



MIT Open Access Articles

The Trouble with Social Computing Systems Research

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation	Bernstein, Michael S. et al. "The Trouble with Social Computing Systems Research." in Proceedings of the ACM CHI Conference on Human Factors in Computing Systems, CHI 2011, May 7-12, 2011, Vancouver, BC.
As Published	http://chi2011.org/program/byAuthor.html
Publisher	ACM SIGCHI
Version	Author's final manuscript
Citable link	http://hdl.handle.net/1721.1/64444
Terms of Use	Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.

The Trouble With Social Computing Systems Research

Michael S. Bernstein
MIT CSAIL
Cambridge, MA
msbernst@mit.edu

Ed H. Chi °
Google Research
Mountain View, CA
chi@acm.org

Mark S. Ackerman °
University of Michigan
Ann Arbor, MI
ackerm@umich.edu

Robert C. Miller °
MIT CSAIL
Cambridge, MA
rcm@mit.edu

Abstract

Social computing has led to an explosion of research in understanding users, and it has the potential to similarly revolutionize systems research. However, the number of papers designing and building new sociotechnical systems has not kept pace. We analyze challenges facing social computing systems research, ranging from misaligned methodological incentives, evaluation expectations, double standards, and relevance compared to industry. We suggest improvements for the community to consider so that we can chart the future of our field.

° These authors contributed equally to the paper and are listed alphabetically.

Copyright is held by the author/owner(s).

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

ACM 978-1-4503-0268-5/11/05.

Author's Keywords

Social computing, systems research, evaluation

ACM Classification Keywords

H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces (Evaluation/Methodology).

General Terms

Design, Human Factors

Introduction

The rise of social computing is impacting SIGCHI research immensely. Wikipedia, Twitter, Delicious, and Mechanical Turk have helped us begin to understand people and their interactions through large, naturally occurring datasets. Computational social science will only grow in the coming years.

Likewise, those invested in the *systems* research community in social computing hope for a trajectory of novel, impactful sociotechnical designs. By systems research, we mean *research whose main contribution is the presentation of a new sociotechnical artifact, algorithm, design, or platform*. Systems research in social computing is valuable because it envisions new ways of interacting with social systems, and spreads these ideas to other researchers and the world at large.

This research promises a future of improved social interaction, as well as useful and powerful new user capabilities. Traditional CSCW research had no shortage of systems research, especially focusing on distributed teams and collaboration [1][17][30]. In some ways, systems research has already evolved: we dropped our assumptions of single-display, knowledge-work-focused, isolated users [10]. This broader focus, married with a massive growth in platforms, APIs, and interest in social computing, might suggest that we will see lots of new interesting research systems.

Unfortunately, the evidence suggests otherwise. Consider submissions to the Interaction Beyond the Individual track at CHI 2011. Papers submitted to this track that chose “Understanding Users” as a primary contribution outnumbered those that selected “Systems, Tools, Architectures and Infrastructure” by a ratio of four to one this year [26]. This 4:1 ratio may reflect overall submission ratios to CHI, represent a steady state and not a trend, or equalize out in the long term in terms of highly-cited papers. However, a 4:1 ratio is still worrying: a perceived publication bias might push new researchers capable of both types of work to study systems rather than build them. If this happens en masse, it might threaten our field’s ability to look forward.

In this paper we chart the future of social computing systems research by assessing three challenges it faces today. First, social computing systems are caught between social science and computer science, with each discipline de-valuing work at the intersection. Second, social computing systems face a unique set of challenges in evaluation: expectations of exponential growth and criticisms of snowball sampling. Finally:

how can academic social computing research compete or cooperate with industry? Where possible, we will offer proposed solutions to these problems. They are not perfect – we hope that the community will take up our suggestions, improve them and encourage further debate. Our goal is to raise awareness of the situation and to open a conversation about how to fix it.

Related Work

This paper is not meant to denote all social or technical issues with social computing systems. For example, CSCW and social computing systems suffer from the socio-technical gap [2], critical mass problems [25], and misaligned incentives [15]. These, and many others, are critical research areas in their own right.

We are also not the first to raise the plight of systems papers in SIGCHI conferences. All systems research faces challenges, particularly with evaluation. Prior researchers argue that reviewers should moderate their expectations for evaluations in systems work:

- Evaluation is just one component of a paper, and issues with it should not doom a paper [23][27].
- Longitudinal studies should not be required [22].
- Controlled comparisons should not be required, if the system is sufficiently innovative or aimed at wicked problems [14][22][29].

Not all researchers share these opinions. In particular, Zhai argues that existing evaluation requirements are still the best evaluation strategy we know of [35].

Others have also discussed methodological challenges in HCI research. Kaye and Sengers related how psychologists and designers clashed about study methodology in the conversation of discount usability analysis methods [18]. Barkhuus traced the history of

evaluation at CHI and found fewer users in studies and more papers using studies in recent years [3].

Novelty: Between A Rock and A Hard Science

Social computing systems research bridges the technical research familiar to CHI and UIST with the intellectual explorations of social computing, social science and CSCW. Ideally, these two camps should be combining methodological strengths. Unfortunately, they can actively undermine each other.

Following Brooks [8] and Lampe [20], we split the world of sociotechnical research into those following a computer science engineering tradition (“Builders”) and those following a social science tradition (“Studiers”). Of course, most SIGCHI researchers work as both – including the authors of this paper. But, these abstractions are useful to describe what is happening.

Studiers: Strength in Numbers

Studiers’ goal is to see proof that an interesting social interaction has occurred, and to see an explanation of why it occurred. Social science has developed a rich set of rigorous methods for seeking this proof, both quantitative and qualitative, but the reality of social computing systems deployments is that they are messy. This science vs. engineering situation creates understandable tension [8]. However, we will argue that the prevalence of Studiers in social computing (reflected in the 4:1 paper submission ratio) means that Studiers are often the most available reviewers for a systems paper on a social computing topic.

Social computing systems are often evaluated with field studies and field experiments (*living laboratory* studies [10]), which capture ecologically valid situations. These

studies will trade off many aspects of validity, producing a biased sample or large manipulations that make it difficult to identify which factors led to observed behavior. When Studiers review this work, even well-intentioned ones may then fall into the Fatal Flaw Fallacy [27]: rejecting a systems research paper because of a problem with the evaluation’s internal validity that, on balance, really should not be damning. Solutions like online A/B testing, engaging in long-term conversations with users, and multi-week studies are often out of scope for systems papers [22]. This is especially true in systems with small, transient volunteer populations. Yet Studiers may reject a paper until it has conclusively proven an effect.

Social computing systems are particularly vulnerable to Studier critique because of reviewer sampling bias. Some amount of methodological sniping may be inevitable in SIGCHI, but the skew in the social computing research population may harm systems papers here more than elsewhere. In particular, it is more likely that the qualified reviewers on any given social computing topic will be Studiers and not Builders: there are relatively few people who perform studies on tangible interaction, for example, but a large number of those researching Facebook are social scientists.

Builders: Keep It Simple, Stupid – or Not?

Given the methodological mismatch with Studiers, we might consider asking Builders to review social computing systems papers. Unfortunately, these papers are not always able to articulate their value in a way that Builders might appreciate either.

Builders want to see a contribution with technical novelty: this often translates into *elegant complexity*.

Memorable technical contributions are simple ideas that enable interesting, complex scenarios. Systems demos will thus target flashy tasks, aim years ahead of the technology adoption curve, or assume technically literate (often expert) users. For example, end user programming, novel interaction techniques, and augmented reality research all make assumptions about Moore's Law, adoption, or user training.

Social computing systems contributions, however, are not always in a position to display elegant complexity. Transformative social changes like microblogging are often successful *because* they are simple. So, interfaces aimed ahead of the adoption curve may not attract much use on social networks or crowd computing platforms. A complex new commenting interface might be a powerful design, but it may be equally difficult to convince large numbers of commenters to try it [19]. Likewise, innovations in underlying platforms may not succeed if they require users to change their practice.

Caught In the Middle

Researchers are thus stuck between making a system *technically* interesting, in which case a crowd will rarely use it because it is too complex, and simplifying it to produce *socially* interesting outcomes, in which case Builder colleagues may dismiss it as less novel and Studier colleagues may balk at an uncontrolled field study. Here, a CHI metareviewer claims that a paper has fallen victim to this problem (paraphrased)¹:

The contribution needs to take one strong stance or another. Either it describes a novel system or a novel social interaction. If it's a system,

then I question the novelty. If it's an interaction, then the ideas need more development.

For example, Twitter would not have been accepted as a CHI paper: there were no complex design or technical challenges, and a first study would have come from a peculiar subpopulation. It is possible to avoid this problem by veering hard to one side of the disciplinary chasm: recommender systems and single user tools like Eddi [6] and Statler [32] showcase complexity by doing this. But to accept polarization as our only solution rules out a broad class of interesting research.

A Proposal for Judging Novelty

The combination of strong Studiers and strong Builders in the subfield of social computing has immense potential if we can harness it. The challenge as we see it is that social computing systems cannot articulate their contributions in a language that either Builders or Studiers speak currently. Our goal, then, should be to create a shared language for research contributions. Here we propose the Social/Technical yardstick for consideration. We can start with two contribution types.

Social contributions change how people interact. They enable new social affordances, and are foreign to most Builders. For example:

- New forms of social interaction: e.g., shared organizational memory [1] or friendsourcing [7].
- Designs that impact social interactions: for example, increasing online participation [4].
- Socially translucent systems [13]: interactive systems that allow users to rely on social intuitions.

Technical contributions are novel designs, algorithms, and infrastructures. They are the mechanisms

¹ All issues pointed out by metareviewers are paraphrased from single reviews, though they reflect trends drawn from several.

supporting social affordances, but are more foreign to Studiers. For example:

- Highly original designs, applications, and visualizations designed to collect and manage social data, or powered by social data (e.g., [6], [33])
- New algorithms that coordinate crowd work or derive signal from social data: e.g., Find-Fix-Verify [5] or collaborative filtering.
- Platforms and infrastructures for developing social computing applications (e.g., [24]).

The last critical element is an interaction effect: paired Social and Technical contributions can increase each other's value. ManyEyes is a good example [34]: neither visualization authoring nor community discussion are hugely novel alone. The combination, however, produced an extremely influential system.

Evaluation: Challenges in Living Labs

Evaluation is evolving in human-computer interaction, and social computing is a leading developer of new methodologies. Living laboratory studies [10] of social computing systems have broken out of the university basement, focused on ecologically valid situations and enabled many more users to experience our research. However, evaluation innovation means that we are the first to experience challenges with our methods.

In this section, we identify emerging evaluation requirements and biases that, on reflection, may not be appropriate. These reflections are drawn from conversations with researchers in the community. They are bound to be filtered by our own experience, but we believe them to be reasonably widespread.

Expecting Exponential Growth

Reviewers often expect that research systems have exponential (or large) growth in voluntary participation, and will question a system's value without it. Here is a CHI metareviewer, paraphrased:

As most of the other reviewers mentioned, your usage data is not really compelling because only a small fraction of Facebook is using the application. Worse, your numbers aren't growing in anything like an exponential fashion.

There are a number of reasons why reviewers might expect exponential growth. First, large numbers of users legitimize an idea: growth is strong evidence that the idea is a good one and that the system may generalize. Second, usage numbers are the *lingua franca* for evaluating non-research social systems, so why not research systems as well? Last, social computing systems are able to pursue large-scale rollouts, so the burden may be on them to try. We agree that if exponential growth does not occur, authors should acknowledge this and explore why.

However, it misses the mark to *require* exponential growth for a research system. One major reason this is a mistake is that it puts social computing systems on unequal footing with other systems research. Papers in CHI 2006 had a median of 16 participants: program committees considered this number acceptable for testing the research's claims [3]. Just because a system is more openly available does not mean that we need orders of magnitude more users to understand its effects. Sixteen friends communicating together on a Facebook application may still give an accurate picture.

Another double standard is a conflation of *usefulness* and *usability* [21]. Usefulness asks whether a system

solves an important problem; usability asks how users interact with the system. Authors typically prove usefulness through argumentation in the paper's introduction, then prove usability through evaluation. Evaluations will shy away from usefulness because it is hard to prove scientifically. Instead, we pay participants to come and use our technology temporarily (assuming away the motivation problem), because we are trying to understand the effects of the system once somebody has decided to use it. This should be sufficient for social computing systems as well. However, reviewers in social computing systems papers will look at an evaluation and decide that a lack of adoption empirically disproves any claim of usefulness. ("Otherwise, wouldn't people flock to the system?") However, why require usefulness – voluntary usage – in social computing systems, when we assume it away – via money – with other systems research?

A final double standard is whether we expect risky hypothesis testing or conservative existence proofs of our systems' worthiness [14]. A public deployment is the riskiest hypothesis testing possible: the system will only succeed if it has gotten almost everything right, including marketing, social spread, graphic design, and usability. Systems evaluations elsewhere in CSCW, CHI and UIST seek only existence proofs of specific conditions where they work. We will not argue whether existence proofs are always the correct way to evaluate a systems paper, but it is problematic to hold a double standard for social computing systems papers.

The second major reason it is a mistake to require exponential growth is that a system may fail to spread for reasons entirely unrelated to its research goals. Even small problems in the social formula could doom a

deployment: minor *channel factors* like slow logins or buggy software have major impact on user behavior [31]. Rather than immediate success, we should expect new ideas to spawn a series of work that gets us continually closer. Achieving Last.fm on the first try is very unlikely – we need precursors like Firefly first. In fact, we may learn the most from failed deployments of systems with well-positioned design rationale.

Get Out of the Snow! No Snowball Sampling

Live deployments on the web have raised the specter of snowball sampling: starting with a local group in the social graph and letting a system spread organically. CHI generally regards snowball sampling as bad practice. There is good reason for this concern: the first participants will have a strong impact on the sample, introducing systematic and unpredictable bias into the results. Here, a paper metareviewer calls out the sampling technique (paraphrased):

The authors' choice of study method – snowball sampling their system by advertising within their own social network – potentially leads to serious problems with validity.

Authors must be careful not to overclaim their conclusions based on a biased sample. However, some reviewers will still argue that systems should always recruit a random sample of users, or make a case that the sample is broadly representative of a population.

This argument is flawed for three reasons. First, we must recognize that *snowball sampling is inevitable in social systems*. Social systems spread through social channels – this is fundamental to how they operate. We need to expect and embrace this process.

Second, random sampling can be an impossible standard for social computing research. All it takes is a single influential user to tweet about a system and the sample will be biased. Further, many social computing platforms like Twitter and Wikipedia are beyond the researcher's ability to recruit randomly. While an online governance system might only be able to recruit citizens of one or two physical regions, leading to a biased sample, it is certainly enough to learn the highest-order bit lessons about the software.

Finally, snowball sampling is another form of convenience sampling, and convenience sampling is common practice across CHI, social sciences and systems research. Within CHI, we often gather the users in laboratory or field studies by recruiting locals or university employees, which introduces bias: half of CHI papers through 2006 used a primarily student population [3]. We may debate whether convenience sampling in CHI is reasonable on the whole (e.g., Barkhuus [3]), but we should not apply the criteria unevenly. However, we should keep in mind that convenience and snowball sampling are widely accepted methods in social science to reach difficult populations.

A Proposal for Judging Evaluations

Because our methodological approaches have evolved, it is time to develop meaningful and consistent norms about these concerns. An exhortation to take the long view and review systems papers holistically (e.g., [14][22][29]) can be difficult to apply consistently. So, in this section we aim for more specific suggestions.

Separate Evaluation of Spread from Steady-State

We argued that exponential growth is a faulty signal to use in evaluation. But, there are times when we should

expect to see viral spread in an evaluation: when the research contribution *makes claims about spread*.

We can separate two types of usage evaluations: *spread* and *steady-state*. Most social systems go through both phases: (1) an initial burst of adoption, then (2) upkeep and ongoing user interaction. Of course, this is a simplification: no large social system is ever truly in steady-state due to continuous tweaks, and most new systems go through several iterations before they first attract users. But, for the purposes of evaluation, these phases ask clear and different questions. An evaluation can focus on how a system spreads, or it can focus on how it is used once adopted.

Paper authors should evaluate their system with respect to the claims they are making. If the claims focus on attracting contributions or increasing adoption, for instance, then a spread evaluation is appropriate. These authors need to show that the system is increasing contributions or adoption. If instead the paper makes claims about introducing a new style of social interaction, then we can ignore questions of adoption for the moment and focus on what happens when people have started using the system. This logic is parallel to laboratory evaluations: we solve the questions of motivation and adoption by paying participants, and focus on the effects of the software once people are using it (called *compelled tasks* in Jared Spool's terminology).

Authors should characterize their work's place on the spread/steady-state spectrum, and argue why the evaluation is well matched. They should call out limitations, but evaluations should not be required to address both spread and steady-state usage questions.

Make A Few Snowballs

As we argued, it is almost impossible to get a random sample of users in a field evaluation. Authors should thus not claim that their observations generalize to an entire population, and should characterize any biases in their snowballed sample. Beyond this, we propose a compromise: a few different snowballs can help mitigate bias. Authors should make an effort to seed their system at a few different points in the social network, characterize those populations and any limitations they introduce, and note any differences in usage. But we should not reject a paper because its sample falls near the authors' social network. There may still be sufficient data to evaluate the authors' claims relatively even-handedly. Yes, perhaps the evaluation group is more technically apt than the average Facebook user; but most student participants in SIGCHI user studies are too [3].

Treat Voluntary Use As A Success

Finally, we need to stop treating a small amount of voluntary use as a failure, and instead recognize it as *success*. Most systems studies in CHI have to pay participants to come in and use buggy, incomplete research software. Any voluntary use is better than many CHI research systems will see. Papers should get *extra* consideration for taking this approach, not less.

Research At A Disadvantage with Industry

Social computing systems research now needs to forge its identity between traditional academic approaches and industry. Systems research is used to being ahead of the curve, using new technology to push the limits of what is possible. But in the age of the social computing, academic research often depends on access to industry platforms. Otherwise, a researcher must function as an

entire start-up – marketing, engineering, design, QA – and compete with companies for attention and users. It is not surprising that researchers worry whether start-ups are a better path (see Landay [22] and associated comments). If they stay in academia, researchers must satisfy themselves with limited access to platforms they didn't create, or chance attracting a user population from scratch and then work to maintain it. What path should we follow to be the most impactful?

One challenge is that, in many domains, users are largely in closed platforms. These platforms are typically averse to letting researchers make changes to their interface. If a researcher wanted to try changing Twitter to embed non-text media in tweets, they should not expect cooperation. Instead, we must re-implement these sites and then find a complicated means of attracting naturalistic use. For example, Hoffman et al. mirrored and altered Wikipedia, then used AdWords advertisements cued on Wikipedia titles to attract users [16]. Wikidashboard also proxied Wikipedia [33].

Many would go farther and argue that social computing systems research is a misguided enterprise entirely. Brooks argues that, with the exception of source control and Microsoft Word's Track Changes, CSCW has had no impact on collaboration tools [8]. Welsh claims that researchers cannot really understand large-scale phenomena using small, toy research environments.²

Academic-industry partnerships represent a way forward, though they do have drawbacks. Some argue that if your research competes with industry, you should go to an industry lab (see Ko's comment in

² <http://matt-welsh.blogspot.com/2010/10/computing-at-scale-or-how-google-has.html>

Landay [22]). Industrial labs and academic collaboration have worked well with organizations like Facebook, Wikimedia, and Microsoft. However, for every academic collaboration, there are many non-starters: companies want to help, but cannot commit resources to anything but shipping code. Industrial collaboration opens up many opportunities, but it is important to create other routes to innovation as well.

Some academics choose to create their own communities and succeed, for example Scratch [28], MovieLens³, and IBM Beehive [12]. These researchers have the benefit of continually experimenting with their own platforms and modifying them to pursue new research. Colleagues' and our own experience indicate that this approach carries risks as well. First, many such systems never attract a user base. If they do, they require a large amount of time for support, engineering and spam fighting that pays back little. This research *becomes a product*. Such researchers express frustration that their later contributions are then written off as "just scaling up" an old idea.

Even given the challenges, it is valuable for researchers to walk this line because they can innovate where industry does not yet have incentive to do so. Industry will continue to refine existing platforms [11], but systems research can strike out to create alternate visions and new futures. Academia is an incubator for ideas that may someday be commercialized, and for students who may someday be entrepreneurs. It has the ability to try, and to fail, quickly and without negative repercussion. Its successes will echo through replication, citations and eventual industry adoption.

³ <http://www.movielens.org/>

Conclusion

Social computing systems research can struggle to succeed in SIGCHI. Some challenges relate to our research questions: interesting social innovations may not be interesting technically, and meaningless to social scientists without proof. In response, we laid out the Social/Technical yardstick for valuing research claims. Other challenges are in evaluation: a lack of exponential growth and snowball sampling are incorrectly dooming papers. We argued that these requirements are unnecessary, and that different phases of system evaluation are possible: spread and steady state. Finally, we considered the place of social computing systems research with respect to industry.

As much as we would like to have answers, we know that this is just the beginning of a conversation. We invite the community to participate and contribute their suggestions. We hope that this work will help catalyze the social computing community to discuss the role of systems research more openly and directly.

Acknowledgments

The authors would like to thank Travis Kriplean, Mihir Kedia, Eric Gilbert, Cliff Lampe, David Karger, Bjoern Hartmann, Adam Marcus, Andrés Monroy-Hernandez, Drew Harry, Paul André, David Ayman Shamma, and the alt.chi reviewers for their feedback.

References

- [1] Ackerman, M. 1994. Augmenting the Organizational Memory: A Field Study of Answer Garden. In *Proc. CSCW '94*.
- [2] Ackerman, M.S. 2000. The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. *Human-Computer Interaction* 15(2).
- [3] Barkhuus, L., and Rode, J.A. 2007. From Mice to Men – 24 years of evaluation in CHI. In *Proc. alt.chi '07*.

- [4] Beenen, G., Ling, K., Wang, X., et al. 2004. Using social psychology to motivate contributions to online communities. In *Proc. CSCW '04*.
- [5] Bernstein, M.S., Little, G., Miller, R.C., et al. Soylent: A Word Processor with a Crowd Inside. In *Proc. UIST '10*.
- [6] Bernstein, M.S., Suh, B., Hong, L., Chen, J., Kairam, S., and Chi, E.H. 2010. Eddi: Interactive topic-based browsing of social streams. In *Proc. UIST 2010*.
- [7] Bernstein, M.S., Tan, D., Smith, G., et al. 2010. Personalization via Friendsourcing. *TOCHI* 17(2).
- [8] Brooks, F. 1996. The computer scientist as toolsmith II. *Communications of the ACM* 39(3).
- [9] Brooks, F. 2010. *The Design of Design*. Addison Wesley: Upper Saddle River, NJ.
- [10] Chi, E.H. 2009. A position paper on 'Living Laboratories': rethinking ecological designs and experimentation in human-computer interaction. HCI International 2009.
- [11] Christensen, C.M. 1997. *The Innovator's Dilemma*. Harvard Business Press.
- [12] DiMicco, J., Millen, D.R., Geyer, W., et al. 2008. Motivations for social networking at work. In *Proc. CSCW 2008*.
- [13] Erickson, T., and Kellogg, W. 2000. Social Translucence: An approach to designing systems that support social processes. *TOCHI* 7(1).
- [14] Greenberg, S., and Buxton, B. 2008. Evaluation considered harmful (some of the time). In *Proc. CHI '08*.
- [15] Grudin, J. 1989. Why groupware applications fail: problems in design and evaluation. *Office: Technology and People* 4(3): 187-211.
- [16] Hoffmann, R., Amershi, S., Patel, K., et al. 2009. Amplifying community content creation with mixed initiative information extraction. In *Proc. CHI '09*.
- [17] Ishii, H., and Kobayashi, M. 1992. ClearBoard: a seamless medium for shared drawing and conversation with eye contact. In *Proc. CHI '92*.
- [18] Kaye, J., and Sengers, P. 2007. The evolution of evaluation. In *Proc. alt.chi 2007*.
- [19] Kriplean, T., Toomim, M., Morgan, J., et al. 2011. REFLECT: Supporting Active Listening and Grounding on the Web through Restatement. In *Proc. CSCW '11 Horizon*.
- [20] Lampe, C. 2010. The machine in the ghost: a socio-technical approach to user-generated content research. WikiSym keynote. <http://www.slideshare.net/cliff Lampe/the-machine-in-the-ghost-a-sociotechnical-perspective>
- [21] Landauer, T. 1995. *The Trouble with Computers: usefulness, usability and productivity*. MIT Press, Cambridge.
- [22] Landay, J.A. 2009. I give up on CHI/UIST. Blog. <http://dubfuture.blogspot.com/2009/11/i-give-up-on-chiuiist.html>.
- [23] Lieberman, H. 2003. Tyranny of Evaluation. *CHI Fringe*.
- [24] Little, G., Chilton, L., Goldman, M., and Miller, R.C. TurKit: Human Computation Algorithms on Mechanical Turk. *UIST '10*.
- [25] Markus, L., 1987. Towards a "critical mass" theory of interactive media: Universal access, interdependence, and diffusion. *Communication Research* 14: 491-511.
- [26] Morris, M. Personal communication. December 22, 2010.
- [27] Olsen Jr., D. 2007. Evaluating User Interface Systems Research. In *Proc. UIST '07*.
- [28] Resnick, M., Maloney, J., Monroy-Hernández, A., et al. 2009. Scratch: programming for all. *Commun. ACM* 52(11).
- [29] Rittel, H.W.J. and Webber, M.M. 1973. Dilemmas in a general theory of planning. *Policy Sciences* 4:2, pp. 155-169.
- [30] Roseman, M., and Greenberg, S. 1996. Building real-time groupware with GroupKit, a groupware toolkit. *TOCHI* 3(1).
- [31] Ross, L., and Nisbett, R. 1991. *The person and the situation*. Temple University Press.
- [32] Shamma, D., Kennedy, L., and Churchill, E. Tweetgeist: Can the twitter timeline reveal the structure of broadcast events? In *CSCW 2010 Horizon*.
- [33] Suh, B., Chi, E.H., Kittur, A., and Pendleton, B.A. 2008. Lifting the veil: improving accountability and social transparency in Wikipedia with WikiDashboard. *Proc. CHI '08*.
- [34] Viegas, F., Wattenberg, M., van Ham, F., et al. 2007. ManyEyes: a Site for Visualization at Internet Scale. *IEEE Trans. Viz. and Comp. Graph.* Nov/Dec 2007: 1121-1128.
- [35] Zhai, S. 2003. Evaluation is the worst form of HCI research except all those other forms that have been tried. Essay published at *CHI Place*.