Programming the Global Brain

Programming the Global Brain

Abraham Bernstein,[a] Mark Klein,[b] and Thomas W. Malone [b]

MIT Center for Collective Intelligence

MIT Center for Collective Intelligence Working Paper No. 2011-04

To appear in the Communications of the ACM May 2012, Vol. 55, Issue 5

November 2011

MIT Center for Collective Intelligence
Massachusetts Institute of Technology
http://cci.mit.edu

[a] Department of Informatics, University of Zurich, Zurich, Switzerland, bernstein@ifi.uzh.chn
[b] MIT – Sloan School of Management, Cambridge, MA, m_klein@mit.edu & malone@mit.edu

Programming the Global Brain

Abraham Bernstein,[a] Mark Klein,[b] and Thomas W. Malone[b]

New ways of combining networked humans and computers—whether they are called collective intelligence, social computing, or various other terms—are already important and likely to become truly transformative in domains from education and industry to government and the arts. These systems are now routinely able to solve problems that would have been unthinkably difficult only a few short years ago, combining the communication and number-crunching capabilities of computer systems with the creativity and high-level cognitive capabilities of people. And all this is super-charged by the emergent generativity and robust evaluation that can come from the many eyes of large crowds of people. As the scale, scope, and connectivity of these human-computer networks increase, we believe it will become increasingly useful to view all the people and computers on our planet as constituting a kind of "global brain."

We still only poorly understand, however, how to "program" this global brain. We have some stunning success stories (e.g. Wikipedia, Google), but most applications still fail, or require a long series of trial-and-error refinements, for reasons we only poorly understand. Because people are involved, programming the global brain is deeply different from programming traditional computers. In this article, we will consider this challenge, exploring the nature of these differences as well as issuing a call-to-arms describing open research challenges that need to be met in order to more fully exploit the enormous potential of the global brain.

What Makes the Global Brain Different?

There are already literally hundreds of compelling examples of the global brain at work, collectively representing the contributions of many millions of people and computers[1,2]. These range from systems where individuals perform simple micro-tasks (mturk.com) to where they compete to solve complex engineering problems (innocentive.com). Some systems harness the "wisdom of crowds" in contexts that range from "citizen science"[3] (fold.it, galaxyzoo.org) to predicting box office performance (hsx.com). Open idea ecologies (where crowds of people share, recombine, and refine each other's creative outputs) have produced remarkable results for everything from videos (youtube.com) and encyclopedias (wikipedia.org) to software (Linux). Systems have been developed that look for individual task-focused "geniuses" (marketocracy.com) or, conversely, datamine the activity traces of millions of Internet users (Google's search engine). While these systems cover an enormous range of approaches, it has become clear from these experiences that programming the global brain is different from programming traditional computers in some fundamental ways. We list some of the most important differences below:

> *Motivational diversity*: People, unlike current computational systems, are self-interested and therefore require appropriate incentives—anything from money, fame, and fun to altruism and community—to perform tasks. These incentives have to be carefully designed, moreover, to avoid people gaming the system or causing outright damage. In some cases, one may even use their motivation to do one task to accomplish another, as in reCAPTCHA, where people OCR documents as a side effect of passing a human versus bot test.[4]
>
> *Cognitive diversity*: In most computer systems we deal with a limited range of diversity – in terms of memory, speed, and device access. People, by contrast, vary across many dimensions in the kinds of

[a] Department of Informatics, University of Zurich, Zurich, Switzerland, bernstein@ifi.uzh.ch
[b] MIT – Sloan School of Management, Cambridge, MA, m_klein@mit.edu & malone@mit.edu

tasks they can do well, and their individual strengths are only incompletely understood at best. This implies qualitative differences in how (and how well) we can expect to match tasks and resources in a global brain context.

Error diversity: With traditional computers, we worry much more about outright failure than other kinds of errors. And the other errors are usually highly deterministic and limited in diversity because a relatively small range of software is typically replicated across millions of computers. People, by contrast, are prone to a bewildering and inconsistent variety of idiosyncratic deviations from rational and accurate performance. The global brain, therefore, calls for a radically more capable quality assurance oriented towards the particular kinds of errors that occur with human participants. Fortunately, the global brain also provides access, at least currently, to a huge human "cognitive surplus",[5] so that, for instance, quality mechanisms based on previously-unthinkable levels of redundancy have become practical[2].

These attributes lead, in turn, to the possibility of new, and potentially troubling, forms of *emergence*. Crowds of people, when engaged in solving interdependent problems, can evince emergent behaviors that range from groupthink (where decision-makers converge prematurely on a small subset of the solution space) to balkanization (where decision-makers divide into intransigent competing cliques) to chaotic dynamics (e.g., stock market bubbles and crashes). While emergence is, of course, not unique to the global brain, it is probably made much more challenging by the unprecedented combination of microsecond computer and communications speeds, globe-scale interdependencies, and human diversity.

The Need for New Programming Metaphors

How, then, can we effectively program a global brain, characterized as it is by unique challenges (and opportunities)? We believe that a fundamental requirement is developing powerful new programming metaphors that more accurately reflect the ways people and computers can work together in the global brain. For instance, today's innovative collective intelligence systems embody a set of common design patterns[6], including *collections* (where people create independent items, such as YouTube videos), *collaborations* (where people create interdependent items, such as Linux modules), and various kinds of *group* and *individual decisions* (such as *voting, averaging, social networks,* and *markets*). These design patterns, in turn, can be embodied in various programming metaphors, such as:

- *An idea ecology* – The global brain can host a constant ferment of idea generation, mutation, recombination, and selection, analogous to biological evolution. In this context, programming consists of soliciting collections of items and specifying a fitness function for choosing among them. Interesting examples of this include the MATLAB open programming contests (for software) and YouTube (for videos).
- *A web of dependencies* – Many important problems (such as product, process, and policy definition) can be viewed as collaborations, where multiple diverse agents try to solve interdependent pieces of a larger problem. The global brain can detect when conflicts appear between sub-solutions, as well as guide agents towards a globally consistent result. In this context, programming includes defining the task decomposition and specifying ways of managing the interdependencies among sub-problems. Early examples of this include virtual mockups such as the Digital Pre-assembly system Boeing used in the design of the 777 aircraft.
- *An intellectual supply chain* – For some problems, we can view the global brain as a supply chain, where a sequence of tasks and information flows among people and machines can be specified in advance. In this context, programming can be defined in terms already familiar to computer scientists as processes and dataflows. Interesting examples of this idea include the Turkit[7] and Crowdforge[8] systems, which have been applied to such tasks as writing and editing articles.

- *A collaborative deliberation* – The global brain can also be used to enact decision processes where people and software systems select issues to consider, enumerate and critique solution alternatives, and then choose some subset of these solutions. In this context, programming can be viewed as defining the rules for identifying issues and enumerating, critiquing, and selecting solutions.
- *A radically fluid virtual organization* – Sometimes it's useful to view the global brain as a collection of evanescent virtual organizations, which rapidly coalesce, perform, and disband in light-speed open markets. In this context, programming includes identifying the task requirements and selecting among the organizations that are offering to perform the task. Interesting examples of this idea include odesk.com and elance.com.
- *A multi-user game* – Many tasks can be presented as a multi-user game, where useful outcomes are achieved as a result, sometimes unintentional, of playing the game. In this context, programming consists of specifying the rules and incentives for game play. Interesting examples of this include fold.it and the Google Image Labeller.

Making such global brain programming metaphors a reality will, in turn, require progress along several fronts:

Creating "social operating systems": An operating system (OS), in the context of a single computer, manages the allocation of hardware and software resources such as memory, CPU time, disk space, and input/output devices. A social operating system, in addition to doing all these things, will also have to manage the mustering and allocation of human resources to tasks. This will require fast, robust infrastructures for contracts, payments, or other motivational elements, as well as scalable task-to-resource matchmaking such as markets. These will be challenging problems because people (unlike hardware resources) are diverse in all the ways described above. But providing easy-to-use solutions for the problems of finding and motivating human participants—rather than requiring each system developer to solve this problem individually—will greatly facilitate programming the global brain.

Defining new programming languages: Conventional programming languages are, out of necessity, fully prescriptive, describing the algorithms to be executed in exhaustive detail. Such languages are often not a good match, however, for specifying tasks with human participants. The programming languages for the global brain will, therefore, need to support a "specificity frontier" of varying degrees of detail in task definition[9]. One end of this frontier involves defining programs that allocate highly specific micro-tasks to people and link them into larger workflows. In the middle ground, we may use constraint-based programs, which specify (e.g., in a game setting) the goals as well as the *limits* on how they can be achieved, but not *how* they should be achieved. At the far end of the specificity frontier, programming may be limited to simply stating incomplete goal specifications. Additionally, we need to expand the range of abstractions such programming languages offer. While traditional programming languages incorporate constructs such as loops and recursion, a global brain programming language may also need to incorporate abstractions such as group decision processes, contests, and collaborative steps.[10]

Promulgating new software engineering skills: Programmers will need to develop new mindsets about, for example, such basic concepts as what a "program" is and what "testing" and "debugging" mean. They will need to become not just software architects and algorithm implementers, but also *organizational* or even *societal* architects able to think systematically about things like motivations, coalitions, emergence, and so on. Perhaps most fundamentally, they will need transition from a purely "command-and-control" perspective for organizing people to one oriented around cultivating and coordinating[11] societies made up of many diverse independent players.

A Call to Arms

We have attempted to identify, in this short article, some of the key challenges, opportunities, and strategies involved in programming the emerging global brain. Learning to do this well is, perhaps, even more urgent than many people realize. Our world is faced with both existential threats of unprecedented seriousness (such as the environment) and huge opportunities (such as for scientific and social progress). We believe that our ability to face the threats and opportunities of the coming century will be profoundly affected by how well, and soon, we can master the art of programming our planet's emerging global brain.

---

[1] See, for example, Doan, A., Ramakrishnan, R., & Halevy, A. (2011) Crowdsourcing systems on the World-Wide Web. Communications of the ACM, **54**, 4, 86-96.

[2] Quinn, A. J., & Bederson, B. B. (2011) Human Computation: A Survey and Taxonomy of a Growing Field, *CHI 2011*, May 7–12, 2011, Vancouver, BC, Canada.

[3] Hand, E. (2010). Citizen science: People power. *Nature*, 466, 685-687.

[4] von Ahn, L., Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008) reCAPTCHA: Human-based character recognition via Web security Measures. *Science*, 321, 1465-1468

[5] Shirky, C. (2010) Cognitive Surplus: *Creativity and Generosity in a Connected Age.* New York, Penguin Press

[6] Malone, T. W., Laubacher, R., & Dellarocas, C. (2010) The Collective Intelligence Genome, *Sloan Management Review*, Spring 2010, *51*, 3, 21-31.

[7] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller (2010). TurKit: human computation algorithms on mechanical turk. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology (UIST '10)*. ACM, New York, NY, USA, 57-66. DOI=10.1145/1866029.1866040 http://doi.acm.org/10.1145/1866029.1866040

[8] Kittur, A.; Smus, B.; and Kraut, R. (2011). Crowdforge: Crowdsourcing complex work. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'11)*. ACM, New York, NY, USA

[9] Abraham Bernstein (2000), How can cooperative work tools support dynamic group processes? Bridging the specificity frontier, In *Proceedings of the International Conference on Computer Supported Cooperative Work (CSCW'2000)* 2000, ACM.

[10] Patrick Minder and Abraham Bernstein (2011). CrowdLang - First Steps Towards Programmable Human Computers for General Computation. In *Proceedings of the 3rd Human Computation Workshop (HCOMP 2011)*, AAAI-Press

[11] Malone, T. W. (2004) *The Future of Work*. Boston, MA: Harvard Business School Press.