

# Privacy, Trust, Agents & Users: A Review of Human-Factors Issues Associated With Building Trustworthy Software Agents

Andrew Patrick

Institute for Information Technology  
National Research Council of Canada  
Andrew.Patrick@nrc.ca

Version 1.6

March 18, 2002

## ABSTRACT

*Developing software agents that users will trust with sensitive information is very difficult. The nature of the twice-removed relationship between the users and their tasks is described, and the concerns and opportunities of this relationship are explored. A model of agent acceptance is then proposed based on earlier work on user attitudes towards e-commerce transactions. This model describes how feelings of trust and perceptions of risk combine in opposite directions to determine a user's final acceptance of an agent technology. Specific factors that contribute to trust and risk are reviewed using both descriptive and prescriptive approaches where relevant research is reviewed and specific system and interface design features are recommended.*

## 1. INTRODUCTION

There is increasing interest within the software community in developing intelligent software agents. This interest is a result of a growing frustration with using direct manipulation interfaces (i.e., mice, 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 (Bradshaw, 1997). In addition, software that acts on behalf of a user may be useful for protecting the identity and privacy of the user. 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.

An agent can be defined as an entity that operates autonomously without direct user control, but under commands previously issued by the user. A classic example of an agent is a butler or secretary who makes decisions and commitments on behalf of their bosses. There is often a close relationship between the agent

and their "user" so that, for example, the butler learns his boss' likes and habits and is able to anticipate and respond effectively, even if the boss is not present. The idea behind software agents is to capture the power and effectiveness of human-human, boss-butler relationships in human-computer, user-agent software systems. The goal is to develop software that acts like a butler or secretary, taking actions, anticipating problems, making decisions, and improving the life of the user (Negroponte, 1997).

The level of autonomy and independence of an agent can be described in terms of "active" agents that independently perform actions on behalf of the users (e.g., make purchases or business commitments), and "advice" agents that merely provide advice or suggestions for the users to consider. In addition, the sophistication of agents can range from simple scripts that run periodically on a user's machine, to complex programs that travel autonomously across a network while performing remote tasks on behalf of the user (mobile agents).

Most of the software agents in use today (see [www.agentland.com](http://www.agentland.com) for examples) are relatively simple advice systems. For example, Lieberman et al.'s (2001) Letizia system is an agent that autonomously searches for WWW pages based on what users are currently viewing in their browser. The agent simply presents pages that may be of interest and the user can choose to attend to or ignore these suggestions. One example of an active agent in use today is the proxy bidding service found on eBay.com (<http://pages.ebay.com/help/buyerguide/bidding-prxy.html>). This agent autonomously submits bids on behalf of the user according to a maximum price specified when the agent is launched. This agent truly acts on behalf of the user because it makes financial commitments without direct user control.

### 1.1. Concerns About Agents

As agents become more active and more sophisticated, the implications of their actions and any errors they make will become more serious. With today's GUI interfaces, errors made by the user or software 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" (e.g., making a purchase commitment) and, depending on the complexity of the agent, it may not be clear what went wrong. For example, the agent may have failed to "understand" the instructions, or made an error during execution (Erickson, 1997).

Moreover, in order for agents to operate effectively and truly act on behalf of their users, they may be given information that is confidential or sensitive. This includes financial details (e.g., credit cards numbers) and personal contact information (e.g., telephone numbers) that should not be shared indiscriminately on public networks. Thus, along with the excitement about agents and what they can do, there is concern about the security and privacy issues that will result. Negroponte (1997) describes the ideal agent as the equivalent of "a well-trained English butler" who knows your needs, likes and habits. Negroponte 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)

In order for agents to be accepted, users will have to trust them with private information, and the agents will have to handle that information in a secure fashion. This trust becomes very important where users may suffer physical, financial, or psychological harm because of the actions of an agent (Bickmore & Cassell, 2001). 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 to the user. Without this information the users may assume that systems are not secure and private when they are, or that their privacy is being protected when it is not. For example, users of corporate e-mail systems often assume a high degree of privacy, when in fact there can be very little. Courts have repeatedly ruled that employers can use private e-mail messages and such

messages have been used in court cases (Weisband & Reinig, 1995). Developing and maintaining the appropriate levels of trust will be very difficult. The focus of the current paper is to review the human-factors issues relevant to developing trusted agents. The interface and system design issues that can lead to trust will be reviewed, along with the factors that increase perceived risk. The combination of trust and risk will determine the willingness of users to accept and use agent technologies.

### 1.2. The PISA Project

The current paper is being prepared as part of a three-year European/Canadian research project called PISA (Privacy Incorporated Software Agent, see <http://www.pet-pisa.nl>). The aim of PISA is to address the privacy and trust concerns of agent technologies directly. Privacy Enhancing Technologies (PET) will be developed during the project to demonstrate that a secure technical solution can protect the privacy of users when they use intelligent agents. Moreover, this privacy protection will meet the requirements being established by regulatory bodies in Europe and elsewhere, and seed the software community and standards groups with privacy-enhanced software and systems. One component of the PISA project is to ensure that human-factors issues are properly addressed, and this paper is the first step in that process. Other human factors work will focus on specific interface design issues, implementation of prototypes and modules, and assessments of the usability and acceptance of the resulting agent software.

### 1.3. A Reference Case: The Job-Searching Agent

To facilitate discussions, the PISA researchers have defined reference agents to be described and explored in detail. One such case is a job-searching agent, which will search the Internet for jobs on behalf of its users. The agent will carry information about the user, including sensitive information such as the current employer, salary history and salary expectations. The agent will also know the preferences and career aspirations of the user, and will use this information when traveling to different job search sites on the Internet. The agent must match the requirements and characteristics of potential new jobs with the information it knows about its user. It may even modify the description of its user to fit the requirements of the position (e.g., emphasizing managerial experience for a business position or emphasizing publications for an academic position). Moreover, it will most often do this without revealing

the full details about the identity of the user. This is important when users do not want their current employer to know they are searching for a new position. Other information, such as salary expectations, will have to be used when searching for suitable positions, but not be revealed to potential employers until an appropriate time.

This reference case illustrates a number of security and privacy concerns, including, for example:

- What information does the agent store and how does it control distribution of that information?
- How does the agent decide what private information is shared at each stage in the interaction, and with whom?
- How can it be guaranteed that the information shared by the agent with another entity will not be given to a third party which has not received permission from the agent to read this information?

The human-factors issues associated with agent technologies can also be explored in this reference scenario. Some obvious example questions are:

- What interfaces are appropriate for instructing agents about the information to share, and when?
- How can the system provide reassurance that a user's instructions were followed? How can users look for errors or problems?
- What interface needs to be built so users can track the actions of their agents?

This reference case will be used throughout the remainder of the paper to illustrate human factors problems and solutions.

## 2. AGENTS AND TRUST

It is clear that a trusting relationship must develop between the user and the agent. Users must be confident that the agent will do what they have asked, and only what they have asked. Moreover, to be effective the agent must be trusted with sensitive information, and use it only in appropriate circumstances. Since the trust between a user and an agent is so important, it is useful to examine the nature of trust in detail.

### 2.1. What is Trust?

Most generally, trust can be defined as "a generalized expectancy... that the word, promise, oral or written statement of another individual or group can be relied upon" (Rotter, 1980, p. 1). In the context of software agents, this means that the agent can be relied upon to do what it was instructed to do. 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" (Zand, 1972). Without the vulnerability, there is no need for the trust. In the context of software agents, it means no longer controlling the software directly, letting the process act on one's behalf, and accepting the risks that this may entail. Bickmore and Cassell (2001) go on to 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" (p. 397).

This concept of making oneself vulnerable in order 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 the oncoming driver 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 (2001) describe trust as a process of uncertainty reduction. By trusting others to act as we expect them to act, we can reduce the things we have to worry about.

Taking a computer science approach, Marsh (1994) has defined trust in terms of the behavior of the person doing the trusting. Thus, trust is "the behavior X exhibits if he believes that Y will behave in X's best interest and not harm X". In the context of agents, this means behaving in a way that is appropriate if the agent will always have your best interests in mind, and cause you no harm.

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.**

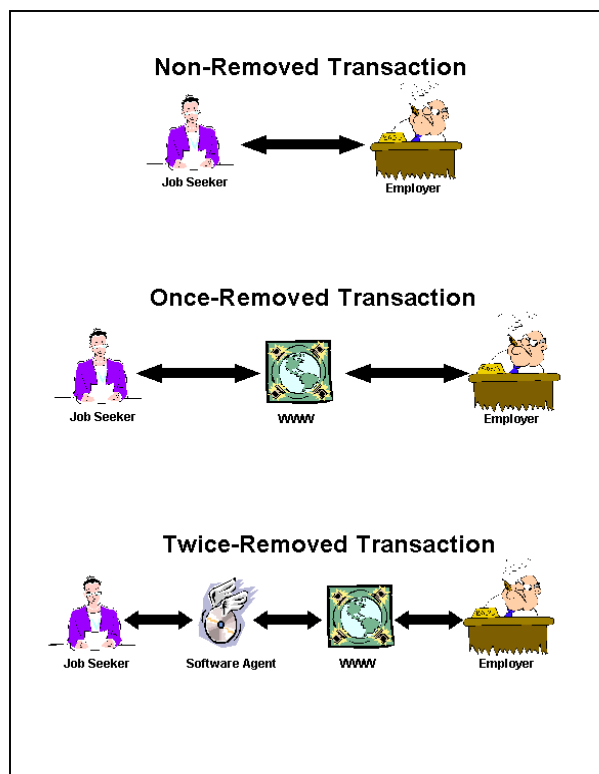
## 2.2. The Problem of Trusting Agents: Interactions Twice-Removed

Users may have difficulty trusting software agents because the user ends up working on a task that is twice-removed from the interface (See Figure 1). Consider the example of a user who is using a job-searching agent. A traditional, non-removed method of searching for a job would be to talk to employers directly, perhaps by visiting their offices. Here the job seeker is interacting directly with the potential employer to get information about the position (the top panel in Figure 1). A more modern method of searching for a job is to work in a computer-mediated fashion where the job seeker interacts with a computer program, perhaps a WWW browser, to view information that has been created by the employer (the middle panel in Figure 1). Thus, the interaction between the job seeker and the employer is once-removed. (Riegelsberger & Sasse, 2001, refer to this as a dis-embedded transaction.) 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, in turn, would search the Internet and gather information that has been provided by the employer. There is no direct connection between the user and the job-seeking activities (the bottom panel in Figure 1). Thus, the interaction between the job seeker and the potential employer is twice-removed (or dis-dis-embedded).

Research has shown that developing trust during once-removed interactions can be difficult, let alone trusting in twice-removed interactions. For example, Rocco (1998) showed that interpersonal trust is reduced markedly when communication is computer-mediated. Also, a numbers of studies, to be summarized below, have found that it can be quite difficult to develop trust during once-removed e-commerce interactions.

There are many valid reasons why users may be hesitant to trust software agents. Cheskin (1999) argued that disclosing personal information might involve more personal risk than financial interactions because personal assets like self-respect, desirability, reputation, and self-worth can be more valuable than money. Also, since agents operated autonomously outside of the user's vision and control, things may go wrong that the user does not know about, or cannot correct.

Youll (2001) has also described the issues involved in trusting agents. First, the user must make their instructions clear to the agent. This instructing phase could fail for a number of reasons: (1) the user does not clearly define the instructions, (2) the agent does



**Figure 1: Explanation of Twice-Removed Transactions**

not fully understand the instructions, or (3) the user and the agent interpret identical instructions differently.

Second, if the instructions have been understood, the user must be confident that the agent will execute its instructions properly, and only perform the tasks that the user intended. Third, the user must be confident that the agent will protect information that is private or sensitive. Finally, regarding the confidentiality of the information entrusted to the agent, the user must have confidence that the agent is not attacked or compromised in some way, 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, there are also valid reasons why users might make the choice to trust agents. Again Youll (2001) describes the advantages that agents can bring to a task. Due to the twice-removed nature of the interactions between the end-user and the task, agents are well suited for tasks that require high degrees of privacy. An agent can establish its own identity on the network, and protect the identity of the end-user. An example of how this can be done was seen in the Lucent Personalized Web Assistant (LPWA; Gabber, et al. 1999), which acted as a proxy for users who wanted

to navigate the WWW without revealing their true identities. Such services can even go so far as to establish new pseudonyms for each and every transaction, making it very difficult to establish a link back to the user.

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

### 3. BUILDING SUCCESSFUL AGENTS: A SUMMARY MODEL

Most of the research to date on privacy and trust has been focused on (once-removed) e-commerce interactions. However, the lessons are very relevant and extendable to agent interactions, and they provide a good starting point until more research is conducted on agent technologies. An important contribution to research on e-commerce trust is a path model of e-commerce customer loyalty proposed by Lee, Kim, & Moon (2000), as is shown in Figure 2. These authors describe how attitudes towards e-commerce will be determined by the amount of trust instilled in the user, and the amount of cost perceived by the user. Trust and cost combine together, in opposite directions, to determine the overall acceptance. In addition, Lee et al. identify a number of factors that contribute to trust, such as shared values and effective communication. They also identify factors that lead to perceived cost, such as the level of uncertainty.

An extended model of agent acceptance developed for this paper is shown in Figure 3. Here acceptance of the agent technology is determined by the combination of trust and perceived risk. The contributing factors identified by Lee et al. are included, along with factors identified by other researchers. This section reviews this model of agent acceptance in detail.

An important feature of Lee et al.'s e-commerce model, and the model of agent acceptance proposed here, is the separation of trust from perceived risk. The idea is that feelings of trust and risk can be established quite independently, and together they determine the final success of the agent technology. Trust contributes to the acceptance of the agent in a positive direction, while risk contributes in a negative direction. The effect is that the two factors interact with each other, so that agents instilling a low degree of trust may still be successful if there is also a low perceived risk. On the other hand, in very risky situations it may be that no amount of trust will offset the risk perceived by the

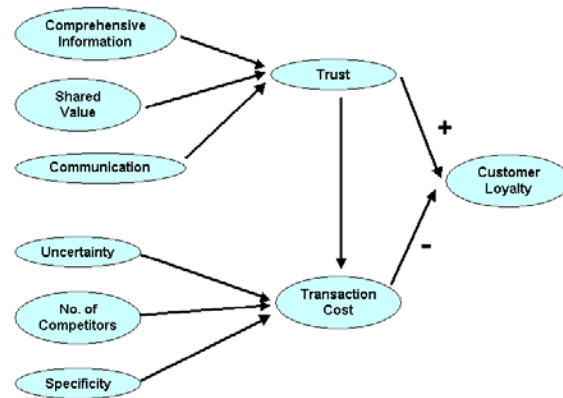


Figure 2: Lee, Kim, & Moon's Model of e-Commerce Loyalty

user, and the agent will never be accepted. Rotter (1980), in his review of the social psychology of interpersonal trust, supports this idea that trust and risk are separate concepts, and both contribute to the final behavior of an individual. Grandison and Sloman (2000) also describe trust and risk as opposing forces that combine during decision making about a service or an e-commerce transaction.

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

#### 3.1. Factors Contributing to Trust

As is shown in Figure 3, trust is a complex, multifaceted concept that is influenced by a number of factors (e.g., Grandison & Sloman, 2000). In this section a number of factors contributing to feelings of trust are described, and specific design recommendations are made for building trustworthy agents.

##### 3.1.1. Ability to Trust

The first factor that contributes to the trust a user may place in an agent service is their ability to trust. A number of researchers have proposed that people have a general ability to trust that forms a kind of baseline

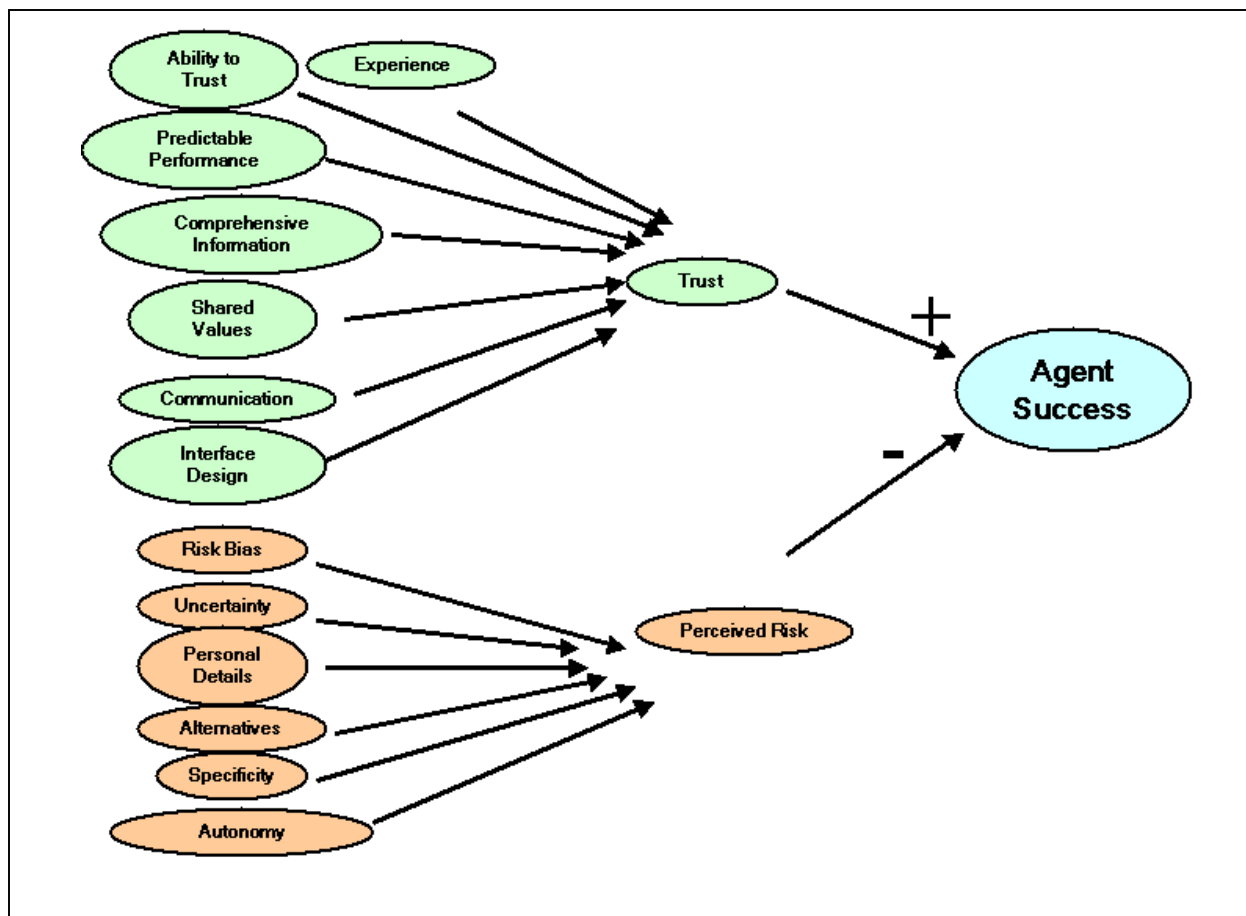


Figure 3: A model of agent success.

attitude when they approach any trust situation, and some people have a higher baseline level of trust than others. For example, Marsh (1994) describes "basic trust" as a person's general propensity to trust or not trust. This basic trust is part of their personality, and is one of the factors that contribute when making decisions about trust. Similarly, Rotter (1980) showed that there is a generalized trust that is "a relatively stable personality characteristic" (p. 1). Rotter also demonstrated that high and low trusters had markedly different opinions and behaviors (e.g., high trusters were less likely to cheat or lie, were seen as happier, and more attractive).

Directly related to the issue of trust on computer networks, Craner et al. (1999) surveyed Internet users about their attitudes towards privacy and trust. The survey respondents were then classified into groups that differed in their concerns about online privacy, following a scheme originally proposed by Westin (1991). The first group (27%) was only marginally concerned with online privacy and was quite willing to provide personal information when visiting WWW

sites. This group did have some concerns, such as the desire to remove themselves from marketing mailing lists, but they were generally quite trusting. The second group (17%) was at the opposite extreme, and was labeled "privacy fundamentalists". These users were extremely concerned about privacy and were generally unwilling to provide any information to WWW sites, even when privacy protection measures were in place. The third and largest group (56%) was labeled 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 WWW.

Thus, we have abundant evidence that people differ in their basic tendency to trust. Perri 6 (2001; yes his surname is the numeral 6) cautions, however, that basic trust can be misleading because people's perceptions are heavily modified by the context. He suggests the people's trust can change quickly depending on the context and their experience, and it is important not to

overemphasize the role of general personality characteristics.

When building agent systems that users will have to trust, developers should take into account the fact that users may differ in their general ability to trust. Some users may willingly trust an agent system with little reassurance of the privacy protection measures in place, while others may be very reluctant to give their trust. This means that interfaces must be flexible and be able to provide more information and reassurance for users that require it.

### 3.1.2. Experience

The second factor that contributes to trust is experience. It is clear that users can change their willingness to trust based on their experiences (Marsh, 1994; 6, 2001). If they have been harmed in some way, for example, they may be less willing to trust in the future. This change in trust may be specific to the situation or it may 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 (Grandison & Sloman, 2000). This means that trust can be "transitive", being passed from user to user.

Designers of agent systems should ensure that users are able to have positive experiences so they can develop trust. This means providing ample information on the operation of the agent (feedback). In addition, designers should support a sharing function so users can relate their experiences and trusting attitudes can be shared and spread (assuming the experiences are positive ones). This may mean collecting testimonials or anecdotes that can be shared with other users.

### 3.1.3. Predictable Performance

Another factor that can lead to agent trust is predictable performance. Systems and interfaces that perform reliably and consistently are more likely to be trusted by users. Bickford (1997) describes three important principles for predictable performance and its role in building trust:

1. consistency: The interface and system behave the same way each time they are used. For example, certain functions are always accessed in the same way, and always lead to the expected result.
2. aesthetic integrity: The interface has a consistent look and feel, throughout the entire

system. This includes the page design, buttons, text styles, etc.

3. perceived stability: The system should appear stable to the user. It should not crash. There should be no changes without users' knowledge, and users must be kept informed about any operational issues, such as upgrades or downtimes.

Another aspect of predictable performance is response time. Users prefer response times that are consistent and predictable, rather than variable and unpredictable (Shneiderman, 1997).

The resulting recommendation is that developers should ensure that the interface is consistent and predictable. This may mean adopting a style guide or interface guideline that is used in all parts of the system. Developers should also ensure that the system behaves consistently, and appears to be stable. Human factors evaluation techniques that may be useful for testing these aspects of a design are reviewed in Section 4.

### 3.1.4. Comprehensive Information

Another important factor in determining users' trust of a system is the amount of information provided. Systems that provide comprehensive information about their operation are more likely to be understood, and more trusted. Norman (2001) suggests that agent systems must provide an image of their operation so that users can develop a mental model of the way the system works. It is through this model that they will develop expectations and attitudes towards the system. Norman argues that users will develop mental models and assumptions about the system even when no information is provided, and these models may be wrong. To prevent this, developers should explicitly guide the model development by showing the operation of the system.

The importance of internal models of system operation was recently demonstrated by Whitten & Tygar (1999). This study tested users ability to use a PGP system to certify and encrypt e-mail. The results showed that the majority of the users were unable to use the system to perform the task. In fact, 25% of the users e-mailed the secret information without any protection. An analysis of the errors and an evaluation of the interface led these researchers to conclude that the major source of the problems was that users did not understand the public key model used in the PGP system. The PGP interface that was tested failed to provide the comprehensive information about how public key encryption works,

and the roles and uses of public and private keys. Without this information, users often developed their own ideas about how the system worked, with disastrous results.

Another example of a system that does not provide comprehensive information is the "cookies" module used in WWW browsers (Bickford, 1997). Cookies are small files that are assembled by WWW sites and stored on users' computers. Later, they can be retrieved by the WWW sites and used to identify repeat visitors, preferences, and usage patterns. The problem with cookies is that they can store a variety of information about users (including sensitive information) and yet their operation is invisible. Unless users explicitly change their browser options, they do not know when cookies are created or retrieved. In addition, most WWW browsers do not provide any way of viewing the cookies. They omit such simple functions as listing what cookies are stored on a system, and an ability to view the information stored within them. The P3P initiative (Reagle & Cranor, 1999) is an attempt to give users more control over cookies.

Developers of agent technologies must provide comprehensive information about how the system works. The role of the agent must be explained, and its operation must be obvious. This may mean allowing users to observe and track the actions performed by an agent, both in real-time and after the fact. In addition, effective interfaces should be developed for viewing and altering the information stored by agents.

### 3.1.5. Shared Values

Another factor that can lead to users trusting agents is the establishment of shared values between the user and the agent. That is, to the extent that the user feels that the agent values the things that they would, they will have more trust in the agent. In interpersonal relationships, these shared values are often built through informal interactions, such as the small talk that occurs in hallways or during coffee breaks. Bickmore and Cassell (2001) tested the role of small talk (informal social conversation) in building trustworthy agents. These researchers included small talk capabilities in a real estate purchasing agent called REA. REA was a life-sized conversational agent embodied as an animated figure on a large computer screen. REA was able to engage in small talk conversations designed to increase feelings of closeness and familiarity. For example, REA conversed about the weather, shared experiences, and her laboratory surroundings. In one experiment, a condition that included small talk interactions was compared with another condition that only involved

task-oriented interactions. The task in the experiment was to determine the users' housing needs, and this included gathering personal information about how much the user could afford to spend, and how large a house was required. When measures of trust and willingness to share personal information were examined, the results showed that the condition that involved informal social dialogues led to higher levels of trust among 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. For example, privacy policies can be clearly articulated so that users can compare their concerns with the policies in place (Cheskin, 1999).

### 3.1.6. Communication

Another factor that determines the amount of trust is the amount and effectiveness of communication between the agent and the user. Norman (1997) argues that continual feedback from the agent is important for success. This feedback should include having the agent repeat back its instructions so it is clear what the agent understood. Also, error messages should be constructed so that it is clear what was understood, and what needs to be clarified. In addition, through communication it should be made clear what the capabilities and limits of the agent are.

### 3.1.7. Interface Design

The final factor that can contribute to trust of an agent is the design of the interface itself. This means the look and feel of the software that is used to control the agent. This area includes such factors as appearance, functionality, and operation. Many of the generic attributes of good interface design also apply to designing agent interfaces. So, Norman's (1990) recommendations about "visible affordances" are relevant here, which means that whenever possible the function of an interface component should be clear from its visible appearance.

Cheskin (1999) completed an examination of interface designs that can communicate feelings of trust. They did this in the context of e-commerce WWW sites, but the lessons can also be applied to agent systems. In this study users were invited to comment about different e-commerce WWW sites that differed in interface design. The results were summarized in six fundamental interface characteristics that communicate trust (many of these points reinforce the factors described above):

1. brand: Trust can be influenced by the extent that users are already aware of the service provider, and their feelings about that provider. Providers that are already trusted in other contexts, such as real-world stores, may also be trusted in the new context.
2. navigation: Trust is influenced by the ease of finding things, which results from clear, logical presentation and consistent design.
3. fulfillment: Trust is enhanced if the process for getting a task done is clear and traceable.
4. presentation: Feelings of trust can be increased if material is presented clearly, if the layout is clean and functional, and the presentation is professional. There are reasons why banks appear as they do -- to instill trust. Broken windows and peeling paint do not instill trust. The appearance should be professional and official looking, like money or certificates.
5. technology: Trust can be built if the site appears to work smoothly and quickly.
6. logos of assurance: Including icons and text that represent seals of approval or assurances of safety can increase feelings of trust.
1. status indicators: Allowing the user to see the status of the actions can increase confidence and trust.
2. displaying data already entered: Displaying the data to be used by the system before it is launched can increase trust.
3. continuous visibility: Making the operation of the system transparent is important.
4. tracking: Allowing tracking of the activities can build trust.
5. recourse: Providing a mechanism to recall or undo an action can lead to more confidence and trust in an interface.
6. trial runs: Supporting trial runs or demonstrations may be a good technique to reassure users that the system is operating as they intended.
7. fast response times: providing fast, consistent response times can lead to positive feelings about the system being used.

In another empirical study, Kim & Moon (1998) examined a number of interface design factors and their effects on levels of trust in an e-banking service. The result was a series of recommendations on the visual components of the interface:

1. use "clip art" graphics, and large, 3-dimensional images
2. use cool colours
3. use pastel shades
4. use low brightness
5. use colors symmetrically

Riegelsberger & Sasse (2001) also examined the role of various interface characteristics in building trust. In this study a mockup e-commerce interface was developed to include various interface components. Potential users then "walked through" the interfaces and provided comments (see Section 4 for a description of assessment methods). The result was a list of interface components that could build trust, and these again reinforce some of the factors reviewed above:

A controversial issue in designing trustworthy interfaces is the value of anthropomorphism. Does creating a human-like interface, perhaps with an animated character and a conversational interface, lead to more feelings of trust? Bickmore and Cassell (2001) argue that an animated character that can engage in small talk conversations can lead to shared values and higher trust in some users. Similarly, Laural (1997) argues that we have years of experience interacting with other people, and these skills can be transferred to interactions with computers. Moreover, agent systems are human-like because of their ability to perform autonomous actions, so an anthropomorphic interface is appropriate. However, others (Norman, 1997; Riegelsberger & Sasse, 2001; Erickson, 1997) have argued that such anthropomorphism can lead to disappointment if the interface does not live up to expectations. If the agent cannot really behave like a human, then having a human-like interface may actually diminish trust rather than build it. Erickson (1997) has relayed some anecdotes where users question the motivation of human-like "guides", and sometimes became quite angry if the character does not behave as expected. 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.

### 3.2. Factors Contributing to Perceived Risk

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

#### 3.2.1. Risk Perception Bias

Similar to basic trust, users may 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 has been described by Perri 6 (2001) as 4 basic approaches to risk analysis:

1. fatalism: users feel that they have no control, and risk decisions are out of their hands
2. hierarchy: users feel that risks should be contained by controls and regulation
3. individualism: users feel that risks should be taken when appropriate for the individual
4. enclave: users feel that risks are systemic and should be handled with pressure, dissent, and market systems

Agent system designers should consider these basic approaches to risk assessment. It may be useful to design system features that address each of these areas. For example, an agent system may contain information to explain how users can have control over the risks they are taking. Also, a system can include information about the controls being put in place and the regulations that are being followed. Finally, allowing users to share information and experiences, and communicate with the system developers, may lead to feelings of empowerment and fewer concerns about risk.

#### 3.2.2. Uncertainty

Another method to reduce risk perception is to reduce uncertainty. The more users know about a system and how it operates, the less they worry about taking risks (assuming all that they learn is positive). This is highly related to the "comprehensive information" and "communication" factors for building trust.

#### 3.2.3. Personal Details

An obvious factor in risk perception is the amount of sensitive information being provided. If more personal details are being provided to the agent, perceptions of risk are likely to increase. System developers should only ask for information that is necessary to do the job, and avoid where possible information that may be especially sensitive. Exactly what information the users consider sensitive may require some investigation. For example, Cranor, Reagle, & Ackerman (1999) found that phone numbers were more sensitive than e-mail addresses because unwanted phone calls were more intrusive than unwanted e-mail messages.

#### 3.2.4. 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 method to search for a job is to use a new agent technology, users may feel they are taking more risks than situations where there are multiple methods (i.e., non-agent WWW interfaces, phone calls, employer visits).

#### 3.2.5. Specificity

Similarly, if there is a sole supplier of a service, users may feel they are at more risk from exploitation than situations where there are multiple suppliers. In the job-searching example, it means that users may be more comfortable if there are multiple job searching agents to choose from.

#### 3.2.6. 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. As discussed previously, agents can range from low risk advice-giving systems to higher risk, independent acting agents. Lieberman (2002) advocates developing advice agents and avoiding, for now, agents that truly act on their own. Advice systems have the advantage that they 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 what advice their users accept. In the job-searching example, it may be most appropriate for the agent to suggest possible jobs that the user should apply for, rather than completing the application autonomously.

#### 4. CHECKING YOUR WORK: HUMAN-FACTORS EVALUATION TECHNIQUES

Most of the standard human factors evaluation techniques are appropriate when developing agent technologies. It is beyond the scope of the current paper to provide an exhaustive review, but this section does present a brief review and some notes about the particular applicability for agent design. (Interested readers should consult one of the many books available on usability testing, such as Nielson, 1993; Mayhew, 1999; or Shneiderman, 1997.)

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 may be one-on-one interviews or focus group sessions. For example, researchers might conduct a needs analysis to determine the tasks that should be performed by the agent, and how it should be accomplished. Users may also be questioned about their preferences and concerns, and this may be particularly important for discovering concerns about privacy and sensitive information.

Another technique that will be valuable during the early design stages is heuristic evaluation. Here researchers with expert knowledge examine a prototype system against a set of criteria. These criteria may come from general background knowledge about human factors, or specific recommendations such as those presented earlier in Section 3. 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 may 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 early in the design process, 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, and specific performance measures are recorded under controlled conditions. For example, researchers may record the number and type of errors, or the time needed to complete a task. Empirical tests can be expensive and time-consuming to conduct, so they are often reserved for the final stages of product development.

#### 5. CONCLUSIONS

Intelligent, autonomous agents have the potential to facilitate complex, distributed tasks and protect users' privacy. However, building agents users will trust with personal and sensitive information is a difficult design challenge. Agent designers must pay attention to human factors issues that are known to facilitate feelings of trust. These include providing transparency of function, details of operation, feedback, and predictability. They must also consider factors that lead to feelings of risk taking. This means reducing uncertainty, collecting the minimal amount of information, and carefully considering the amount of autonomy an agent will have.

#### REFERENCES

- 6, P. (2001). Can we be persuaded to become PET-lovers? Paper presented at the *OECD Forum Session on Privacy Enhancing Technologies*. Paris, Oct. 8.
- Bickford, P. (1997). *Human interface online: A question of trust*. [http://developer.iplanet.com/viewsource/bickford\\_trust.html](http://developer.iplanet.com/viewsource/bickford_trust.html)
- Bickmore, T., & Cassell, J. (2001). Relational agents: A model and implementation of building user trust. *Proceedings of SIGCHI '01*, March 31-April 4, Seattle, WA, USA. pp. 396-403
- Bradshaw, J.M. (1997). An introduction to software agents. In J.M. Bradshaw (Ed.), *Software agents*. Menlo Park, CA: AAI Press/MIT Press.
- Cheskin Research & Studio Archetype/Sapient (1999). *eCommerce Trust Study*. <http://www.cheskin.com/think/studies/ecomtrust.html>
- Cranor, L.F., Reagle, J., & Ackerman, M.S. (1999). *Beyond concern: Understanding net users' attitudes about online privacy*. AT&T Labs-Research Technical Report TR 99.4.3. <http://www.research.att.com/library/trs/TRs/99/99.4/>
- Erickson, T. (1997). Designing agents as if people mattered. In J.M. Bradshaw (Ed.), *Software agents*. Menlo Park, CA: AAI Press/MIT Press.
- Gabber, E., Gibbons, P., Matias, Y., & Mayer, A. (1997) How to make personalized web browsing simple, secure, and anonymous. *Proceedings of Financial Cryptography 97*, February, 1997, Springer-Verlag, LNCS 1318. <http://www.bell-labs.com/project/lpwa/papers.html>
- Grandison, T., & Sloman, M. (2000). A survey of trust in Internet applications. *IEEE Communications Surveys, Fourth Quarter 2000*.

- <http://www.comsoc.org/livepubs/surveys/public/2000/dec/grandison.html>
- Kim, J., & Moon, J.Y. (1998). Designing towards emotional usability in customer interfaces -- trustworthiness of cyber-banking system interfaces. *Interacting with Computers*, 10, 1-29.
- Laurel, B. (1997). Interface agents: Metaphors with character. In J.M. Bradshaw (Ed.), *Software agents*. Menlo Park, CA: AAAI Press/MIT Press.
- Lee, J., Kim, J., & Moon, J.Y. (2000). What makes Internet users visit cyber stores again? Key design factors for customer loyalty. *Proceedings of CHI '2000, The Hague, Amsterdam*. pp. 305-312.
- Lieberman, H. (2002). Interfaces that give and take advice. In J.M. Carroll (Ed.), *Human-Computer Interaction in the New Millennium*. N.Y.: ACM Press, 2002.
- Marsh, S. (1994). *Formalising trust as a computational concept*. PhD Thesis, University of Stirling, Scotland.  
<http://www.iit.nrc.ca/~steve/Publications.html>
- Mayhew, D.J. (1999). *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design*. Morgan Kaufmann.
- Negroponte, N. (1997). Agents: From direct manipulation to delegation. In J.M. Bradshaw (Ed.), *Software agents*. Menlo Park, CA: AAAI Press/MIT Press.
- Nielsen, J. (1993). *Usability Engineering*. Boston, MA: Academic Press.
- Norman, D.A. (1990). *The Design of Everyday Things*. Currency/Doubleday.
- Norman, D.A. (1997). How might people interact with agents. In J. Bradshaw (Ed.), (1997). *Software agents*. Menlo Park, CA and Cambridge, MA: AAAI Press/The MIT Press.  
<http://www.jnd.org/dn.mss/agents.html>
- Norman, D.A. (2001). *How might humans interact with robots? Human robot interaction and the laws of robotology*.  
[http://www.jnd.org/dn.mss/Humans\\_and\\_Robots.html](http://www.jnd.org/dn.mss/Humans_and_Robots.html)
- Reagle, J., & Cranor, L.F. (1999). The platform for privacy preferences. *Communications of the ACM*, 42, 48-55.
- Riegelsberger, R. & Sasse, M.A. (2001). Trustbuilders and trustbusters: The role of trust cues in interfaces to e-commerce applications. Presented at the *1st IFIP Conference on e-commerce, e-business, e-government (i3e)*, Zurich, Oct 3-5 2001.  
[http://www.cs.ucl.ac.uk/staff/jriegels/trustbuilders\\_and\\_trustbusters.htm](http://www.cs.ucl.ac.uk/staff/jriegels/trustbuilders_and_trustbusters.htm)
- Rocco, E. (1998). Trust breaks down in electronic contexts but can be repaired by some initial face-to-face contact. *Proceedings of CHI 98, Los Angeles, USA*. pp. 496-502.
- Rotter, J.B. (1980). Interpersonal trust, trustworthiness, and gullibility. *American Psychologist*, 35(1), 1-7.
- Shneiderman, B. (1997). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley.
- Weisband, S.P., & Reinig, B.A. (1995). Managing user perceptions of email privacy. *Communications of the ACM*, 38(12), 40-47.
- Westin, A.F. (1991). (with Louis Harris & Associates). *Harris-Equifax Consumer Privacy Survey*. Atlanta, GA: Equifax, Inc.
- Whitten, A. & Tygar, J.D. (1999). Why Johnny can't encrypt: A usability evaluation of PGP 5.0. *Proceedings of the 9th USENIX Security Symposium*, August 1999.  
<http://www.cs.cmu.edu/~alma/johnny.pdf>
- Youll, J. (2001). Agent-Based Electronic Commerce: Opportunities and Challenges. Position statement for panel discussion on Agent-Based Electronic Commerce: Opportunities and Challenges, in the *5th International Symposium on Autonomous Decentralized System with an Emphasis on Electronic Commerce*, March 26-28, 2001, Dallas, Texas, USA  
<http://www.media.mit.edu/~jim/projects/atomic/publications/youll-mit-isads.pdf>
- Zand, D.E. (1972). Trust and managerial problem solving. *Administrative Science Quarterly*, 17, 229-239.