NRC Publications Archive
Archives des publications du CNRC

Building Trustworthy Software Agents

Patrick, Andrew

This publication could be one of several versions: author’s original, accepted manuscript or the publisher’s version. / La version de cette publication peut être l’une des suivantes : la version prépublication de l’auteur, la version acceptée du manuscrit ou la version de l’éditeur.

Publisher’s version / Version de l'éditeur:

IEEE Internet Computing, 6(5), 2002

NRC Publications Record / Notice d'Archives des publications de CNRC:
https://nrc-publications.canada.ca/eng/view/object/?id=c08240b3-47ee-4a9f-a8df-bf949244eeec
https://publications-cnrc.canada.ca/fra/voir/objet/?id=c08240b3-47ee-4a9f-a8df-bf949244eeec

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L’accès à ce site Web et l’utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D’UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at
PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n’arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research
Council Canada

Conseil national
de recherches Canada

Institute for
Information Technology

Institut de technologie
de l'information

NRC-CNRC

*Building Trustworthy Software Agents **

Patrick, A.
November 2002

* published in IEEE Internet Computing, 6(5). (In press). NRC 44958.

Copyright 2002 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Canada

Building Trustworthy Software Agents

ANDREW S. PATRICK
National Research Council of Canada

To appear in IEEE Internet Computing, Nov./Dec. 2002.

Feelings of trust and perceptions of risk combine in opposite directions to determine a user's final acceptance of an agent technology.

Intelligent software agents are drawing increased interest within the software community. Such agents would act autonomously on behalf of their owners, perhaps handling routine communications, making travel arrangements, or shopping for the best prices. This interest in agents is a result of frustration with using direct manipulation interfaces (such as mice and GUIs) for increasingly complex tasks, information overload, and a need to exploit the rapidly expanding network of distributed information and services. These trends are leading to a desire for software that explores, anticipates, adapts, and actively assists its users in ways not possible today. In addition, software that acts on behalf of a user can help protect the user's identity and privacy. By including privacy protection measures and having an agent perform tasks on behalf of a user, anonymity can be maintained and the agent can share only the personal information that the user desires.

Today, many computer users execute complex tasks across a rapidly expanding network of distributed information and services using only direct manipulation tools such as mice and graphical user interfaces. The frustration that results is leading to a desire for software that explores, anticipates, adapts, and actively assists its users. The software community is addressing this desire by developing intelligent software agents that act autonomously on their owners' behalf, perhaps handling routine communications, making travel arrangements, or shopping for the best prices. With privacy protection measures, the person employing the agent can maintain anonymity and control the personal information the agent shares.

As agents become more active and sophisticated, however, the implications of their actions become more serious. With today's GUIs, user and software errors can often be easily fixed or "undone." An agent performing actions on behalf of a user could make errors that are very difficult to "undo" (such as making faulty airplane reservations) and, depending on the agent's complexity, it might not be clear what went wrong. Moreover, for agents to operate effectively and truly act on their users' behalf, they might need confidential or sensitive information. This includes financial details (such as credit cards numbers) and personal contact information (such as telephone numbers). Thus, along with the excitement about agents and what they can do, there is concern about the resulting security and privacy issues. It is not enough to assume that well-designed software agents will provide the security and privacy users need; assurances and assumptions about security and privacy need to be made explicit.

This article proposes a model of the factors that determine agent acceptance, based on earlier work on user attitudes towards e-commerce transactions, in which feelings of trust and perceptions of risk combine in opposite directions to determine a user's final acceptance of an agent technology.

Agents and Trust

Negroponte[1] describes the ideal agent as the equivalent of "a well-trained English butler" who knows your needs, likes, and habits. He goes on to describe the privacy issues:

> All of us are quite comfortable with the idea that an all-knowing agent might live in our television set, pocket, or automobile. We are rightly less sanguine about the possibility of such agents living in the greater network. All we need is a bunch of tattletale or culpable agents. Enough butlers and maids have testified against former employers for us to realize that our most trusted agents, by definition, know the most about us. (p. 62)

For agents to be successful, users will have to trust them with private information, and agents will have to handle that information in a secure fashion. Trust becomes very important if an agent's actions can cause its user physical, financial, or psychological harm.[2] Thus, users must be confident that the agent will do what they ask, and only what they ask.

It is clear that a trusting relationship must develop between the user and the agent. Because trust between user and agent is so important, it is useful to examine the nature of trust.

What Is Trust?

Most generally, trust is "a generalized expectancy… that the word, promise, oral or written statement of another individual or group can be relied upon."[3] In the context of software agents, this means that the agent can be relied upon to do as instructed. But trust is more than that; it is "the condition in which one exhibits behavior that makes one vulnerable to someone else, not under one's control."[4] Without vulnerability, there is no need for trust. In the context of software agents, trust means no longer controlling the software directly, letting the process act on one's behalf and accepting the risks this might entail. Bickmore and Cassell describe trust as "people's abstract positive expectations that they can count on [agents] to care for them and be responsive to their needs, now and in the future."[2]

This concept of making oneself vulnerable to accomplish a goal is essential for understanding trust. Without trust, virtually all of our social relationships would fail, and it would become impossible to function normally. If we can't trust oncoming drivers to stay in their lane, then it would become impossible to drive. If we don't trust the shopkeeper to deliver the goods we pay for, then simple purchases would become very awkward. We make ourselves vulnerable to others every day, but we are usually comfortable in doing so because we trust that their actions will not be inappropriate or harmful. Bickmore and Cassell describe trust as a process of uncertainty reduction.[2] By trusting others to act as we expect them to act, we can reduce the number of things we have to worry about.

Taking a computer science approach, Marsh has defined trust in terms of the behavior of the person doing the trusting.[5] Thus, trust is the behavior X exhibits if he or she believes that Y will behave in X's best interest and not harm X.

For our purposes, then, trust can be defined as users' thoughts, feelings, emotions, or behaviors that occur when they feel that an agent can be relied upon to act in their best interest when they give up direct control.

Interactions Twice Removed

Part of the difficulty in trusting software agents is that users often end up working on a task that is *twice removed* from the interface. Consider a job seeker. In a traditional, nonremoved job search, an individual talks to employers directly, perhaps by visiting their offices. In Figure 1a, the job seeker is interacting directly with the potential employer to get information about the position. A more modern job search takes place through a computer, where the job seeker interacts with a computer program, often a Web browser, to view information created by the employer; thus, the interaction between the job seeker and the employer in Figure 1b is *once removed.* (Riegelsberger and Sasse refer to this as a *disembedded transaction.*[6])

With a job-searching agent, the job seeker would interact with a computer program, perhaps an agent control interface, to provide instructions to the agent. The agent would then search the Internet and gather information provided by the employer. There is no direct connection between the user and the job-seeking activities (Figure 1c). Thus, the interaction between the job seeker and the potential employer is twice removed (or dis-disembedded). Research has shown that developing trust can be difficult during once-removed interactions, let alone twice-removed interactions.

There are many valid reasons why users might hesitate to trust personal software agents. Interactions involving personal information might be riskier than financial interactions, because personal assets like self-respect, desirability, reputation, and self-worth can be more valuable than money.[7] Also, because agents operate autonomously, outside the user's vision and control, things might go wrong that the user does not know about or cannot correct.

Youll has also described the issues involved in trusting agents.[8] First, the user must make their instructions clear to the agent. This instructing phase could fail for a number of reasons:

- the user does not clearly define the instructions,
- the agent does not fully understand the instructions, or
- the user and the agent interpret identical instructions differently.

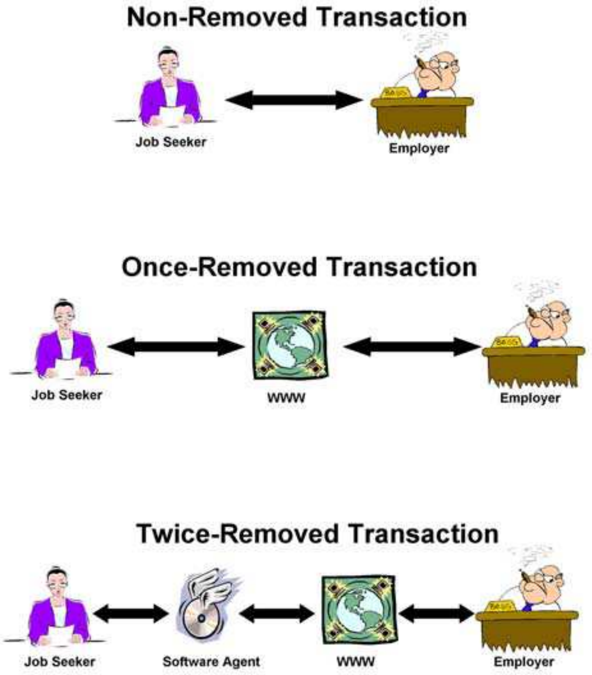

Figure 1. Levels of interaction in traditional and modern job searches. (a) A direct transaction between a job seeker and an employer; (b) A once-removed transaction where a job seeker views an employer's Web site; (c) A twice-removed transaction where the job seeker's agent views the employer's Web site.

Second, if the agent understands the instructions, the user must be confident that it will execute the instructions properly, and will only perform the tasks the user intended. Third, the user must be confident that the agent will protect private or sensitive information. Finally, the user must be confident that the agent will not be attacked or compromised, such as through "hacking" or "sniffing." With all of these concerns, developing a trusting relationship between users and their agents is a difficult task.

On the other hand, the twice-removed nature of the interactions between end user and task makes agents well-suited for tasks requiring high degrees of privacy. An agent can establish its own identity on the network, protecting the identity of the end user.[8] The Lucent Personalized Web Assistant (www.bell-labs.com/project/lpwa), for example, acted as a proxy for users who wanted to navigate the Web without revealing their true identities. Such services can even go so far as to establish new pseudonyms for every transaction, making it very difficult to establish a link back to the user.

Agents are also well-suited for situations in which interaction policies must be established and followed. Because software agents are embodied in explicit computer code, it is possible to establish and follow clearly defined privacy policies, rather than rely on heuristics or emotions.

Building Successful Agents

An important contribution to research on e-commerce trust is the path model of e-commerce customer loyalty developed by Lee, Kim, and Moon.[9] They describe how the amount of trust instilled in the user **versus** the cost perceived by the user will determine attitudes toward e-commerce. Trust and cost combine, in opposite directions, to determine overall acceptance.

Figure 2 shows an extended model to explain agent acceptance. Here, user acceptance of agent technology is determined by the combination of trust and perceived risk. The contributing factors Lee, Kim, and Moon identified are included, along with factors identified by other researchers.

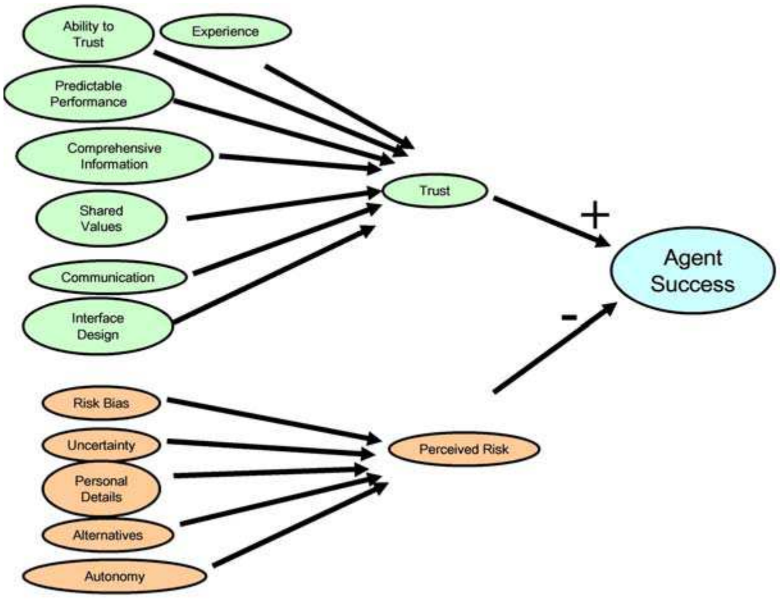

Figure 2. A model of agent success. It builds on the Lee, Kim, and Moon model, adding factors of trust and risk important for software agents

An important feature of this e-commerce model, and of the model of agent acceptance this article proposes, is the separation of trust from perceived risk. Feelings of trust and risk can be established quite independently, and together they determine the agent's success. Trust contributes to the agent's acceptance, while risk contributes to its rejection. The two factors interact, so that agents instilling a low degree of trust can still succeed if the perceived risk is also low. On the other hand, in very risky situations it is possible that no amount of trust will offset the perceived risk, and the user will never accept the agent. Rotter, in his review of the social psychology of interpersonal trust, supports this idea that trust and risk are separate concepts, and that both contribute to an individual's resulting behavior.[3] Grandison and Sloman also describe trust and risk as opposing forces that combine during decision-making about a service or an e-commerce transaction.[10]

An important feature of the proposed agent acceptance model is that the risk being described is the risk *perceived by the user*. This perception might or might not be related to the actual risk of the technology employed in the agent system. For example, a user's credit card number might be encrypted with a very strong encryption technique, but if the user believes that the number will be disclosed inappropriately, this fear contributes to the perceived risk and works against acceptance of the agent technology.

Trust Factors

According to the model, users will be more accepting if they developing feelings of trust. Several factors affect how much trust a user will place in an agent.

Ability to trust. A user's general ability to trust affects whether or not the user will accept the agent. A number of researchers have suggested that people have a baseline attitude when they approach any trust situation, and that some people have a higher baseline level of trust than others. For example, Marsh describes "basic trust" as a person's general propensity to trust or not to trust.[5] This basic trust is part of an individual's personality, and is one of the factors that contribute to making decisions about trust. Similarly, Rotter showed that there is a generalized trust that is "a relatively stable personality characteristic."[3]

To study the issue of trust on computer networks, Cranor, Reagle, and Ackerman surveyed Internet users about their attitudes toward privacy and trust.[11] The survey respondents were then classified into groups that differed in their concerns about online privacy. The first group (27 percent) was only "marginally concerned" with online privacy and was quite willing to provide personal information when visiting Web sites. This group did have some concerns, such as the desire to remove themselves from marketing mailing lists, but was generally quite trusting. The second group (17 percent) was at the opposite extreme, labeled "privacy fundamentalists." These users were extremely concerned about privacy and were generally unwilling to provide any information to Web sites, even when privacy protection measures were in place. Cranor et al. labeled the third and largest group (56 percent) the "pragmatic majority" because they had some concerns about privacy, but also had developed tactics for dealing with those concerns. For example, they would often look for privacy protection methods or statements when navigating the Web.

When building agent systems that users will have to trust, developers should consider the fact that some users will trust an agent system with little reassurance of the privacy protection measures in place, while others will be very reluctant to trust. Thus, interfaces must be flexible enough to provide more information and reassurance for users that require it.

Experience. Clearly, experience can change users' willingness to trust.[5,12] If they have been harmed in some way, or if they have found an agent to be incompetent, either in the information provided or in carrying out its actions, they might become less trusting. This change in trust might be specific to the situation or it might be a change in their general ability to trust. Changes in trust can also come about indirectly because of the experiences or recommendations of others.[10] This means that trust can be "social," transmitted from user to user.

To ensure that users have positive experiences, agent system designers should provide ample information to users about how the agent operates. In addition, designers should support a sharing function that lets users relate their experiences and spread their trusting attitudes (assuming their experiences are positive). This might mean collecting testimonials or anecdotes to share with other users.

Predictable performance. Bickford[13] describes three important principles for predictable performance and its role in building trust: consistency, aesthetic integrity, and perceived stability. Users are more likely to trust systems and interfaces that perform reliably and consistently. To ensure a coherent and predictable interface, a designer might adopt a style guide or interface guideline to be used in all parts of the system. Predictable performance also involves response time: users prefer consistent response times to variable response times.

Developers should also make sure the system behaves consistently and seems stable. Human factors evaluation techniques are useful for testing these aspects of a design (see the "Checking Your Work" section, below).

Comprehensive information. Another important factor in determining whether users will trust a system is the amount of information it provides. Users are more likely to understand, and thus trust, systems that provide comprehensive operations information. According to Norman,[14] users will develop mental models and assumptions about a system even when no information is provided, and these models can be wrong. Developers should therefore explicitly guide model development by carefully explaining and depicting how the agent service works, so that its role and operation are obvious. This might mean allowing users to observe and track an agent's actions, both in real-time and after the fact. In addition, interfaces should allow users to view and alter the information stored by agents.

Shared values. Users are more comfortable trusting agents with common values. That is, to the extent that users feel an agent values what they value, they will have more trust in the agent. Informal interpersonal interactions — such as social conversations that occur in hallways or during coffee breaks — often build these shared values. Bickmore and Cassell tested the role of small talk in building trustworthy agents.[2] A test condition that involved informal social dialogues with an agent led to higher levels of trust and willingness to share personal information for extroverted users (it is not clear why this effect was not found for introverted users). Values between agents and their users can also be shared explicitly. Agents could articulate their privacy policies, for example, so users can compare the current policies with their privacy concerns.[7]

Communication. The amount and effectiveness of communication between the agent and the user also determine the level of trust. Norman argues that continual agent feedback is important for success.[14] An agent should repeat a user's instructions so it is clear that the agent understood, for example. Also, error messages should explicitly state what was understood and what needs to be clarified. In addition, the agent service should make clear the agent's capabilities and limits.

Interface design. The final trust factor is the design of the interface itself, the look and feel of the software used to control the agent. This area includes such factors as appearance, functionality, and operation. Cheskin,[7] Kim and Moon,[15] and Riegelsberger and Sasse[6] have each examined interface designs that can communicate trust. Their research was in the context of e-commerce Web sites, but the lessons apply to agent systems as well. The results can be summarized as a list of fundamental interface characteristics that can communicate trust:

- *Brand.* Trust can be influenced by users' familiarity with and feelings about a service provider. Providers that are already trusted in other contexts, such as real-world stores, can also be trusted in the new context.
- *Navigation.* The ease of finding things, which results from clear, logical presentation and consistent design, can help to generate trust. Also, providing a mechanism to recall or undo an action can lead to more confidence in an interface.
- *Fulfillment.* A clear and traceable process for completing a task enhances trust, as will letting users see the status of actions they requested.
- *Presentation.* Clearly presented material, a clean and functional layout, and a professional presentation increase feelings of trust. The agent interface should appear professional and official, like money or certificates. Other recommendations are to use graphics and images, cool colors, pastel shades, and low brightness.
- *Certifications and logos of assurance.* Including icons and text that represent seals of approval or assurances of safety can increase feelings of trust.

A controversial issue in designing trustworthy interfaces is the value of anthropomorphism. Does a human-like interface, perhaps with an animated character and conversational interaction, lead to more feelings of trust? Bickmore and Cassell argue that an animated character capable of small talk can lead to shared values and higher trust.[2] However, others have argued that such anthropomorphism can lead to disappointment if the interface does not live up to expectations.[6,14,16] If the agent cannot really behave like a human, then having a human-like interface might actually diminish trust rather than build it. Sometimes users question the motivation of human-like "guides," even becoming angry if the character does not behave as expected.[16] Thus, developers of agent systems should only consider anthropomorphic interfaces if they truly reflect the abilities and behaviors of the agent system. Since such human-like abilities are a long way off, it is probably most appropriate to avoid anthropomorphism.

Perceived Risk Factors

The other side of agent success is perceived risk. Other things being equal, users will be more willing to use agent systems if they perceive lower risks. A number of factors influence the amount of perceived risk.

Risk perception bias. As with basic trust, users can have a basic or baseline level of perceived risk. This is probably best described as a bias to perceive situations as being risky or risk-free. This bias in risk perception breaks down into four basic **attitudes**:[12]

1. *Fatalism.* Users have no control and risk decisions are out of their hands.
2. *Hierarchy.* Controls and regulation should contain risks.
3. *Individualism.* Risks should be taken when appropriate for the individual.
4. *Enclave.* Risks are systemic and should be handled with pressure, dissent, and market systems.

Agent system designers should design system features that address each of these approaches to risk assessment. For example, to please individualist users, an agent system might explain how users can control the risks they are taking. To accommodate hierarchical thinkers, a system can include information about its controls and regulations. Finally, allowing fatalistic users to share information and experiences with the system developers and each other can lead to feelings of empowerment and fewer concerns about risk.

Uncertainty. The more users know about a system and how it operates, the less they worry about taking risks (assuming that they learn positive things). Thus, by reducing uncertainty, we can reduce risk perception. This is highly related to the comprehensive information and communication factors discussed in the "Trust factors" section.

Personal details. An obvious factor in risk perception is the amount of sensitive information being provided. The more personal details a user gives to an agent, the more that user's perceptions of risk are likely to increase. System developers should ask only for information necessary to do the job, where possible avoiding information that might be especially sensitive. Exactly what information users consider sensitive might require some investigation. For example, users might consider phone numbers more sensitive than e-mail addresses because unwanted phone calls are more intrusive than unwanted e-mail messages.

Alternatives. Another factor that can lead to feelings of risk is a lack of alternative methods to perform a task. For example, if the only way to buy a special item is to use a new agent technology, users might feel they are taking more risks than if there were multiple methods (nonagent Web interfaces, phone calls, and store visits, to name a few). Similarly, if there is a sole supplier of a service, users may feel they are at more risk from exploitation than if there are multiple suppliers. In the shopping scenario, for example, users might be more comfortable if they have multiple shopping agents to choose from.

Autonomy. Perhaps the most important factor in determining users' feelings of risk towards an agent technology is the degree of autonomy granted to the agent. Agents can range from low-risk advice-giving systems to higher-risk independently acting agents. Advice systems can stay in close contact with the user and receive further instructions as they operate. Further, advice agents can learn by example as they monitor the advice their users accept. In the shopping example, it might be most appropriate for the agent to suggest purchases that the user should consider, rather than completing the transaction autonomously.

Checking Your Work

Most standard human factors evaluation techniques are appropriate when developing agent technologies. Although an exhaustive review of these techniques is beyond the scope of this article, a brief overview illustrates their applicability to agent design.

The first evaluation technique to consider is *qualitative research*. Here researchers talk to potential users about a variety of topics that are important during the design phases. These conversations can be one-on-one interviews or focus group sessions. For example, researchers might conduct a needs analysis to determine the tasks the agent should perform and how it should accomplish them. They might also questions users about their preferences and concerns, which can be particularly important for discovering concerns about privacy and sensitive information.

Another technique valuable during the early design stages is *heuristic evaluation*, in which researchers with expert knowledge examine a prototype system against a set of criteria. These criteria can come from general background knowledge about human factors, or from specific recommendations.

A related technique is a *cognitive walk-through*, where users are brought in and asked to interact with a system under development. Here users are asked to think aloud and provide comments as they try out the system. They might be given specific tasks to perform and questions to guide their comments. Heuristic evaluations and walk-throughs can be very powerful for determining potential problems before much effort is spent building a complete system.

The final evaluation technique is a formal *empirical test*. In these tests, users interact with a complete system, allowing researchers to record specific performance measures under controlled conditions. For example, researchers can note the number and type of errors users make, or the time they need to complete a task. Empirical tests can be expensive and time-consuming, so they are often reserved for the final stages of product development.

Conclusions

These guidelines for building trustworthy software agents are not just theoretical — they can help in the development of real-world systems. For example, I am currently working with the PISA Consortium (Privacy Incorporate Software Agents; www.pet-pisa.nl) to develop a prototype job-seeking service. Users need to entrust agents with such personal information as their name, phone number, e-mail address, and desired salary. The PISA system will succeed only to the extent that users trust the agents to provide a valuable service while protecting their privacy. Testing, to be conducted next year when the prototype is complete, will determine whether the developers were successful in building trust.

References

1. N. Negroponte, "Agents: From Direct Manipulation to Delegation," *Software Agents*, J.M. Bradshaw, ed., AAAI Press/MIT Press, Menlo Park, Calif., 1997.
2. T. Bickmore and J. Cassell, "Relational Agents: A Model and Implementation of Building User Trust," *Proc. Human Factors in Computing Systems* (SIGCHI 2001), ACM Press, New York, 2001, pp. 396-403.
3. J.B. Rotter, "Interpersonal Trust, Trustworthiness, and Gullibility," *American Psychologist*, vol. 35, no. 1, Jan. 1980, pp. 1-7.

4. D.E. Zand, “Trust and Managerial Problem Solving,” *Administrative Science Quarterly*, vol. 17, 1972, pp. 229-239.
5. S. Marsh, “Formalizing Trust as a Computational Concept,” doctoral thesis, University of Stirling, Scotland, 1994; available from http://www.stephenmarsh.ca/
6. R. Riegelsberger and M.A. Sasse, “Trustbuilders and Trustbusters: The Role of Trust Cues in Interfaces to E-Commerce Applications,” *Proc. 1st IFIP Conf. on E-commerce, E-business, E-government* (i3e), Kluwer Academic Publishers, Dordrecht, the Netherlands, 20\01; available at www.cs.ucl.ac.uk/staff/jriegels/trustbuilders_and_trustbusters.htm
7. “eCommerce Trust Study,” research report, Cheskin and Studio Archetype/Sapient, San Francisco, Calif., Jan. 1999; available at www.cheskin.com/think/studies/ecomtrust.html
8. J. Youll, “Agent-Based Electronic Commerce: Opportunities and Challenges,” *Proc. 5th Int’l Symp. on AutonomousDecentralized Systems*, IEEE CS Press, Los Alamitos, Calif., 2001, pp. 146-148; available at www.media.mit.edu/~jim/projects/atomic/publications/youll-mit-isads.pdf
9. J. Lee, J. Kim, and J.Y. Moon, “What Makes Internet Users Visit Cyber Stores Again? Key Design Factors for Customer Loyalty,” *Proc. Human Factors in Computing Systems* (SIGCHI 2000), ACM Press, New York, 2000, pp. 305-312.
10. T. Grandison and M. Sloman, “A Survey of Trust in Internet Applications,” *IEEE Comm. Surveys*, 4th quarter 2000; available at www.comsoc.org/livepubs/surveys/public/2000/dec/grandison.html
11. L.F. Cranor, J. Reagle, and M.S. Ackerman, "Beyond Concern: Understanding Net Users' Attitudes About Online Privacy,". In Ingo Vogelsang and Benjamin M. Compaine, eds. The Internet Upheaval: Raising Questions, Seeking Answers in Communications Policy. Cambridge, Massachusetts: The MIT Press, 2000, pp. 47-70
12. P. 6, “Can We Be Persuaded to Become PET-lovers?,” OECD Forum Session on Privacy-Enhancing Technologies, Organization for Economic Cooperation and Development, Paris, 2001; available at www.olis.oecd.org/olis/2001doc.nsf/c5ce8ffa41835d64c125685d005300b0/52810d7d5d053174c1256b170037d8cc/$FILE/JT00117775.PDF
13. P. Bickford, “Human Interface Online: A Question of Trust,” View Source, Sun Microsystems, Santa Clara, Calif.; available at http://developer.iplanet.com/viewsource/bickford_trust.html
14. D.A. Norman, “How Might People Interact with Agents?,” *Software Agents*, J.M. Bradshaw, ed., AAAI Press/MIT Press, Menlo Park, Calif., 1997; available at www.jnd.org/dn.mss/agents.html
15. J. Kim and J.Y. Moon, “Designing Toward Emotional Usability in Customer Interfaces — Trustworthiness of Cyber-banking System Interfaces,” *Interacting with Computers*, vol. 10, 1998, pp. 1-29.
16. T. Erickson, “Designing Agents as if People Mattered,” *Software Agents*, J.M. Bradshaw, ed., AAAI Press/MIT Press, Menlo Park, Calif., 1997.

Andrew Patrick is a senior scientist at the National Research Council of Canada. He is currently conducting research on human-computer interface issues for trustworthy software agents and the human factors of security systems. He holds a PhD in cognitive psychology from the University of Western Ontario.

Readers can contact the author at andrew.patrick@nrc.ca.