Collective Intelligence and Design Thinking

Paul Murty
University of Sydney, Sydney, Australia

Mercedes Paulini
University of Sydney, Sydney, Australia

Mary Lou Maher
University of Sydney, Sydney, Australia

Abstract

Collective intelligence (or CI) has recently emerged as a potential magnifier of design thinking. A surge of internet based social computing applications are achieving surprising results from people thinking collectively, without the aid or restrictions of formal organisation, supervision, or even payment in the conventional sense. Some of the best known applications, such as Threadless and Top Coder involve design activity. However, applying collective intelligence to more complex forms of designing appears likely to require greater understanding of both collective intelligence and design thinking. This paper considers three questions whose resolution may lead to a more general understanding of design thinking through the lens of collective intelligence; 1) how existing CI applications contribute to design thinking, 2) requirements for Collective intelligence for designing (or CID), and 3) how to support design processes in a CID environment. The authors conclude that existing CI applications are already developing innovative design thinking activities, that there are additional undiscovered ways of applying collective intelligence to designing, and that design activity in CID applications can be largely self organising.

1. Introduction

Collective intelligence (or CI) is an emergent phenomenon that has long existed and evolved in human cultures. It can emerge spontaneously, or be induced, and develop in many forms and settings. Facilitated by Internet technology, CI has emerged as a powerful, economical, human resource. In drawing upon and representing the cognitive contributions of many people, CI offers a potential to magnify design cognition in ways analogous to a functioning brain, drawing upon many neurons.

In this paper, we use the term CI to refer to the phenomena associated with internet-based applications that allow anyone to contribute, in the spirit of the ideas described in "Here Comes Everybody" (Shirky, 2009). Some of the best known internet based CI applications, (often referred to as crowdsourcing) including Wikipedia, Threadless and Top Coder, involve hundreds of thousands of participants interacting, collaborating, or competing with one another. The large numbers are indicators of popularity, success and processing scale, however numbers alone do not explain why or how these applications grew and now function so successfully. Many grew rapidly from modest beginnings. Significant bandwagon effects, evident now, came later. Success stems from the quality of, and personal gain from, the interaction, collaboration and competition, that people experience with one another and with the environment of the application.

New CI applications with fundamentally different approaches, literally new paradigms, are emerging continually. Therefore, the potential contribution of CI to designing and how to achieve it most effectively is an evolving topic.

In an earlier paper (Maher, Paulini, and Murty, 2010), we examined three dimensions of a conceptual space for computer-supported collective design. 1) Representation: technologies that provide shared digital representations of the design artifact, 2) Communication: technologies to support communication and collaboration and 3) Motivation: principles, incentives and structures that motivate designers and others to participate in collective design. Following an analysis of six CI applications the study found that successful CI attracts and facilitates participation from individuals who are intrinsically motivated to participate, for deeper personal reasons than financial reward, career or social advantage. An associated finding was that the wider the spectrum of motivational factors supported by the system, the more likely the application is to succeed and produce useful outputs.

This paper takes a step forward to consider issues and questions related to design thinking including; how existing CI applications contribute to design thinking, requirements for CID, and principles for organising a CI environment to support design processes.

2. How existing CI applications contribute to design thinking

Few systems based on CI occupy the design domain. However, CI has been successfully utilised for activities associated with designing, notably, at the ideation and evaluation stages. We describe four: Threadless, Kasparov v Team World, Top Coder and Webcanvas.

Threadless is a web application that utilises crowdsourcing. Participants are encouraged to contribute T-shirt designs, which are voted for by the user community. The most popular designs are selected for manufacture. Voting results are made available to motivate and guide designers towards a winning design. Although no explicit collaboration occurs on the designs, Threadless supports a thriving community, with an engaged public discourse on the designs, which enables collaboration to occur without being directly affirmed in the specification of each design.

Threadless relies on its community at three key places in the design cycle. At the conceptualisation phase, designs are generated by the community, for free. At the evaluation phase, the community votes for their favourite designs, providing the company with free market analysis. As a result of this kind of crowdsourcing, the community acts as designers, clients, and the potential market, ultimately purchasing the products.

Two activities commonly associated with designing are lateral thinking and problem-solving. A chess game was played over the Internet in 1999, between Gary Kasparov the reigning world chess champion, at the time and Team World, a diverse assembly of five consulting chess champions, chess clubs distributed internationally, many thousands of amateur players and strong chess analysis software. Chess, with its clearly defined rules provides a highly structured environment where participants need to assess various complex scenarios, to choose the strongest move. Design strategies were invoked during gameplay to conceptualise solutions and problem-solve. The collaboration and internal competition of the community, coupled with the computed aggregation of their ideas ensured that each move was formidable. One unprecedented strong move was made against Kasparov. Unfortunately, a breakdown in communication resulted in one uninformed move, ultimately leading to Kasparov's victory. Kasparov later affirmed the significant of the game, stating: “The sheer number of ideas, the complexity, and the contribution it has made to chess make it the most important game ever played.” This was all possible through the collective efforts of a diverse community.

Top Coder is an intriguing example of how a commercially successful website, combining crowdsourcing, and competition can be applied across many aspects of a complex design process. Top Coder provides coding solutions to software design problems, which it presents to its community as individual design challenges. The ultimate product is typically a synthesis of individual solution modules. The site offers a choice of coding tasks and encourages coders to compete for prize money, status and the privileges of being a top coder. Incentives are designed to ensure individuals are motivated to participate, whatever their level of knowledge or experience. Detailed coder statistics and rankings are displayed. Support is provided in the forum pages, which also serve as a platform for socialising and distributing the group's intelligence. Coders can also post a project for others to complete. The group is hierarchical and appears to embody graduated levels of collaboration. Coding is an ideal design task for CI, as it has clear inputs and outputs, a well-defined process in the middle, and results are quantifiable. However a more recent venture, Top Coder Studio, extends the Top Coder business model to logo design, web design, print design, and idea generation.

Webcanvas is a shared sketching application, comprising an online wall, on which people doodle. The wall has tremendous zooming capabilities, allowing small spaces to be enlarged and filled. Webcanvas has no business model. It aspires to be an ongoing collaborative artwork and a verification of the extent to which an open community can produce a work that is sympathetic and responsive to the individual efforts of its members. In a design scenario, open-ended shared representational tools like Webcanvas could host group sketching, supporting design analyses and conceptualisation. Webcanvas is an example of self-organising design activity with no direct attribution to individual designers. Technologies such as Webcanvas can lead to changes in the way a design task is approached and the kinds of discoveries that may be made.

Existing software applications can be appropriated as tools to support CI. Virtual worlds, online forums, and community sketching sites, originally created for different purposes, can be used in a more focused way for collaborative designing. Not all CI applications provide platforms for their communities. Some, like the I Love Bees project (McGonigal 2008) simply publish data in social networking sites and allow the community to find their own resources to support collaboration. Success for these systems relies heavily upon a deep understanding of the unique requirements of the target community, as well as flexibility and adaptive responsiveness to changing needs.

These systems generally, embody ways of utilising the attributes of collective intelligence to achieve key objectives in the design process, such as ideation and evaluation, within a variety of business models. Of equal importance is their ability to support individual interests and to maintain active communities, through effective incentives, thereby achieving stability and sustainability.

3. Requirements for CID

We propose that CID developers should draw upon knowledge of both; 1) requirements of computer support for designing, by individuals and groups, and 2) existing CI models. These areas contribute complementary models and support environments for collective design. In an earlier paper (Maher, Paulini, and Murty 2010), we developed a conceptual space for understanding CI that includes three sets of requirements: communication, representation, and motivation. They characterize successful CI applications in terms of how internet technologies satisfy these requirements as a guide for developing successful CID applications. In this paper, two additional requirements, guidance and self organisation are also introduced.

3.1 Communication

Effective communications, including shared representations across multiple platforms, play a key role in developing concepts and providing design commentary. CID applications need to be communication-rich and diverse, supporting both synchronous and asynchronous modes, direct, and indirect communications, multiple content types and high speed connections.

3.2 Representation

CID applications are more likely, than individual or collaborative scenarios, to require multiple shared representations, to achieve a shared understanding and to support visualization, analysis and synthesis, among a large diverse population. Representation media includes voice, text, sketches, 2D or 3D models, immersive virtual environments.

3.3 Motivation

The success of collective intelligence applications relies on motivated people. It is important to invoke, build and reinforce motivation and also to not demotivate. Key motivating objectives are, to attract, welcome, intrigue, challenge, encourage and reward participation. Extrinsic motivators such as recognition, social opportunities, career and material rewards are also associated with many CI applications. However it is also likely and advantageous that many CI participants be influenced by intrinsic motivations such as ideology, challenge, or fun. Intrinsic motivation is valuable for its durability and its association with creativity (Csíkszentmihályi 1998). A likely general rule is that, for most advanced or most designerly applications, the more motivators the better.

3.4 Guidance

Guidance is a both motivator and a practical necessity. A variety of guidance modes are required, eg. inform, orient, respond, elicit. User interfaces require flexibility, to match different levels of familiarity and use patterns, and to grow with the community.

3.5 Self organisation

In any organisation, economies in performance and of scale can be achieved by controlled devolvement of micro-management to lower level participants. This is particularly true of CI applications. While they are typically controlled from the top, self-organization is the predominant mode of management at the crowd scale. Achieving self organization appears to include two considerations: 1) individual and collective agency for low level tasks and 2) negotiated collective agency for high level wholistic decisions. Opportunities to incorporate self organisation may be increased if higher levels of intrinsic motivation can be achieved.

4. Supporting design processes in a CID Environment

Studies of designers have identified that conceptual design settings, or situations which require design intervention, have common properties. They are characterised by ill-defined or wicked problems that are not soluble simply by collecting and synthesing information. Instead designing requires interpretation, or pre-structuring of situations. Design proceeds by a parallel or iterative, counter-play, of conjecture and a variety of other acts, or processes that precede and follow, in which solutions and problems tend to emerge and develop together. Often what is vital only becomes evident when designing takes place Cross (1999). This dynamic has been variously interpreted by different theorists as argumentation (Rittel 1972), a negotiation (Lawson 1997), and a reflective conversation (Schön 1983). Darke (1979) observed that the conjectures of expert designers were derived from particular ideas, interpretations, or pre-structures, she referred to as primary generators. For this discussion, designing is portrayed as a conjectural process in which: 1) Conjectures emerge from generators, exploration and/or discoveries, 2) Conjectures influence generators, exploration and discovery, as shown in Figure 1.

The quality of design from a CI application, like more conventional designing, may depend on successful facilitation of conjectural (generate and test) processes undertaken by individuals acting alone, or with others, subject to the properties or rules of the CI application structure.

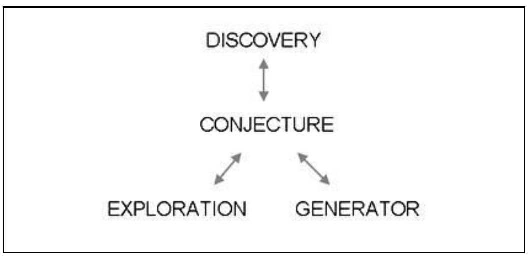

Figure 1. Design Conjecture model (Murty 2009)

4.1 Conjecture

Design conjecturing typically means putting forward a possible solution or approach which can be checked or tested against the design requirements. Conjectures may range from bold generalisations to tentative first thoughts. Testing methods and criteria may be the outcome of a similar conjectural process, conducted earlier or in parallel, with or without the capacity to "learn" from experience. In Threadless, designs are submitted as propositions, by each designer. In Top Coder, a software task is decomposed into modules and coding propositions for each module are also put forward by individual coders. A CID system needs to provide an interactive environment which assists and motivates its user population to conceive, propose, recognise and evaluate conjectures.

4.2 Generator

A generator can be viewed as a particular class of conjecture which conceptualises a stage or aspect of a design or its situation, providing a basis for further conjectures. Darke (1979) proposes a key effect of a powerful or "primary generator" is to reduce the range of possible solutions, thereby simplifying the problem. The designer of a housing project, for example, may propose a tower, row house or walk-up apartment configuration as the primary generator. Further understanding of the design can be gained from testing the generator, by deriving and testing further conjectures from it. The row house solution may be tested by generating two alternative site arrangements, such as building along the site contours or stepping across them.

4.3 Exploration

Exploring, in the context of designing means investigating or searching requirements, potential design generators and conjectural solutions in a developing design space. Design exploration typically involves a combination of physical and cognitive activities, such as modeling, analysing, experiencing, reflecting and discussing. Individual and group exploratory sessions in many design domains may involve sketching and verbal descriptions of alternatives. Given the situated nature of designing one act may lead to and inform another, in almost any order.

An individual engaged in designing, either solo or as a collaborator, perceiving the transactions of designing directly, may experience the succession of events with little difficulty; no less normal than driving in traffic perhaps. Frequently, collaborations involve people in mutually supportive roles, such as leader, note taker, assistant, etc. When managed well, this structuring reduces the cognitive load and the workload per individual and improves coherence and predictability, enabling individuals to concentrate on what they do best and thereby achieve more.

As CID can not replicate the directness and convenience of an organised group of colleagues, working together in the same room, a CID system needs to excel in other ways, by supporting multiple levels of parallel explorations. There may be thousands of participants exploring simultaneously, individually, in collaboration, or as part of a crowd source. In addition the parallelism is multidimensional. There is duplication and there will be different start and end times and conditions, different subjects and different findings.

To produce coherent information which guides and motivates participants, from many separate processes, we propose that the CID system must be capable of supporting: 1) a broad range of shared representation types and alternatives, 2) interaction among participants, proposing and testing generators and conjectures, 3) reporting and dissemination of knowledge, and 4) self organisation and reorganisation. CID applications aiming to tackle more complex design tasks, than (say) Threadless and Top Coder, may require greater communication-richness and diversity than these limited applications.

4.4 Discovery

A discovery is typically an unexpected and novel experience. It may, for example, occur in the form of a new awareness, understanding, recognition or an idea. Individuals make discoveries in many unexpected places and ways, whether working alone or with others; and they make different kinds of discoveries. The significance of different kinds of discoveries, here, is not so much their features, but rather their effects in a design setting. In contrast to many CI applications, distributing many separate tasks among participants, design thinking can require a wholistic sensibility in addition to attention to detail. The effects of some discoveries may be relatively trivial, but a more revelatory experience, may go right to the core of a design. In a CID application, involving many parallel processes, this possibility indicates that a form of part-to-whole attention facility may be required.

5. Conclusions

Three aspects of CID have been considered in this paper:

- How existing CI applications contribute to design thinking,
- Requirements for CID, and
- Support for design processes in a CID environment.

The existing CI applications are revealing in different ways. Threadless began as a simple graphic design application, but from it there has emerged a thriving community engaged in public discourse, without explicit application support. Kasparov v Team World was not explicitly a design application but it involved many thousands of people in complex strategic thinking and aggregation of ideas and voluntary collaborations leading to at least one new powerful move. Top Coder provides solutions to software design problems, and commercial success has led to its recently entering other design areas, unrelated to computer software. These few examples are sufficient to demonstrate an important point, that an emerging collective intelligence is not limited by the scope envisaged when the application was created.

Successful CI requires an active motivated participant population. Applications need to be communication rich in order to facilitate the interchange of information, development of concepts. Multiple, shared representations are required to achieve shared understanding, and facilitate designing. The importance of motivation was stressed and a range of motivation objectives were identified. The importance of intrinsic motivation was further highlighted. So too was the observation that, for the most advanced, or most designerly applications, the more motivators the system supports the more likely it is to become successful. Guidance and the enabling of self organisation were also described.

A CID environment, supports conjectural design processes, including conjecture, generators, exploration and discovery. A CID system provides an interactive connective environment which motivates its user population and facilitates and manages multiple simultaneous collective design processes. Successful undertakings, such as Wikipedia and Top Coder have achieved remarkable successes in supporting numerous parallel participant processes. These two systems, an online encyclopedia and a software producer, are entirely different in many respects. But their common features include: coherent sub-dividable tasks, a strong central control group to give structure to new initiatives, explicitly expressed rule based organisation, welcoming culture, readily available training information, permission to initiate tasks, few or no directions towards or away from particular tasks, directions on how to do things focusing on operational matters, rather than content, and structured peer review or voting procedures. In general, the need for top down management is minimised and the concept of self-organisation is promoted.

Whether collective groups are more or less prone to problems associated with informal hierarchies and critical situations associated with collaborative groups is a topic of research in progress (Badke-Schaub & Frankenberger, 1999). One strength of crowd sourcing is that effects of unwanted inputs can be moderated by a variety of filtering controls, peer reviews and voting and by having large numbers of participants contributing. Problems at the top level, among off-line executives who may overrule the collective, could be more problematic for CI applications than for other organisations, given the great importance of intrinsic motivation among the on-line participants. Transparency of communication across levels can minimise misapplied authority and help sustain necessary participant motivation. But ultimately, applications that fail to master the basics of CI, or lose that mastery, will not survive and many others, with better business models, will replace them.

Finally, one may ask; can CID really lead to new design thinking activities and strategies? We venture a confident "yes" to both prospects. We can see supporting evidence already, in Top Coder and Threadless. Moreover, the presence of CI on the Internet is an instructive example of itself, in the sense of being a carrier of collective intelligence, about collective intelligence. The growing stream of new and diverse CI applications demonstrates an abundance of previously undiscovered ways of appreciating and applying the considerable intelligence, and willingness to think, of many people worldwide. Are new design thinking strategies possible in CID? Could cell phones do more than make calls? Be prepared for surprises as the technology of CID unfolds and evolves.

References

Badke-Schaub, P. and Frankenberger, E. 1999, 'Analysis of design projects', *Design Studies,* 20, 5, pp 465-480.

Csíkszentmihályi, M. 1998, *Finding Flow: The Psychology of Engagement With Everyday Life*, Basic Books, New York.

Cross, N. 1999, 'Natural intelligence in design', *Design Studies,* 20, pp 25-39.

Darke, J. 1979, 'The primary generator and the design process', *Design Studies*, 1, 1, pp 36-44.

Lawson, B. R. 1997*, How Designers Think,* Architectural Press, Oxford.

Maher, M.L., Paulini, M. and Murty, P. 2010, 'Scaling up: From individual design to collaborative design to collective design', *Design Computing and Cognition '10*, Stuttgart, Germany. 10-14 July.

McGonigal, J: 2008, Why I love bees: A case style in collective intelligence gaming, in K Salen (ed) *The Ecology of Games: Connecting Youth, Games, and Learning*, Cambridge, MA pp. 199-228.

Murty, P. 2009, 'Comparing paradigms with practice: The design conjecture cycle', in N. Gu, M. J. Ostwald and A. Williams (eds), *Computing, Cognition and Education: Recent Research in the Architectural Sciences*, ANZAScA, Newcastle and Sydney, pp. 93-110.

Rittel, H.W.J. 1972, 'On the Planning Crisis: Systems Analysis of the 'First and Second Generations', *Bedriftsokonomen,* Nr. 8, pp 390-396.

Schön, D.A. 1983, *The Reflective Practitioner.* New York, Basic Books.

Shirky, C. 2009, *Here Comes Everybody: The Power of Organizing Without Organizations*, Penguin Group.