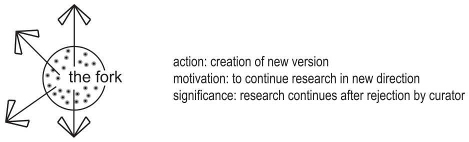

the fork
action: creation of new version
motivation: to continue research in new direction
significance: research continues after rejection by curator

Figure 5: The Fork

4.4 Interdependency of Elements

The four proposed elements of the organizational flow only serve to define specific points in an iterative process. It is their relationships, connections and requests that create a natural process of curatorship.

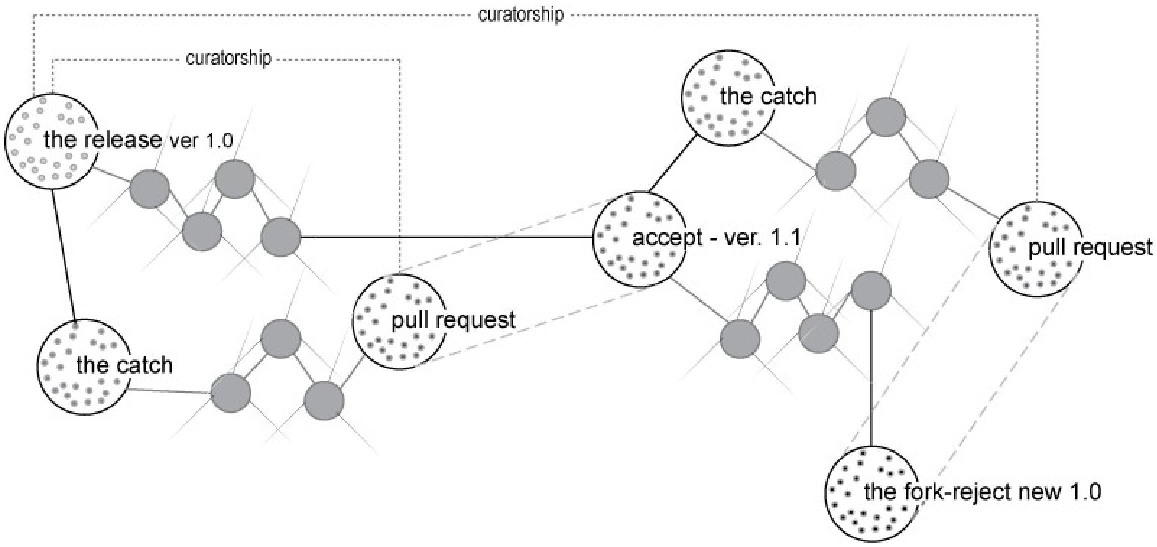


Figure 6: System Interdependency (variation)

Natural curatorship is a process whereby peer-review, or curatorship, occurs only as needed and is conducted by the beneficial nature of new research within the organization flow (i.e. does this new or developed research help me?). It is the need for new sets of data (the catch) or with the decision to, or not to, incorporate new findings (the pull request and the fork) which triggers curatorship. Although there are a multitude of possible relationships, there are only a few key relationships that define the organizations flow.

The release and *the catch* have a symbiotic relationship. The catch researcher needs new research and the release researcher needs further development. Since the flow is structured to be controlled from the bottom-up the Release researcher does not have the ability to request specific development (so not to limit scope), this is decided by the catch researcher in a self-guided exploration. At this stage of the relationship, the research can respond in one of three ways;1) the research can be developed in parallel (indicated by the filled circles in Diagram 5),2) it can be returned for incorporation of the whole (*pull request*) or 3) it can fork (by choice or denial by originator). Prior to the pull request, there is natural curatorship. The catch is driven by demand, if the research is not relevant, it will not be caught nor developed.

The *pull request* and *the release* are defined as a question rather than as a relationship.. The pull request is simply a request, or an offer, to incorporate new findings into the original release of research. This is a critical point of curation when the originator decides relevance of the new findings based on the original intent of the release. Denial of incorporation is not necessarily a negative response to new work but an assertion on the part of the curator to maintain an appropriate roadmap for the project. In fact, the denial of a pull request can be seen as a complement, the curator may feel that the new findings have contributed significant new knowledge that should remain separated (or fork) allowing for further development by others (a new *catch*).

The *fork* is a new starting point, and it is related to previous releases through a parent-child relationship. It is the benefactor of transferable knowledge but is a contributor for potential new catches. The fork, in many ways, exemplifies the organizational flow. It is the child of the release, the catch and the pull request, as well as a product of curatorship. This does not mean the fork's research is immediately relevant and significant. Since natural curatorship on the original line of research continues with a new catch, the point of the fork is a *project stub*. A project stub is one that shows potential but needs developing. Every fork will produce a main line of enquiry and a stub line left for later expansion. If both lines of the fork are developed, it may be that parts of the development line of the second is considered for reintegration into the first if there is relevance.

It is these four elements and their connections were relevance is decided. What can not be described in this proposal is the results of such a system when fully aggregated. Each of the four elements can branch to create new vectors of research growing the knowledge base exponentially. The aggregation of curated research will take on its own significance, one that will only be apparent in retrospect.

5.0 Conclusion

The systems tests and proposed system elements examine how to integrate scientific research, product innovation and deep knowledge within associated, allied and external disciplines into a design process which tends to ignore, eliminate or discount this knowledge as a relevant design factor . The analysis and subsequent proposals also speculate the start of a culture of transferable and repeatable knowledge in the architectural design community. So often, design is treated with the Hegelian legacy of genius and intuition, with each architectural designer seeking inspiration from personal sources, rather than drawing from a deep database of social, cultural and technical predictability premised on integrated technical components within an aesthetic and formal system. Now the cultural and technological conditions have allied for architectural research to become less introverted. The opportunity exists for a new design culture, one rooted in research, openness, innovation and transferable knowledge, which places the advancement of the profession over oneself.

Bibliography

Bird, Christian, David Pattison, Raissa D'Souza, Vladimir Filkov and Premkumar Devanbu. 2008. Latent social structure in open source projects. In *SIGSOFT'08/FSE-16: Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*. New York,: ACM, 24–35.

Bird, Patterson, D'Souza, Filkov and Devanbu. *Latent Social Structure in Open Source projects*,

Crowston, K. and Howison, J. 2004. The Social Structure of Free and Open Source Software. *First Monday*, 10(2), 2005.

Henderson, Rebecca M. And Kim B. Clark. 1990. Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly* 35 (1): 9-30.

Hillier, William. 1996. *Space is the Machine*. Cambridge: Cambridge University Press

Howe, Jeff. *The Rise of Crowdsourcing*. 2006. Wired 14.06, http://www.wired.com/wired/archive/14.06/crowds.html, retrieved 01/29/10

Jun, Seung Joon. and Barnett, George. 2009. The Structure of Open Source Software: A Network Analysis of Open Source Software Project. Paper presented at the annual meeting of the *International Communication Association*, Sheraton New York, New York City, NY, <Not Available>. 2009-05-25. <http://www.allacademic.com/meta/p14715_index.html>

Keller, Rudi. 1994. *On Language Change: The invisible hand in language.* New York: Routledge.

Krogh, Georg von, Sebastian Spaeth, and Karim R. Lakhani, February 2003. *Community, Joining, and Specialization in Open Source Software Innovation: A Case Study*. MIT Sloan Working Paper No. 4413-03,.

Lakhani, Karim R. and Eric von Hippel. 2002. How open source software works: “free” user-to-user assistance. *Research Policy* 32, 923-943.

Lakhani, Karim R. and Jill A. Panetta. 2007. The Principles of Distributed Innovation. *Innovations: Technology, Governance, Globalization* 2, (3): 97–112.

Lakhani, Karim R., Lars Bo Jeppesen, Peter A. Lohse, and Jill A. Panetta. 2007. *The value of openness in scientific problem solving*. HBS Working Paper Number: 07-050. Cambridge: Harvard University.

Linstone, Harold A. and Murray Turoff, Editors.1975. *The Delphi Method: Techniques and Applications*, Linstone & Turoff.

Merton, R.K. *The Sociology of Science: Theoretical and empirical investigation*. Chicago, University of Chicago Press, [1942] 1973.

Raymond, Eric Steven. 2001. *The Cathedral and the Bazaar*. Sebastopol: O'Reilly Media.

Stevens, James, Philip Plowright and Anirban Adhya. 2009. Rethinking Models of Architectural Research: We don't do objects. Paper presented at ARCC 2009 - Leadership in Architectural Research, between academia and the profession, San Antonio, TX, 15-18 April 2009.

Van Den Bergh, Nan. Renaming: Vehicle for Empowerment. In Penfield, Joyce. 1987. *Women and Language in Transition*. Albany: SUNY Press: 130-136.

von Hippel, Eric and Georg von Krogh. 2003. Open source software and the “private-collective” innovation model: Issues for organization science. *Organizational Science* 32 (2): 209–233.