The Human Element in Social Networking

Forrest Shull

I HATE TO spoil anyone's field of dreams—but nevertheless, just because you built it doesn't mean they will come.

Social networking sites—and the energy, enthusiasm, and creativity that they can help unleash for all sorts of goals—are increasingly ubiquitous. Moreover, such sites and their volunteer communities are becoming such an embedded part of our lives that in many cases we simply take it for granted. But for all the successful social networks that we rely on, many others never get off the ground—and many that seem promising never reach

ports or frustrates various ways for them to interact, and that limits the amount and type of information they can share. But it's still only one part of the social network.

Elsewhere in this issue, you'll find articles discussing technical aspects of creating and supporting social networks for software development. But I was curious about what lessons have been learned more generally, especially about what differentiates successful from not-so-successful networks—and how software developers can support social networks more effectively.

A Nation of Neighbors

I first contacted Ben Shneiderman, a professor in the Department of Computer Science and founding director (1983–2000) of UMD's Human-Computer Interaction Laboratory. He's published several well-regarded books and received numerous accolades for his research on HCI and the impact of computers on human and societal capabilities.

One of his recent projects is notable for its impressive goals: the Nation of Neighbors (NoN; www.cs.umd.edu/hcil/non). On one level, NoN is designed to facilitate real-time collaboration within neighborhood communities across the US—especially between community members and law enforcement officials. On another level, NoN provides a rich testbed for the development of visualization and analytic tools aiming to improve the understanding of social networks. NoN currently supports more than 400 communities across the country, of which 40 communities have been particularly successful and durable. In this sense, NoN provides a natural laboratory for studying which factors facilitate successful social networks.

Shneiderman emphasized that, at a high level, his findings show that success often has more to do with people issues

Success often has more to do with people issues than technology.

the critical mass that provides a meaningful experience for their members.

Software can provide a critical part of the equation. It's software, after all, that provides the infrastructure for members to come together, that sup-

For that, I turned to two experts who have been building and studying ambitious systems that rely on social networks here at the University of Maryland (UMD), which is becoming a leading institution in this field.

STAFF

Lead Editor

Brian Brannon

bbrannon@computer.org

Manager, Editorial Services

Jenny Stout

Editors

Camber Agrelius and Linda World

Publications Coordinator

software@computer.org

Production Editor/Webmaster

Jennie Zhu

Contributor

Alex Torres

Cover Artist

Eero Johannes

Director, Products & Services

Evan Butterfield

Senior Manager, Editorial Services

Lars Jentsch

Senior Business Development Manager

Sandra Brown

Membership Development Manager

Cecelia Huffman

Senior Advertising Coordinator

Marian Anderson

manderson@computer.org

CS PUBLICATIONS BOARD

Thomas M. Conte (chair), Alain April,
David Bader, Angela R. Burgess, Greg Byrd,
Koen DeBosschere, Frank Ferrante,
Paolo Montuschi, Linda I. Shafer,
and Per Stenström

MAGAZINE

OPERATIONS COMMITTEE

Paolo Montuschi (chair), Erik R. Altman,
Nigel Davies, Lars Heide, Simon Liu,
Cecelia Metra, Michael Rabinovich,
Shari Lawrence Pfleeger, Forrest Shull,
John Smith, Gabriel Taubin,
George K. Thiruvathukal, Ron Vetter,
and Daniel Zeng

Editorial: All submissions are subject to editing for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Software* does not necessarily constitute endorsement by IEEE or the IEEE Computer Society.

To Submit: Access the IEEE Computer Society's Web-based system, ScholarOne, at <http://mc.manuscriptcentral.com/sw-cs>. Be sure to select the right manuscript type when submitting. Articles must be original and not exceed 4,700 words including figures and tables, which count for 200 words each.

IEEE prohibits discrimination, harassment and bullying: For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

than technology. Beyond that, however, several patterns provide important implications for developers of software with a social networking component.

Motivation through Recognition

Shneiderman described how NoN communities began in various ways: some by police departments, some by community groups, some by individuals. Regardless of their origin, community success hinged on whether at least one person—regardless of his or her role—stepped up as a leader to keep things moving, responding to queries and initiating discussion. Leaders are knowledgeable and informed. Simply put, they care about the success of a community and work to sustain it.

Of course, such leaders are likely volunteers and make their own decisions about where to invest their time. So, how can developers design systems that facilitate the creation of leaders?

"We're only at the beginning stages of trying to figure out the motivational structure for successful online communities," Shneiderman said. However, his research of existing social networks has resulted in the "Reader to Leader" framework, which encapsulates key factors that shape how a fairly small percentage of social network users grow to become contributors and how an even smaller percentage go on to become more active collaborators.¹

One of the key implications for system designers is that social networking sites must give users a pathway for advancement—that is, any type of user should have a well-defined "next level" of involvement if they're interested in investing additional time. Successful systems also provide recognition and rewards as users advance from one level to another.

Users at any level should have access to system features that facilitate their roles and rewards the effort they've invested. Shneiderman says

that system designers should consider giving users at the upper levels access to more data for monitoring the community and more power to allow them to respond quickly to situations. Leaders are not only involved in the social network's successes, they're also the ones the community relies on for defusing problems and keeping things running smoothly.

Above all, system designers shouldn't overlook simple mechanisms for letting users thank one another, such as allowing them to "like" or comment on some content on the site, or showing the number of downloads, citations, or links to the work. An important part of building a cohesive online community is letting members know that their efforts are appreciated.

Flexibility and Feedback

Chances are that if you work in a large organization, you're familiar with a lessons learned or best practices repository. I've seen many of these myself and have found that they're increasingly being designed to leverage a social network where employees can share their experiences and address each other's questions. Yet, in most cases, adding a social networking element hasn't had a huge effect on their success.

Shneiderman says that designing such sites in a top-down manner that doesn't take user's motivations into account is unlikely to lead to success. Finding the right design is complicated. All sorts of social concerns—for example, self-interest, altruism, and moral principles—can motivate users, and finding the right way to support them in different environments can be difficult.

Thus, it's necessary to build the capacity to try new things into a system and learn from the results. As an example, Shneiderman described an intervention by his colleague, Gerhard Fischer, who introduced a fairly simple tweak to a large corporation's

social network site—letting users award points to the respondents who provided the most help with a question—which increased its effectiveness tremendously.² It’s important to note that this particular tweak isn’t the key—the solutions that work in large corporations might not work in other contexts—it’s the ability to try such innovations and understand their effect on results.

My conversation with Ben Shneiderman contained many more useful nuggets than I could summarize in this article, and he’s a wonderful tour guide through this area of research. I’m sure you will enjoy listening to the audio highlights from our discussion, at <http://youtu.be/yFG3lL6HJ38>.

The Children’s Library

To get another perspective on how these principles show up in system design, I also spoke with Anne Rose, a staff researcher at UMD who manages the volunteer effort for another free and publicly available site, the International Children’s Digital Library (ICDL; <http://en.childrenslibrary.org>). The ICDL isn’t a social networking site, but it provides a popular service over a software infrastructure that relies on the efforts of a volunteer community.

The success of this initiative is reflected in its up to 130,000 unique visitors each month. The ICDL began as

the world for children ages 3 to 13. The ICDL aims to provide literature from different cultures to help promote tolerance and understanding among children in different countries and contains books from 65 countries and in 61 languages.

Covering so many different languages requires a pool of more than 4,000 active volunteers. Clearly, these are the energized and enthusiastic community members that are so important to a network’s success. “I just can’t keep up with everything that people are willing to do for us,” said Rose.

I talked with Rose about her lessons learned for so successfully involving and keeping up the motivation from such a diverse group, and how the software can best support that.

As in my conversation with Shneiderman, I could only fit a few highlights out of many into this article. The ICDL is a hugely interesting cross-cultural endeavor, and I heartily recommend interested readers to listen to my entire discussion with Rose at <http://youtu.be/M3R4-XLBRYE>.

Matching Tasks to Users

Our discussion of the ICDL afforded me an opportunity to see how the themes Shneiderman identified as important for NoN were instantiated in a different context.

volunteers might be motivated by collectivism (they’re interested in advancing the field of children’s literature or in having more books from their own language included); some might find a measure of self-interest in participation (they want to see more of this type of content available for their children or use their translations on their resume). Trying to design a site that addresses just one motivation is likely to miss important issues for certain segments of the volunteers.

Shneiderman emphasized that having multiple levels of participation with different types of recognition was important, and the ICDL is no exception. However, the ICDL focuses more on providing different levels of activities to best match volunteers’ skill levels and available time. For example, entering metadata on searchable books is a necessary, although perhaps not exciting task, and can be done with a smaller investment of time. This can be a good activity for more casual contributors, while members with more time and a different skill level might take on a full translation of a book.

Eliciting User Needs

The ICDL team invested great effort in understanding both their volunteers and their user communities. They worked with children ages 7 to 11 in four countries to design a searchable interface for the library. Despite this upfront investment, they still found issues that arise from cross-cultural problems. For example, to keep the interface simple, much of the functionality is represented by icons. To search on humorous books, the team had originally used an icon of a child sticking his tongue out; they later discovered this was considered disrespectful in some cultures and needed to be changed.

This experience speaks to the need to allow the system to evolve as developers discover what makes sense and

Trying to design a site that addresses just one motivation is likely to miss important issues.

a research project and is currently operated by a nonprofit foundation. After a decade in operation, it contains more than 4,600 books from around

Rose discussed a wide variety of reasons why people volunteer. Although she felt that altruism was often the compelling issue, she also noted that some

proves useful to different segments of the user network. The mechanisms for gathering input from the social network might change over time as the community does. The ICDL is a good example in which a particular mechanism—focus groups—was applied in the beginning, but more opportunistic feedback mechanisms could be utilized later as the number of visitors to the site became more significant.

Sites that help members define their own needs might find that this leads to effective features that the system designers would never have imagined. For example, having worked with children and having studied how they select or remember books, Rose's group incorporated unusual search features that made sense to their user community—for example, selecting books by the

color of their cover or by the feelings that the children associated with reading that story.

This is an exciting time for work on social networking principles. To gain a more complete picture of how such communities grow and develop requires truly collaborative and interdisciplinary work that brings together not only software designers but also domain experts and sociologists.

Still, one thing that's exceptionally clear is that by leveraging such systems effectively, software engineers are developing the power to affect societal issues. The current technology has advanced to a point where researchers and developers can study and improve the

IEEE Software

HOW TO REACH US

WRITERS

For detailed information on submitting articles, write for our Editorial Guidelines (software@computer.org) or access www.computer.org/software/author.htm.

LETTERS TO THE EDITOR

Send letters to

Editor, *IEEE Software*
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
software@computer.org

Please provide an email address or daytime phone number with your letter.

ON THE WEB

www.computer.org/software

SUBSCRIBE

www.computer.org/software/subscribe

SUBSCRIPTION CHANGE OF ADDRESS

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Be sure to specify *IEEE Software*.

MEMBERSHIP CHANGE OF ADDRESS

Send change-of-address requests for IEEE and Computer Society membership to member.services@ieee.org.

MISSING OR DAMAGED COPIES

If you are missing an issue or you received a damaged copy, contact help@computer.org.

REPRINTS OF ARTICLES

For price information or to order reprints, send email to software@computer.org or fax +1 714 821 4010.

REPRINT PERMISSION

To obtain permission to reprint an article, contact the Intellectual Property Rights Office at copyrights@ieee.org.

Call for Articles

IEEE Software seeks practical, readable articles that will appeal to experts and nonexperts alike. The magazine aims to deliver reliable information to software developers and managers to help them stay on top of rapid technology change. Submissions must be original and no more than 4,700 words, including 200 words for each table and figure.

Author guidelines: www.computer.org/software/author.htm
Further details: software@computer.org

www.computer.org/software

IEEE
Software

FIGURE 1. NodeXL analysis of a filtered Twitter network graph for the hashtag #IEEE. Node size indicates number of Twitter followers (for scale, IEEE Spectrum has 24.5K followers), and boxes show the results of clustering by connectedness. This analysis shows major sources of tweets on this hashtag, a natural breakdown of sources into groups, and the interconnectedness of those groups.

mechanisms and motivations behind social groups. See some of the resources related to the November 2010 special issue of *IEEE Computer* (www.cs.umd.edu/hcil/tmsp) and the free network analysis tool NodeXL (<http://nodexl.codeplex.com>)—for just one example of a NodeXL analysis, note that it's been used to visualize the Twitter network graph around the hashtag #ieee (see Figure 1).

The two systems that I discuss here address ambitious goals such as community safety and cultural understanding. In the words of Shneiderman, “The things we’ve talked about are increasingly important and address national priorities... So the software designers who try out these methods and create motivational structures through the software are potentially enormously influential for the future security and wealth of the US and every country in the world.”

References

1. J. Preece and B. Shneiderman, “The Reader-to-Leader Framework: Motivating Technology-Mediated Social Participation,” *Assoc. for Information Systems Trans. Human-Computer Interaction*, vol. 1, no. 1, 2009, pp. 13–32; <http://aisel.aisnet.org/thci/vol1/iss1/5>.
2. A. Gorman and G. Fischer, “Toward an Analytic Framework for Understanding and Fostering Peer-Support Communities in Using and Evolving Software Products,” *Proc. Int'l Conf Communities and Technologies (C&T 09)*, Penn. State Univ., 2009, pp. 1–9; <http://l3d.cs.colorado.edu/~gerhard/papers/2009-CandT-paper.pdf>.

FORREST SHULL is a division director at the Fraunhofer Center for Experimental Software Engineering in Maryland, a nonprofit research and tech transfer organization, where he leads the Measurement and Knowledge Management Division. He's also an adjunct professor at the University of Maryland College Park and editor in chief of *IEEE Software*. Contact him at fshull@computer.org.

IEEE computer society

Corporate Affiliate Program

Increases technical training while cutting costs.

Provides company-wide, employee access to 4,300 technical courses, 600 technical and business books, dozens of Brainbench Exams and free or discounted training webinars and software development certifications.

For more information, call 1-855-727-3632 or email us at cap@computer.org

See www.computer.org/software-multimedia for multimedia content related to this article.