

Social media has become globally ubiquitous, transforming how people are networked and mobilized. This forum explores research and applications of these new networked publics at individual, organizational, and societal levels.

— Shelly Farnham, Editor

Computer Supported Collective Action

Aaron Shaw, Northwestern University, Haoqi Zhang, Northwestern University, Andrés Monroy-Hernández, Microsoft Research, Sean Munson, University of Washington, Benjamin Mako Hill, University of Washington, Elizabeth Gerber, Northwestern University, Peter Kinnaird, Carnegie Mellon University, Patrick Minder, University of Zurich

Social computing systems play an increasingly important role in helping individuals take collective action—that is, action taken by multiple people in pursuit of the same goal or collective good [1]. Successful *computer supported collective action* (CSCA) connects crowds and communities of participants, lowers the cost of communication and deliberation, and helps to coordinate action, thereby enabling new forms of collaboration unimaginable a decade ago [2,3].

Despite the many successful and failed deployments of CSCA systems, there has been little systematic thinking about CSCA as a process or an integrated design problem. Existing systems focus on supporting particular parts of the collective action process, and there are few examples of systems that support *participatory and end-to-end collective action*, in which a crowd or community comes together to formulate goals and objectives, brainstorm ideas for action, form plans, and mobilize a critical mass of participants to take action.

This broader CSCA process perspective suggests that systems can fail not only *within* stages and systems of collective action but also at failure points *between* stages and systems. These transitional failure points offer a major opportunity for researchers on sociotechnical systems and designers to contribute novel integrative approaches to these problems.

PATTERNS IN CSCA

We propose five stages or “patterns” that recur across a number of CSCA examples and that computing systems attempting to support end-to-end collective action ought to address. The design of systems for collective action—much like architecture, urban planning, or software development—can benefit from the adoption of such patterns as templates for approaching recurrent problems and solutions. Figure 1 represents these patterns as a sequence of stages. In the gaps between the patterns we summarize the types of problems that are likely to arise, which can help designers approach these questions.

Pattern 1. Identify a problem and others who care about it.

Very often, the first step of any form of collective action involves identifying a salient issue or concern that affects a public of some sort. Because this process can be very difficult and sensitive (i.e., Whose problems are salient? Why do other people not agree that my problems matter?), a CSCA system and system designer can play a critical role in

establishing and supporting trust, facilitating meaningful discussion, encouraging active listening, and valorizing good-faith participation.

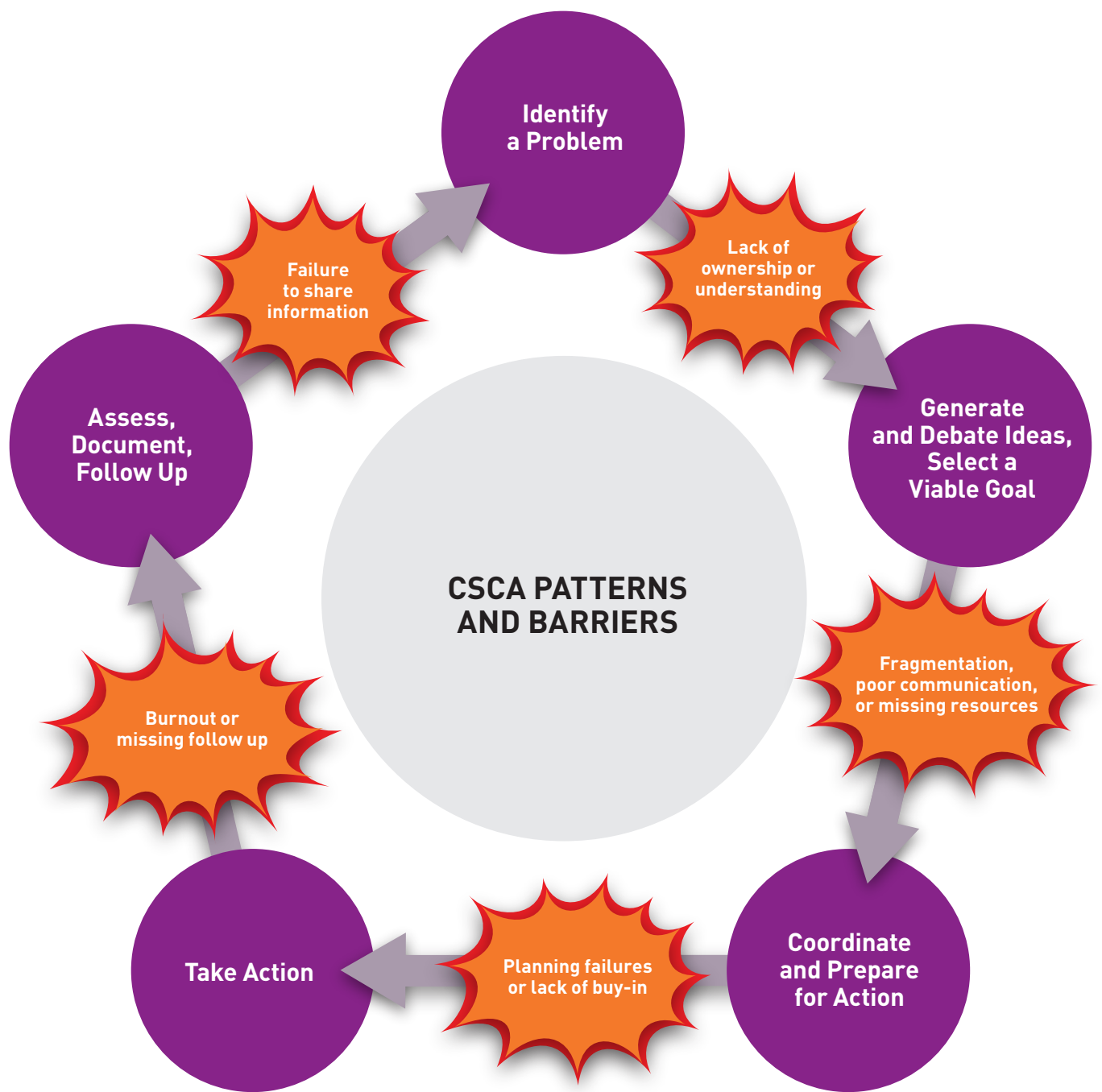
Systems that support CSCA at this nascent stage are ultimately those that encourage connection, expression, and listening. Social media tools like Facebook, Twitter, or Instagram can serve these purposes. Existing networks and ties are often the ideal foundation upon which social computing systems can facilitate more effective, asynchronous communication among groups that share geographic space, interests, or concerns.

Pattern 2. Generate, debate, and select viable solutions. Once an opportunity for action and a group interested in pursuing it have coalesced, the development and proposal of solutions becomes the priority. Effective CSCA systems can support the collection and substantive comparison of ideas, promote accessible discussion, and keep conversations moving toward actionable decisions.

Existing social computing systems that support this pattern tend to gather many ideas from multiple perspectives, structure ideation and deliberation, clarify opportunities for participation, create milestones, and facilitate transitions toward a decision. A number of existing tools support ideation and discussion in various forms; however, few support effective deliberation and discussion on the way toward the articulation of some goal. For example, the ConsiderIt

Insights

- Social computing systems that support solutions to collective action problems must, by luck or design, navigate complex, interdependent processes.
- Future research and design in CSCA should adopt an integrative perspective in order to facilitate end-to-end solutions and overcome gaps between existing tools.



Cross-Cutting Barriers:

- Marshalling resources
- Routing tasks and attention
- Lack of self or group efficacy
- Unclear transitions
- Aversion to revisit choices
- Lack of visibility
- Missing evaluation criteria

Figure 1. Computer supported collective action patterns shown as a sequence of stages. Gaps between the patterns show the types of problems that are likely to arise.

system supports contentious debate, but not necessarily deliberation toward some clear goal or objective [4]. AllOurIdeas supports a competition between crowdsourced proposals but does not offer a framework for sustained, nuanced dialogue along the way.

Even when some committed core or critical mass within the collective can identify a solution they wish to pursue, they must still find a way to convert this into action. Often voting or some other selection procedure fills this gap, providing a transitional state between ideation and action. In this way, the act of people selecting or voting in favor of a particular solution establishes that they have made a commitment to the group as well as to the chosen path of action.

Pattern 3. Coordinate and prepare for action. Once the goal has been identified, the work must be planned and the group must take the steps necessary to move toward it. Who takes the lead? What tasks are necessary? What milestones are there along the way and how will the participants know when they have reached them? Coordination and preparation for collective action involves understanding what work needs to be done, what skills are required to do it, who can and will do what, and in what timeframe the tasks must be done. CSCA systems that support these processes facilitate planning, recruitment, division of labor, coordination, supportive leadership, sustained motivation, tractable units of participation, and accountability.

The design principles that characterize successful approaches to this pattern echo characteristics of effective project management and team leadership. Generally, the creation of clear objectives, well-defined criteria for progress and evaluation, and open lines of communication provide the foundations for successful coordination and preparation. Providing reminders of the long-term objectives through social comparisons can also sustain the motivation of participants. In addition, the creation of trust among team members, the definition of

roles, and some mechanisms of accountability also tend to help projects navigate this process.

Systems that exemplify some of these attributes and solutions include classical project management tools such as Redmine. A number of online mobilization platforms and organizations have developed systems to support effective project planning and coordination, including Causes, Avaaz, and MoveOn.

Pattern 4. Take action. With a plan, people, and resources in place, it is time to act. Resources need to come together, and ongoing coordination is likely to be required to marshal resources effectively. Action often brings about new information, and some people may not show up, so the planned work design and assignment may have to change on the fly.

In this pattern, CSCA systems become particularly valuable when they facilitate responsive, low-latency, distributed communication that participants can adapt to suit unanticipated needs under variable conditions. Robust infrastructure also becomes more important at this stage than perhaps any other. Finally, in moments of action, people often rely on proven existing tools that they have experience and familiarity with, implying that the best technical systems for supporting action are usually not the newest or the most innovative.

Successful systems in this pattern tend to help participants understand and visualize their contribution to the collective goal. Similarly, in instances where actions involve collocation, effective organizing technologies can help leaders establish dynamic protocols for on-site communication and responsive contingency plans.

Once the goal has been identified, the work must be planned and the group must take the steps necessary to move toward it. Who takes the lead?

Mobile technologies, including SMS, Twitter, and other tools that take advantage of the speed and reliability of short, simple messaging, appear to be especially suitable for many such circumstances.

Pattern 5. Assess, document, and follow up. Following any effective or ineffective action, good follow-up provides a foundation for future work. Useful activities at this stage include assessment and documentation intended to record discussions and actions, so that people can revisit the project without having to repeat or reverse-engineer the activity and outcomes. Good documentation indicates areas that need further improvement or follow-on work—topics that participants and leaders can discuss together as they reflect on their experiences. Assessment and documentation also help participants celebrate success and learn from failures or unforeseen difficulties, enabling more effective future action.

While we list this as the last pattern, documentation should occur throughout the process. Social media tools that facilitate taking and sharing photos, recording audio, writing notes, and interviewing participants are all valuable to this objective. Depending on the type of action and the goals behind the documentation, participants might benefit from engaging in this process with their own mobile technology. Whenever possible, leaders and participants should attempt to plan the evaluation and documentation from the start and allocate sufficient resources for all of these activities.

INTERSECTIONS AND BARRIERS BETWEEN PATTERNS

The gaps *between* these components reflect areas neglected by previous analyses and design interventions, resulting in a large number of failures of collective action.

There are systems, for example, Reddit or Ubuntu's Brainstorm, that excel in supporting both ideation and the selection of ideas. Once a particular deliberation is done and a particular goal has been chosen, there is very limited technological help to transition *out* of this stage and *into*

the detailed planning and execution stages. Unless the initiator or another user of the system volunteers to take responsibility, nothing becomes of the goal created, and then vetted and selected, by the crowd. When such tools exist, they are often limited to problems with very specific pathways. For example, bug-tracking systems in code repositories (e.g., Github) support identification of a bug, more detailed description about the problem, discussion of potential solutions, and eventual contribution and integration of patches or rejection of the problem as a problem at all. For more open-ended problem and solution spaces, however, these tools break down.

Because there are strong dependencies between the patterns, there are many opportunities for errors that stem from the lack of a systematic perspective. For example, in the ideation stage, groups often lack clear evaluation criteria for ideas or knowledge of what resources will be necessary to complete the task. When a crowd suggests ideas, they may propose solutions that are impossible or that require resources that are unlikely to be available to the collective that will ultimately act. Because ideation is often so divorced from action, the result can be descriptions of ideas that are not clear or complete enough to be actionable. Overcoming issues like these across the many pieces in a collective action entails projecting each stage both backward and forward in the process, so as to incorporate considerations that might otherwise appear external. It also means moving forward even when uncertainty remains, as potential barriers cannot be resolved ahead of time, or moving backward—a direction in which people can be reluctant to move—to revisit an earlier stage when insurmountable barriers are reached. Social computing systems that support solutions to collective action problems will, by luck or design, take these complex processes into account.

Although designers are far from mastering the individual components of collective action, we have developed a number of good tools for different types of projects at different stages of action. However,

Although designers are far from mastering the individual components of collective action, we have developed a number of good tools for different types of projects at different stages of action.

we have devoted very little energy or expertise to the broader challenge of routing projects and helping them flow among this ecosystem of tools. In order to facilitate end-to-end CSCA, we suggest that it is important to build theories and systems around the gaps in the process of collective action shown in our model. Future work on CSCA should focus on lowering the costs associated with transitions between these patterns and stages.

The identification of a single set of patterns or systems that could comprehensively support participatory, general purpose, end-to-end collective action represents an ambitious goal. Existing tools—either the ones that address a particular pattern or the ones that support end-to-end action, but within a very specific domain—provide “first-order approximations” and must gradually be assembled into new, more effective and comprehensive configurations.

Many types of work can inform this line of inquiry, including theoretical elaboration, empirical observation, and system development. We hope to inspire members of the HCI community and beyond to contribute to these efforts.

ACKNOWLEDGMENTS

This article is the outcome of collaboration that began during CrowdCamp, a two-day hack-athon for crowdsourcing, human computation, social media, and

collective intelligence ideas held at the 2013 ACM Conference on Computer Supported Cooperative Work (CSCW) [5].

ENDNOTES

1. Marwell, G. and Oliver, P. *The Critical Mass in Collective Action: A Micro-Social Theory*. Cambridge University Press, 1993.
2. Benkler, Y. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, 2006.
3. Bennett, W.L. and Segerberg, A. *The Logic of Connective Action: Digital Media and the Personalization of Contentious Politics*. Cambridge University Press, 2013.
4. Kriplean, T., Morgan, J., Freelon, D., Borning, A., and Bennett, L. Supporting reflective public thought with ConsiderIt. *Proc. of the 2012 Conference on Computer Supported Cooperative Work*. ACM, New York, 2012.
5. Chilton, L., André, P., Bigham, J., Dontcheva, M., Gerber, E., and Gilbert, E. (2013). CrowdCamp 2013: Rapidly iterating crowd ideas. *Proc. of the 2013 Conference on Computer Supported Cooperative Work Companion*. ACM, New York, 2013, 313–314.

⬇️ **Aaron Shaw** is an assistant professor of communication studies at Northwestern University.

→ aaronshaw@northwestern.edu

⬇️ **Haoqi Zhang** is an assistant professor of electrical engineering and computer science at Northwestern University.

→ hq@northwestern.edu

⬇️ **Andrés Monroy-Hernández** is a researcher at Microsoft Research.

→ amh@microsoft.com

⬇️ **Sean Munson** is an assistant professor of human centered design and engineering at the University of Washington.

→ smunson@uw.edu

⬇️ **Benjamin Mako Hill** is an acting assistant professor of communication at the University of Washington.

→ makohill@uw.edu

⬇️ **Elizabeth Gerber** is an assistant professor of mechanical engineering and communication studies at Northwestern University.

→ egerber@northwestern.edu

⬇️ **Peter Kinnaird** is a Ph.D. candidate in at Carnegie Mellon University.

→ kinnaird@cs.cmu.edu

⬇️ **Patrick Minder** is a Ph.D. student at the University of Zurich.

→ minder@ifi.uzh.ch