Spam and the Social-Technical Gap

The runaway increase in spam cannot be stemmed by technical change alone: We must bridge the gap between social expectations and computer system capability by designing software to meet social requirements.

Brian Whitworth
New Jersey Institute of Technology

Elizabeth Whitworth
Carleton University

A Ferris Research report estimated that in 2003, unsolicited and unwanted e-mail, or spam, cost US companies $10 billion in lost productivity (www.entmag.com/news/article.asp?EditorialsID=5651). A Sunbelt Software poll found that spam now surpasses viruses as the leading unwanted network intrusion (www.itsecurity.com/tecsnews/jul2003/jul141.htm). A 2003 *Time* magazine article reported that major e-mail providers must delete more than 40 percent of all incoming mail at the server, while AOL estimates that fully 80 percent of its inbound e-mail—1.5 billion to 1.9 billion messages a day—consists of spam that the company blocks. Spam currently constitutes up to 30 percent of all in-box messages. Although each user may take only seconds to deal with them, over billions of cases, these spam messages create a serious problem for people, software, and hardware.

Despite spam filters, spammer lists, and antispam laws, the percentage of transmitted spam rose from 20 to 40 percent during the last half of 2002 and continues to rise, with current estimates nudging 80 percent.[1] Although improved filters trash more spam, spammers send ever more in response.

In these spam wars, as filters become more intelligent so do spammers' countermeasures. In May 2003, the amount of spam exceeded nonspam for the first time: More than 50 percent of transmitted e-mail now consists of spam that consumes bandwidth and network resources whether users see it or not.[2] An ISP that needs one server for customers must buy another just for spam no one reads. Providers pass on such costs to users.

Current spam responses—ranging from moral outrage to spam blockers, spamming the spammers, black and white lists, and legal means—have slowed but not stopped spam. By hiding the problem from users, spam blockers could be making it worse.

Legislation like the US CAN-SPAM Act of 2003 (www.spamlaws.com/federal/108s877.html) may merely move spammers overseas. Lists that identify spammers may grow endlessly as spammers change their identities often. IT writers seem in denial of these problems, espousing new Bayesian spam filters while noting that "the problem with spam is that it is almost impossible to define,"[2] and advocating legal solutions while noting that none have worked so far.

The continued growth of spam suggests the need for a new approach. Although most see spam as a personal problem, we suggest it is a social problem that needs a social response. Yet traditional social responses—law, courts, and the judiciary—seem to work poorly in cyberspace. We propose bridging the gap between society and technology by applying social concepts to technology design.[3]

ORIGINS AND IMPLICATIONS

Spam arises from an online social situation that technology created. First, it costs no more to send a million e-mail messages than to send one. Second, hits are a percentage of transmissions, so sending more spam means more sender profit.

Published by the IEEE Computer Society

Spam generators thus logically seek to reach all users. Without responders, spam would not exist, but a small fraction of recipients always respond—and new ones join the Internet every minute. Spammers need only 100 takers per 10 million requests to earn a profit[1]—much less than a .01 percent hit rate. Even if spam blockers successfully blocked 99.99 percent of all spam, spammer transmissions would continue to increase. As more people come online from all over the world, the problem will worsen.

If spam continues to increase, at what transmission percentage will an equilibrium be reached? Can technology forever expand bandwidth and processing beyond the spam challenge? Probably not. Spam potential increases as the square of the number of users. There are 23 million businesses in the US alone. If each business sent just one unsolicited message a year to all Americans, the Internet would be flooded with more than 63,000 e-mail messages per person per day.

Unwanted e-mail can come from anywhere. Why send your resume to selected firms when you can send it to all firms? Why ask selected people to join your club when you can ask everyone? Social politeness suggests restraint, but spam gets results.

Current trends suggest that within a decade, more than 95 percent of Internet transmissions will be spam. The Internet will then transmit vast amounts of information, but minute amounts of meaning. It will be a messaging system that mostly sends messages people never see. Unless developers put a social heart into technological muscle, this outcome is not just possible but likely.

Filtering spam before transmission could reduce the waste, but this raises another problem: It hides spam "false-positives"—real e-mail filtered as spam. Spam filters make two types of errors—wrongly accepting spam and wrongly rejecting genuine e-mail. Reducing the first error inevitably increases the second, so the cost of filtering 99.99 percent of spam is false rejections.

Currently, receivers can recover false rejects from their spam filter's quarantine area, but filtering before transmission means the message never arrives. Inadvertently using spam words could cause an e-mail server to filter your message, and neither you nor the receiver will know it. Imagine a postal system that shredded presumed unwanted mail at input and made mistakes in the process. If users can't rely on e-mail getting through, they could lose confidence in it.

Either way, by technology overload or collapse of social confidence, spam is more than a nuisance—it threatens the e-mail system itself. Some technology optimists suggest this may be a good thing, because the death of e-mail would lead to new and better forms of communication. However, the electronic social disease we call spam may cross application boundaries. Internet instant messaging already has its own spam version, spim,[4] that is increasing faster than spam did.

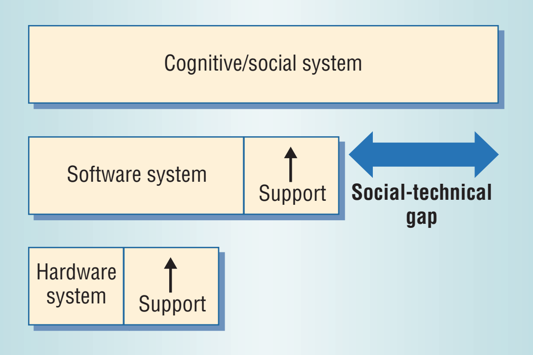

Figure 1. Social-technical gap. In a social-technical system, the system levels may contradict, creating a gap between social expectations and computer capability.

Spam raises simple and subtle questions. A simple question is how can I reduce my spam? A subtle question is how can the e-mail system reduce spam? By analogy, if an athlete can do poorly, what of the game itself? When games fail, the question is not how to improve players, but how to improve the game. For example, one-day cricket reinvigorated the traditional five-day game by changing the entire game, not the people in it.

If spam is a systemic problem, local filters will not solve it. Asking for a tool to stop my spam asks the wrong question. Better to ask how changing the rules of communication can reduce spam.

SOCIAL-TECHNICAL GAP

The main problem facing social software may be the *social-technical gap*: the difference between social expectations and computer system capability.[5] Value-centered computing aims to bridge this gap by building more sociable software.

Figure 1 shows a sociotechnical system as a social system built on a technical one, just as a software system is built atop a hardware system.[3] Here, *technical* means hardware and software. Whether a social system arises from technology or the physical world, it remains a social system. The means of interaction may be electronic, but the people involved are real, which is why an e-mail can be as offensive as a face-to-face comment. Hence, social communication requirements apply equally to virtual and physical interactions.

In Figure 1, higher system levels emerge from lower ones, and thus depend on them, just as software cannot exist without hardware. Yet, to increase system performance, higher-level demands must drive lower-level design. Software requirements like file sharing and networking may drive chip design, and user cognitive skills or limits may drive Web site design. It follows that social technology could be designed to support social requirements. Yet what is a *social requirement*?

Societies that ignore legitimacy do so at their peril.

LEGITIMACY

Society requires accountability—that individuals bear the consequences of their acts—because in society one person's failure can cause another's loss, and someone can benefit from another's success.

The opposite of accountability is unfairness, a complex perception of social situations in which actors take benefits that others earned, or pay no cost for negative acts upon society. Unfairness is not just unequal outcome distribution, but failure to distribute outcome according to action contribution.

Studies suggest that people have a "natural justice" perception of fairness that gauges whether utility gained matches contributions made over time. Thus, people tend to avoid unfair situations, and prefer fair situations even over situations that give personal benefit.

Social systems of law and justice primarily aim to reduce unfairness in society.[6] Yet preventing unfairness is a negative goal, achieved by punitive means such as imprisonment. A positive social goal can be defined as *legitimate interaction,* which treats individuals fairly and benefits the social group.[3]

This complex concept is not defined by fairness alone, as conflict can be fair but harmful. For example, a duel is a fair fight, yet duels are outlawed today.

The term legitimate comes from sociology, where it applies to governments that rule through justice rather than coercion. Legitimacy is more than legality, however. If legitimacy and legality were identical, John Stuart Mill could not talk about the "… limits of power that can be legitimately exercised by society over the individual"[7] because every law would necessarily be legitimate.

Legitimacy includes concepts like:

- *freedom*—people own themselves and thus can be accountable. In modern society, for one to own another is unacceptable. Freedom is fair because people own themselves. Free societies also produce more. Democracy extends freedom to groups—the group owns itself, rather than being owned or enslaved by a king.
- *privacy*—people own their personal information. Privacy derives from freedom: If I own myself, should I not own my information, regardless of where it resides? Privacy is fair if everyone gets it. It also seems a public good.[8]

Progress in legitimate rights seems to correlate with social wealth, while social corruption correlates with poverty (www.transparency.org). Francis Fukuyama argues that societies that support legitimacy prosper, and those that ignore it do so at their peril.[9] Perhaps people in fair societies contribute more work, ideas, and research because others don't steal it. Having more freedom, they also self-regulate more, which reduces security costs significantly. Legitimate interaction seems a requirement for social prosperity, rather than an optional moral extra.

If physical society's principles apply equally to online society, the latter could founder without fairness. By our definition, if spam is unfair to individuals, unprofitable to society, or both, then it is not legitimate communication.

SPAM IS NOT LEGITIMATE COMMUNICATION

Spam is unfair because it is one-way communication. A similar situation occurs with telemarketers, who have your home phone number but invariably refuse to give you theirs. It is also unfair that spammers waste public time at little cost to themselves. Spammers steal time, which in today's world equates to money. Some find this a mild crime, like littering on the Internet, but when litter blocks the streets, it is of concern. Just as a cyberthief who takes a few cents from millions of bank accounts can steal a sizable sum, when spam affects millions of people the productivity loss is significant.

Spam may profit individuals, but it is unprofitable to society if it creates losses that exceed the profits it generates. If 90 percent of spammed people don't buy, do their losses balance the gains of the 10 percent who do? What if 99.9 percent don't buy? By one estimate, it costs about $250 to send one million e-mail messages, which cause about $2,800 in lost wages to society in general (www.ohio.com/mld/beaconjournal/business/5028845.htm). We seem to be well past the point at which spam's social losses outweigh its benefits.

In legitimate trade, people shop where they choose, but spam gives no choice—messages arrive unbidden, welcome or not. Successful societies make people accountable, but in online society spammers are not accountable. If spam communications are neither socially beneficial nor individually fair, they fail the test of legitimate communication on two counts. It is the communication architecture, created by software code, that enables this.

AN E-MAIL CHARGE?

Imposing a charge for e-mail messages would hit spammers' bank accounts. This would reduce spam by changing the game, but metering e-mail would also reduce general usage and benefit.[10] An Internet toll would add no new service, as e-mail already works without charges. The sole purpose of imposing an across-the-board charge would be to punish spammers, but it would slow the flow of information for everyone. Imposing a processing cost instead of a dollar charge would give the same effect. Such responses seem like burning down your house to prevent break-ins.

Who would set the rate, and who would receive each payment? If senders paid receivers, each e-mail would be a money transfer. The cost of administering such a system could outweigh its potential benefit. If e-mail providers received the charge, it would be an e-mail tax, but what global entity could legitimately claim it?

Making the Internet a field of profit could open it to corruption. Spam works because e-mail costs so little, but that is also why the Internet works. Fast, easy, and free communication has benefited us all.

We need a solution that reduces spam but leaves the Internet advantage intact. We need to reduce unfair communication, not charge everyone for what they already have.

LAW AND CYBERSPACE

Modern societies implement legitimacy through the law,[6] but passing laws in virtual worlds has several problems.[3] First, current law assumes a physical world architecture. But virtual worlds work differently, and physical laws may not transfer easily.

Second, virtual worlds frequently change faster than legislators can draft laws. Thus, new functionality—such as cookies—can outstrip the technology's assimilation into law. Spam has already shown it can mutate into new forms, like spim. Each spam variant would require new laws. Yet society takes years to pass them, while Internet applications can change in months.

Third, in cyberspace code *is* law, so the programmers who write the code make the rules. Giving spammers anonymity, or the power to hide their source, can negate any law.

Finally, jurisdiction limits laws, as attempts to legislate telemarketers illustrate. State laws against telemarketers were ineffective against out-of-state calls, and the US nationwide Do-Not-Call list will be ineffective against overseas calls. The many laws of the world's nations can be applied to their respective citizens, but not to a global Internet.

Thus, the long arm of the law struggles to reach into cyberspace. Normal prosecutions require physical evidence, an accused, and a plaintiff. Yet spam can begin and end in cyberspace, e-mail sources are easily spoofed, and, for spam, potential plaintiffs include everyone with an e-mail address.

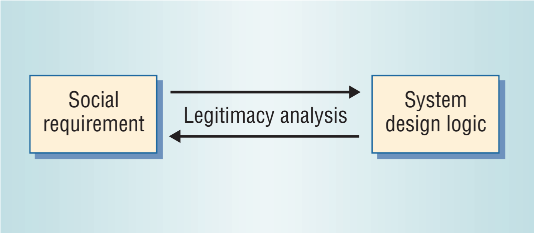

Figure 2. Legitimacy analysis. An online community can translate social statements into social-technical system design, and vice versa.

What penalties apply when each individual loses so little? Even if detected, a spam source can just reinvent itself under another name. Traditional law seems too physically restricted, too slow, and too impotent to deal with a dynamic, global information society.

AOL and similar providers propose antispam laws that exclude all spam but their own. Putting such policies into software practice would require a line of code equivalent to "If sender = AOL, then … else …." This is hardly the "veil of ignorance" from behind which Rawls proposes that laws should operate.[6] A legitimate social solution to spam should apply equally to all.

Also, the legal system operates after the fact—after conflict has arisen. To punish unfairness, it must first occur. Why let unsocial acts like spam develop, then punish them, if the system can be designed for beneficial fairness in the first place?

When societies address legitimacy instead of unfairness, they advance significantly, from commandments and punishments to visionary statements of liberty and equality. Cyberspace is an opportunity to apply several thousand years of social learning to the global electronic village. It makes little sense to design social software in a social vacuum, as if we knew nothing of the nature of society.

LEGITIMACY ANALYSIS

If legitimate interaction increases social prosperity, it seems sensible to design software to support it. As Figure 2 shows, legitimacy analysis aims to translate social requirements into information system specifications.[3] Conversely, it can translate information logic into social terms that a nontechnical community can discuss.

Legitimacy analysis focuses on what should be done socially, not what can be done technically—on social right, not software might. Given that people are more accountable for objects they own, this analysis seeks to specify online object ownership in fair and socially beneficial ways. If owning an object implies rights to act upon it, specifying information object ownership can define which online actors can do what actions to what entities. Technical systems can be designed to support what legitimate ownership implies.

Legitimacy analysis seeks to create consistency between what the social group believes and how the software behaves.

Analysis begins by defining the information system methods and objects. Next, the analysts state accepted community social rights, such as the concept from John Locke's second *Treatise on Government*, that creators have a natural right to own their creations,[7] which copyright also suggests. These social rights may vary between communities.

Then, analysis connects social rights to the IS object-method specification. For example, Locke's creator ownership suggests that those who create items in online bulletin boards should be able to edit them. In practice, existing software often fails to meet this social requirement. In WebCT, for example, item creators cannot edit items they add, although the space controller can. In summary, there are three steps:

1. Define information system objects and methods.
2. State legitimate ownership principles accepted by the community.
3. Analyze information object ownership based on steps 1 and 2.

Legitimacy analysis is a process, not a formula. It seeks to create consistency between what the social group believes and how the software behaves. This process requires the society or community to define legitimate ownership. If a community rejects a social principle, like creator ownership, it rejects its online implications.

Technology support for social rights does not mechanize social acts, because it is the interaction, not an individual act, that is legitimate or not. Specifying freedom, privacy, or democracy in information terms does not automate right acts—it allocates rights to act. For example, society grants people privacy, but does not force them to be private. Online legitimacy could mean someone *can* delete an item, but this does not mean they *must* delete it. The choice is with the person, not the code.

Applying this method to spam forces us to ask who owns the basic elements of e-mail communication, namely messages, channels, and addresses.

Who owns messages?

If an e-mail from one person to another constitutes an offer to own the sent message, not a command to take it, a receiver should be able to reject an e-mail at any time. With postal mail, receivers can write "Return to Sender" on a message and put it back in a mailbox. Returning mail also lets those who sent it know it was not received. A mail system contracts to deliver a message and if it does not, should return the message to the sender.

With e-mail, there is no return-to-sender function, so users use spam filters. The e-mail system reports "Message Delivered," even when receivers never see it because a spam blocker deleted it. Spammers don't know who reads their messages and who doesn't. On the other hand, if spam filters block genuine e-mail messages, senders may assume they are being ignored on purpose. These problems arise when the e-mail transmission process offers receivers no right of return.

This suggests the need for a button that users can press to reject e-mail. The recipient could still delete the message as before, after accepting it. A *rejected* e-mail also disappears from view, but it has not been deleted. Social logic suggests that rejected e-mail belongs not to the receiver who rejected it, nor to the system that delivered it, but to the sender who created it. Hence, as with postal mail, it should be returned to sender, with a failed-to-transmit comment.

Inherent in this proposal is that receiving rejected e-mail is a condition of transmission—a sender should not be able to submit messages to an e-mail system without also receiving messages from that system. This is a basic fairness concept. It means a person can be anonymous to all other people, but cannot be anonymous to, and cannot deny e-mail from, the e-mail system itself.

Hence, accepting rejected e-mail should be part of the transmission contract. Rejected spam would then return to the sender's computer, creating sender costs. While the e-mail system could send an outgoing message once and duplicate it a million times, a million rejections would return individually, consuming sender bandwidth and processing capacity. Spammers would know, factually and financially, who didn't want their mail. It would then pay them to reduce their lists, and also give them the information they need to do so.

The social logic behind this proposal is that messaging is a transfer of ownership where receivers can choose to receive messages or not. Implementing this requirement is an engineering problem, but an e-mail transmission system that controls both the pieces of the communication game and the board itself should be able to enforce a rule that those who send into the system must also receive from the same system.

Who owns communication channels?

Current online systems give any sender the right to open a transmission channel to send an e-mail message to another person. Yet, in physical society, the US Warren and Brandeis ruling[11] gives people the *right to be left alone*. This right not to communicate includes the right to remain silent. If someone knocks on the door, you need not answer. If the telephone rings, you need not pick it up.

This social right exists for reasons of practical public good, not for some abstract ethical reason. Without it, we would be denied a fair society and forced to bear the overhead of conversations that others initiate. This is precisely the problem spam creates. Extending this concept to e-mail suggests that while others can request a communication channel, we should not be forced to grant one.

Currently, spam arrives, wanted or not. Existing technology gives every sender the right to open and use a channel to our in-box. We all differentiate meeting someone new from someone familiar. We may talk to friends, but refuse to talk to strangers. E-mail blissfully ignores this basic social distinction when it gives every new e-mail message the rights of a familiar friend.

How to implement this social requirement online is less clear. One way to discriminate known from unknown e-mail is to recognize the conversation entity of the interaction itself, as Figure 3 shows. If e-mail systems recognized conversations, messages inside a conversation would be *known* and new conversations *unknown*.

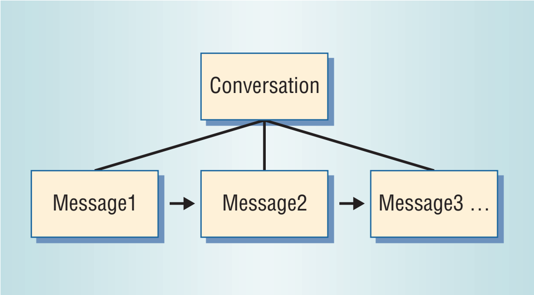

Figure 3. Conversation data entity. Initiating a new e-mail conversation would require an invitation to converse be accepted before the first message is delivered.

A new conversation would involve two separate communicative acts: opening a channel and sending messages. The first, a request to converse, could give channel details, like sender, title, and reciprocity—if replies are accepted. It would not include message content. Only if the receiver agreed would actual messages be sent. This would be like the premessage handshaking that occurs in data communication, or like starting a new thread in a bulletin board rather than adding to an existing one.

To senders, messaging would seem the same, as the system could handle the two steps. However, a conversation denial would be more than a message rejection—it would be an unwillingness to talk at all. To receivers, request-to-converse items could be displayed outside the normal in-box, or in the same in-box but colored differently. To engage in a conversation and receive content, the user could double-click the message. This would reduce spam content transmissions because most people don't click on spam.

Microsoft's plan to offer caller ID for e-mail is a step in the right direction because it gives receivers some channel information. But why not give receivers *all* channel details? Then they could, for example, choose only to receive messages from those who also receive. And why send message content before a channel is approved?

Current challenge-based spam defenses check reciprocity by requiring a reply before accepting an e-mail. This excludes most spammers, who hide themselves and won't accept replies lest they be spammed in return.

E-mail challenges work, but they need two sender transmissions per message, and they still send content. A conversation request system would only require double transmission on the first message and, even then, the message content is sent only once.

Since most people converse mostly with people they know—except for spam—double interactions would be low. E-mail replies would automatically go through and only new contacts would need to be confirmed. Receivers could automatically permit previously accepted senders, generating a list of known communicands. If anyone abused the privilege, the conversation could be closed, and further conversations marked "My Choice" or even "Always Reject."

This system design supports the view that communication is a privilege, not a right. A spin-off would be that e-mail could be structured around conversations, so when you receive an e-mail you could link to any previous conversation messages.

Who owns addresses?

In any online transaction, user data can be extracted into corporate marketing databases. Buying online books on specific topics, such as cancer, for example, could make you a marketing target for a stream of undesired e-mail offers. Yet society agrees that providing credit card data for a sale does not give other rights to that data—indeed, any other use is fraud.

Giving one right to information does not imply giving all rights. The social logic of privacy suggests that people should own their personal e-mail addresses for the same reason they own themselves. Yet making a computer copy gives all power to the copier, who receives a full duplicate, so spammers can do what they want with your e-mail address. It is the computer copy function that disables the social idea of privacy.

Society is trying, by legal means, to let individuals remove their contact details from marketing lists. The US's Do-Not-Call law would let people remove themselves from telemarketing databases, giving them a choice over their personal data. Yet laws without technology support have practical limitations—for example, people on a Do-Not-Call list either receive all marketing calls or none. In contrast, software could give each user a personal list, letting them accept some marketing calls but not others.

Despite the technology infrastructure's lack of support for this social concept, many companies voluntarily accept that people own their personal data. Their e-mail marketing offers a remove function, which lets people remove themselves from the corporate database. Some companies even give users direct editing access to their personal data, such as online personal banking.

People periodically change their physical and e-mail addresses, so why not let customers update their own data? This is as beneficial as having customers select their own goods in a supermarket. Requiring users to tell a company operator to change their data in a computer database is double handling. When customers own and maintain their personal data, company data maintenance costs diminish.

The nature of the computer copy function means personal address data can be sold or passed on. Even if the original company has a remove function, if it has sold your data, the damage is done. Further, any reply could confirm an active e-mail, so requesting to be removed could put you on even more spam lists.

Again, defining the social problem and devising a technical solution are two different issues. The privacy problem requires a solution that returns control of personal data to people, for example, once a sale completes. Others need to *use* our personal data, not *hold* it permanently, so a duplicate copy is overkill.

As with credit card data, businesses require temporary, not permanent, access. One way to give temporary access to data is to provide a link, not a duplicate. The computer copy function could have two forms:

- *full copy*, which creates a duplicate that can, for example, be used for backup, and
- *copy for use*, which creates a link for temporary use of source data.

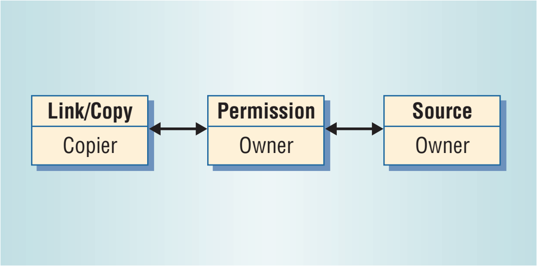

Figure 4. Copy for use. This computer function could let users permissions give conditional access to information they own.

As Figure 4 shows, when making a copy for use, the copier creates a link that points to an access permission from the original owner, which in turn points to the source data, such as an e-mail address. An e-mail marketing address list would then be a list of links, not source data. The list owner could delete any unwanted link. Customers could see the lists they are on by looking at their permissions. Deleting a permission would effectively remove the owner from that list, unless a joint transaction is in process.

The issues are complex, but data access by links could support the social idea of online privacy. It could give people direct ownership of personal data, avoiding subtle legal distinctions, like what is and isn't spam and whether the sender's purpose is charitable or political. Software could let all people own all their personal data online.

Although spam's causes are social, in virtual society it is technology—particularly the core Internet communication architecture—that defines the social interaction environment. Currently, the transmission process gives senders all rights and receivers none, thereby encouraging spam.

The technology that creates the communication playing field cannot be indifferent to whether or not that field is level. Fair communication requires a balance of sender and receiver rights. If code, not law, rules in cyberspace, then code must support this balance or it will not happen. Spam illustrates what happens when technology ignores fair communication.

If we are to close the social-technical gap, technologists must help. Cyberarchitecture will support cybersociety when system designers develop legitimacy specifications in tandem with technical ones. If software is to support society, not undermine it, legitimacy concepts must be taught in core information system design courses, as social-technical requirement. Turning social requirements into design specifications presents a daunting task, but today's computer systems—equal parts social and technical devices—require online fairness as much as bandwidth. Perhaps spam is a wake-up call, a reminder to respect social as well as technical reality. ■

References

1. A. Weiss, "Ending Spam's Free Ride," *netWorker*, vol. 7, no. 2, 2003, pp. 18-24.
2. S.J. Vaughan-Nichols, "Saving Private E-mail," *IEEE Spectrum*, vol. 40, no. 8, 2003, pp. 40-44.

3. B. Whitworth and A. deMoor, “Legitimate by Design: Towards Trusted Virtual Community Environments,” *Behaviour & Information Technology*, vol. 22, no. 1, 2003, pp. 31-51.
4. A. Hamilton, “You’ve Got Spim! Spam Not Annoying Enough? Now Junk Instant Messages Are on the Rise,” *Time*, 23 Feb. 2004, p. 1.
5. M.S. Ackerman, “The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility,” *Human-Computer Interaction*, vol. 15, nos. 2 and 3, 2000, pp. 179-203.
6. J. Rawls, *Justice as Fairness*, Harvard Univ. Press, 2001.
7. J. Somerville and R.E. Santoni, eds., *Social and Political Philosophy: Readings from Plato to Ghandi*, Anchor Books, 1963.
8. P. Regan, *Legislating Privacy, Technology, Social Values, and Public Policy*, Univ. of North Carolina Press, 1995.
9. F. Fukuyama, *The End of History and the Last Man*, Avon Books, 1992.
10. R.E. Kraut et al., “Markets for Attention: Will Postage for E-mail Help?” *Proc. Computer Supported Cooperative Work* (CSCW 02), ACM Press, 2002, pp. 206-215.
11. S.D. Warren and L.D. Brandeis, “The Right to Privacy,” *Harvard Law Review*, 1890, vol. 4, no. 5, pp. 193-220.

***Brian Whitworth** is an assistant professor in the New Jersey Institute of Technology’s Information Systems Department. His research interests include the definition of social-technical computing. He received a PhD in information systems from Waikato University, New Zealand. Contact him at bwhitworth@acm.org.*

***Elizabeth Whitworth** is a graduate student in Carleton University’s Psychology Department. Her research interests include human-computer interaction. She received a BS in human-computer interaction from the New Jersey Institute of Technology. Contact her at ewhitwor@connect.carleton.ca.*