

Trusted Interaction: User Control and System Responsibilities in Interaction Design for Information Systems

Larry L. Constantine, IDSA

University of Madeira, Laboratory for Usage-centered Software Engineering
Constantine & Lockwood Ltd, 58 Kathleen Circle, Rowley, MA 01969, USA
lconstantine@foruse.com

Abstract. Trust emerges from interaction. If trust in information systems is to be promoted, then attention must be directed, at least in part, to interaction design. This presentation will explore issues of trust in the interactions between users and systems from the perspective of interaction design. It will consider a variety of pragmatic aspects in interaction design that impact user trust, including, predictability, interface stability, user control, and the match between expectations and performance. It will critically examine contemporary design practices, such as adaptive interfaces, in terms of their impact on user trust.

1 Introduction

Trust has long been recognized as a crucial element shaping human relationships and human interaction of all kinds. Where there is trust, activities proceed more smoothly, actions are more decisive, and people work with greater confidence. By reducing uncertainty under conditions of interdependence, trust engenders more efficient collaboration [1]. The absence of trust, in contrast, introduces inefficiencies, demanding added vigilance, encouraging protective and unproductive actions, and complicating interaction.

Trust, then, is a potentially important factor in human performance, not only in interpersonal relations but in our relationships with the technology we use in our everyday personal and work lives. When we do not trust the system we are using, we proceed more cautiously and hesitantly. We take additional time, often unnecessary, to think through and plan actions before proceeding, then double-check our actions. We protect ourselves from untrustworthy software with redundant copies of data or take circuitous and inefficient but seemingly safe paths to achieve our goals. Unable to count on “good behavior” from the software, we are reluctant to explore new features that might well prove useful to us.

It is not the purpose of this paper to offer yet another survey of the extensive literature on trust in human relations nor even to review that subset concerned with trust in commercial transactions. Rather the intent is to explore the interconnected issues of user trust and user performance from the perspective of a designer, to gain insights from the broader field of human trust to draw implications regarding how we can design the interface between system and user to promote and sustain trust on the part

of users, and thereby enhance user performance. With this objective in mind, it is appropriate to begin with a focus on the problems of user interface design.

2 Designing Interaction

Although there is wide variability in how applicable terms are defined and employed, the design of user interfaces ultimately encompasses two broad areas, which may conveniently be referred to as presentation design and interaction design. Presentation design covers what is presented to the user, where it is presented, and how it is presented. As such, it involves the designer in decisions about what information and capabilities--the tools and materials available for use--are to be present, how these will be arranged or organized, and how they will look, feel, or sound to the user. For the graphical user interfaces of modern information systems, this often largely reduces to visual design, although other modalities of communication, such as sound, may also play some role. To an industrial designer devising products for consumer, business, or industrial use, many other factors may come into play, such as, weight and shape of devices and the tactile response of manual controls.

In recent years, professional interaction designers have laid claim to virtually the entire territory of external design, but, strictly speaking, interaction design is concerned with the design or planning of the interaction between user and system by way of the user interface. Thus interaction design covers not only how the user will interact with the system but how the system will respond to the user, including how the behavior of the various elements or pieces of the user interface are coupled or interdependent.

2.1 Designing for User Performance

User performance refers to the ability of users of a system to satisfy their intentions and achieve their objectives efficiently and reliably. One would expect that user performance would be at the heart of user interface design, and that was certainly the case with early work in ergonomics and human factors. However, throughout recent decades, a different perspective has dominated the design world, particularly for the Web and in products aimed at the general consumer population. Under the broad rubric of user-centered [2] or human-centered design, this perspective makes users and user satisfaction the central focus of design activities. Its purest and most extreme manifestations can be found in participatory design, which actively involves users as collaborators in the design process, and in user experience design, which broadens the focus of design to encompass every aspect of the user's experience with a system.

User-centered design elevates user satisfaction and user experience to primary importance. Potentially every aspect of users as human participants in use is of interest, including personality, attitudes, feelings, social context, cultural background, personal and work experience, and the like. Although user-centered design methods and techniques vary widely, the essence of all user-centered approaches is iterative design refinement grounded in a rich understanding of users gained through substantial initial user studies and driven through successive rounds of redesign by extensive feedback from user evaluations and user testing.

Growing dissatisfaction with the end results of this process has led to recent calls for a shift in focus from users to usage, from actors to activities [3]. Even the originator of the term “user-centered design,” Donald Norman, has suggested in a highly controversial critique [4] that the current interpretation of user-centered design is flawed and even potentially harmful. He has called for an activity-centered design philosophy that looks more closely at what people are doing and are trying to do.

Activity-centered design in the sense used by Norman and usage-centered design [5] both seek to enhance user performance by devising user interfaces better fitted to the activities, tasks, and operations carried out by users. Users are primarily of interest for the roles they play in relationship to the system being design and for the parts they play in activities. The goal of the designer is to fully understand the tasks of users within these activities and to find the most effective way to support these tasks--broadly, in combination, and in detail.

From this point of view, trust in itself is of little interest except insofar as it improves performance in interaction with the system being designed. While this orientation may seem cold hearted to those steeped in the humanistic traditions of user-centered design, in truth it gets to the heart of why people use systems at all, namely, to accomplish something.

3 Elements of Human Trust

In human social contexts, trust is recognized as complex and variable, but it is, at least in part, an expression of individual propensity. Some of us are more prone to trust from the outset than are others. Trust also clearly depends on the specific interpersonal relationship. We trust some people more than we do others. It can also depend on the situation. You may trust a close colleague with sensitive commercial information but not with your personal health history. Finally, it may depend on particular kinds of social contexts. We are required by the setting to put our trust into certain people at certain times, such as when being treated by medical personnel in an emergency facility. Trust, then, is an attribute of the individual, the relationship, the situation, and the context.

Trust can be viewed from many different perspectives, and many disciplines in the human sciences have weighed in on the matter, including psychology, sociology, and economics. Some recent attempts have tried to distill the essence of this vast literature for various purposes [6, 7].

With an eye to application in electronic commerce, McKnight and Chervany [6] combined the sundry perspectives of diverse fields into an interdisciplinary model of trust concepts that includes dispositional trust, institutional trust (referring to both situations and social structures), and interpersonal trust. They developed a typology of trust based on analysis of some 65 definitions of trust from varied disciplines. These they ultimately distilled down to 16 characteristics organized into four conceptual categories--competence, predictability, benevolence, and integrity--plus a miscellaneous category of otherwise unclassified characteristics, of which the most salient is openness.

Also addressing the domain of electronic commerce on the Web, MIT trust guru Glen Urban advocates for trustworthy Web sites based on eight imperatives [8]:

transparency, quality of products and services, alignment with customers, helping customers help themselves, putting customers to work, comparing products to competitors, trust-based supply chain, and trust as top priority. While some of these clearly relate to business policy and practice rather than interaction design, others might be conceptually linked to interaction design.

Although commerce-oriented Web sites are certainly a special case of information systems, there is considerable overlap with work in other areas of application, such as the Tschannen-Moran and Hoy [8] catalog of facets of trust validated through factor analysis. These facets include: willingness to risk vulnerability, confidence, benevolence, reliability (consistency or predictability), competence, honesty, and openness.

4 Trust in User Interaction

Among the many facets of trust in human social situations, not all are equally salient for human-computer interaction or as likely to yield rich insights for interaction design. Some, like predictability, can be applied directly. Others, such as benevolence or integrity, require a certain amount of creative interpretation. Still others, such as, comparing products to competitors or trust-based supply chain, are limited to very specific contexts. And some, such as, alignment with customers or trust as top priority, are clearly beyond the ken of interaction design.

By reorganizing the elements of the theoretical models and focusing on those of clearest salience and most closely coupled to interaction design, four factors (Table 1) emerge as important in design for trusted interaction: predictability, transparency, competence, and benevolence.

Table 1. Salient factors in interaction design for trusted interaction

Factor	Characteristics
predictability	consistency, reliability, dependability
transparency	openness, visibility, accessibility, directness, clarity
competence	capability, completeness, reliability, accuracy, performance
benevolence	responsiveness, responsibility, safety

4.1 Predictability

“At its most basic level trust has to do with predictability, that is, consistency of behavior and knowing what to expect from others,” [7]. A close reading of numerous sources suggests that predictability is arguably the most important single factor in promoting trust in interaction. We are apt to trust a system that does what we expect and what we ask, that is reliable and behaves as we anticipate, that does not suddenly do something different when we take a particular action. The behavior most damaging to trust is when a system does something unpredictable and unexplainable, particularly if it is unwanted or unasked.

Predictability is often conflated with consistency or even repetitiveness, but predictability is more than mere consistency. Even completely novel features and unprecedented functions can promote trust in the system and encourage exploration if they behave as the user predicts or anticipates [9].

4.2 Transparency

Transparency is the user interface parallel to honesty in human relationships. An interface is transparent when its content and organization are evident to the user, when features and information are visible or readily available where and when needed. Transparency means directness of communication, without obfuscation, misdirection, or disguise. It means that representations are direct, that What You See Is What You Get (WYSIWYG) [5].

Openness is the extent to which relevant information is not withheld [10, 11]. Does an application expose information to the user or keep it hidden? Is important information easily accessed, particularly information needed by users to make decisions and exercise their options? Is the underlying model or structure of the information evident? Does the user interface attempt to hide what is happening or what it does or does it expose these to the user?

Transparency is very closely related to the widely accepted design principle of visibility [5, 12] which admonishes designers to make clear to users what options are available and what actions are possible as what are the results of their actions.

4.3 Competence

Trust is affected by whether the other party possesses the competence or relevant capability to carry out their responsibilities in a relationship. In the context of information systems, competence means that the system is capable of doing what the user needs it to do, that it performs reliably and delivers accurate results. To be considered competent by the user, the system must be complete in the functionality that the user reasonably expects, without flaws or holes in functionality that would render it ineffective. It must perform with sufficient speed and dispatch to meet user needs with respect to timeliness.

4.4 Benevolence

In human relationships, trust depends to some degree on the perception of benevolence, that the other has our best interests at heart or at least is not malevolent. While the application of this concept to interaction design may seem to be a stretch, some aspects of benevolence carry over into human-computer interaction. Benevolence is dependent on respect, on respect for possessions, for boundaries, and for privacy. It has been described as “the confidence that one’s well-being, or something one cares about, will be protected and not harmed” by the other [7]. Benevolence in this sense is obviously connected to the sense of safety and security, the assurance that the system will not bring harm to the user by discarding, corrupting, or destroying the user’s data. Thus the manner in which a system handles user input and possessions, such as data files, is clearly an important contributor to perceived benevolence. A trustworthy

system acts responsibly in handling user information. It does not require re-entry of already provided data. It does not unduly penalize the user for mistakes or make recovery from error difficult.

Benevolence also includes responsiveness (McKnight and Chervany). A system that is responsive to user needs, as expressed in user actions and choices, is, in an important sense, benevolent. Benevolence, like predictability, is linked to user control, the perception that the system will respond as directed and desired, taking responsibility for doing what the user asks when asked.

5 The Process of Trust

How is trust engendered or eroded? How do we, as users, come to trust or distrust the technology we use? For that matter, how do we, as colleagues, come to trust—or distrust—each other? Trust in tools or artifacts is not the same as trust in persons or institutions, but it does, to some large degree, emerge from the same foundations.

A precondition of trust is interdependence, where each party to an interaction depends on the other in some manner [13]. This is clearly the case in human-computer interaction: the system obviously depends on the user for input and direction, while the user depends on the system to perform as requested and provide feedback and the desired results.

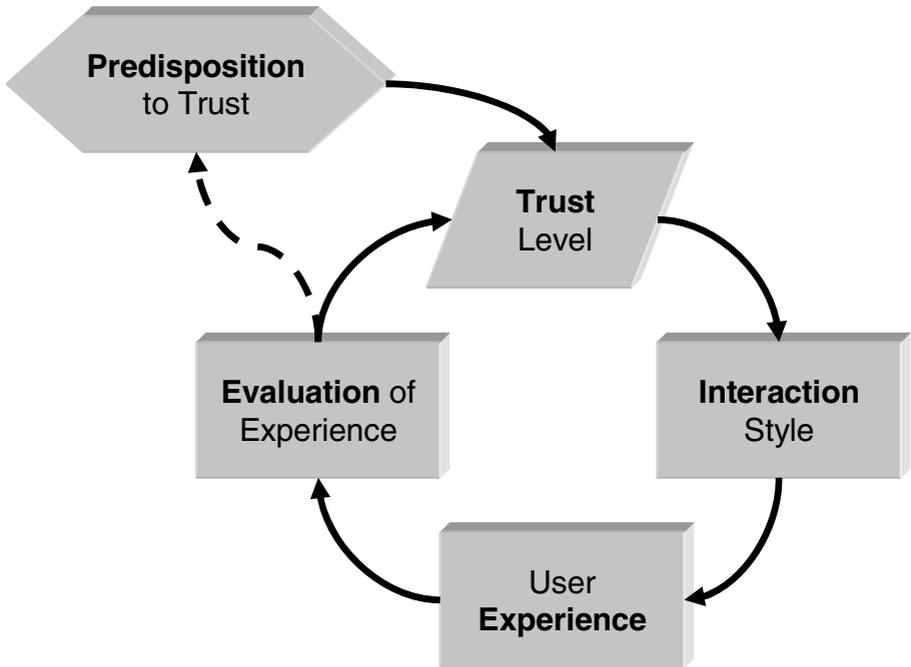


Fig. 1. Feedback model of trust in user-system interaction

Apart from any pre-existing propensity, trust emerges from experience. The evolution of trust in human-computer interaction can be expressed in a simple feedback model with five elements: predisposition, trust level, interaction style, user experience, and evaluation. As represented in Figure 1, the level of trust at any given time is a function of the evaluation of experience in interaction with the system as influenced by the prior level of trust.

Users bring to their first encounter with any system a predisposition to trust that is compounded of many factors, including personality, habitual orientation, and prior experience with other software. This predisposition to trust determines the initial level of trust and shapes the style of early interaction. Level of trust in turn influences interaction experience, as it affects such things as the speed of interaction, propensity to explore new or alternative paths, and use of self-protective mechanisms, all of which can impact system behavior and, consequently, user experience. High levels of initial distrust may slow or even preclude the development of trust.

It is unlikely that most users consciously or deliberately evaluate their experience and thereby alter their level of trust, yet it seems evident that users are typically aware of their level of trust regarding specific software or software in general. (“I find Excel very unpredictable. I can never trust it to do what I expect.” “In general, I don’t trust new releases.” “I think that [XYZ] is a firewall I can trust.”)

One manifestation of distrust is the emergence of defensive interaction involving self-protective measures. An example of defensive interaction is compulsively saving a file after every few paragraphs to avoid losing work or always spelling out the full name of operations to avoid the misinterpretation of a mistyped abbreviation.

The model also closes the loop from interaction in the particular back to the predisposition to trust. Users who are burned by badly behaved software often carry that experience over into future interactions with other software. Defensive interaction can become habitual. For example, when installing new software, many users will not allow the installation process to automatically restart the system but will opt for manual restart at a later time because in the past they have seen installations hang during automatic restart.

6 Designing for Trusted Interaction

A few attempts have been made to tie user interface design and usability to user trust, primarily in e-commerce (for example, [14, 15, 16]). Systems that are easier to use are, not surprisingly, more trusted. Initial trust in Web sites, for example, has been found to be promoted by logical structure, simple and clear classification, ease of navigation, consistency, and the like [16]. Such conclusions are somewhat akin to asserting that well-behaved and well-dressed people are likely to be perceived as trustworthy.

A more interesting question is how specific design practices and interaction techniques influence facets of user trust. In this section, I will consider a number of contemporary user interface design practices in terms of their impact on user trust and user performance.

6.1 Adaptive and Adaptable Interfaces

Stability of the user interface is a critical contributor to predictability. When the same elements are always located in the same place and do the same thing, the interface is *prima facie* more predictable, contributing to user trust and thus enhancing user performance. This is not just a matter of consistency in the traditional sense of similarity of appearance and consistent placement of objects within the user interface; it is also a matter of interaction design.

Adaptive user interfaces are a popular modern style of interaction design. Adaptive interfaces change form and behavior automatically in response to user actions, in principle to adapt dynamically to ever changing tasks and context of use. Adaptable interfaces, by way of contrast, support end-user tailoring of the user interface. Where adaptive user interfaces attempt to anticipate user needs and configure the user interface for the user, adaptable user interfaces leave changes in the configuration under the control of the user.

The popular application suite Microsoft Office provides examples of both kinds of designs. In addition to being able to alter dozens of predetermined options, such as whether or not to check spelling and grammar continuously, users can also enter a special mode that allows them to customize the configuration of menus and toolbars, even to define new toolbars. This is an example of an adaptable interface.

The adaptive menu feature of Microsoft Office is an example, as the name suggests, of adaptive interface design. On initial installation, the default behavior of Office menus is to display short, incomplete lists of commands when a menu is opened, as seen in Figure 2(a). After a short delay or when the user clicks at the bottom of the menu, the complete list of available commands appears, as in Figure 2(b). Which commands display when a menu is first opened depends on what selections the user has made previously. Selecting a command not initially displayed will promote it to the initial list. Commands that are used infrequently automatically disappear from the short list. Thus, not only does each menu change between two distinct forms, short and long, with different arrangements of commands, but the contents and arrangement of the short form seen first can vary from use to use, as seen in Figure 2(c).

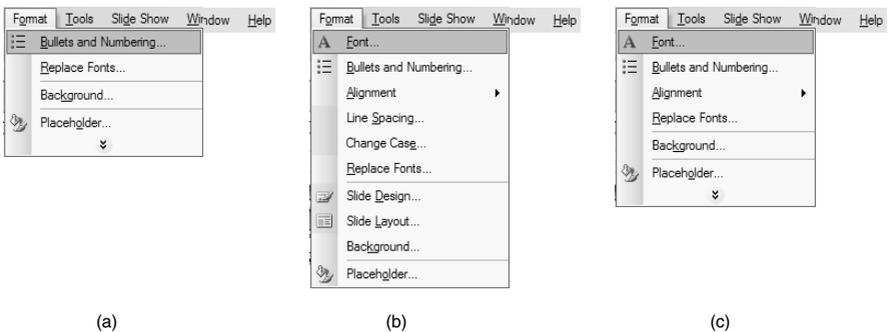


Fig. 2. Adaptive menus in Microsoft Office. Initial short list (a), full list (b), and short list after some use (c).

The design rationale for this approach is that it is supposed to simplify use by adjusting the menu content to the user's actual pattern of usage. The impact on user trust and performance is quite different, however. Adaptive interfaces not only reduce stability and predictability, but they usurp user control. Users describe their experience with adaptive menus in terms of what to them is utter unpredictability. ("I never know what I am going to see when I open a menu. Something that was one place the last time I used it is now suddenly someplace else or hidden where I have to wait for it.")

Performance is degraded because users must actually read down the menu list on every use in order to select the right command. Moreover, they never gain the efficiencies of learned reflexes, such as clicking halfway down the Tools menu to track changes. The constantly changing content and organization also reduces the reliability of interaction by significantly increasing the likelihood of making the wrong selection.

In informal audience surveys of thousands of students, professionals, and business people over recent years, only a handful of regular Office users report finding the feature to be useful. The vast majority have turned it off, and not a few have reported trying unsuccessfully to stop the behavior. (The feature was not originally indexed as such within the help system, and the is not where most users expect it within the Tools | Options... dialog but under Tools | Customize... on the Options tab.)

6.2 Wizards, Agents, and Pseudo-intelligence

Leaving aside the issue of predictability and interface stability, to be effective, adaptive interfaces must correctly anticipate user needs. Often this requires some manner of software intelligence, usually in the form of rules-based or heuristic inference. In principle, a sufficiently intelligent software agent might successfully analyze user actions and anticipate needs. In practice, the limited pseudo-intelligence provided by the software is rarely if ever helpful. The problem is, of course, that not even human beings are very good at anticipating the real needs of others, even in relatively simple situations. For that matter, in the real physical world and in real human relationships, accurate anticipation is not always experienced as a good thing. ("I'd rather do it myself.") People can find it disconcerting to have their every need or intention anticipated through behind-the-scenes manipulation [17].

In the software world, Microsoft Office once again provides a convenient example in the form of the so-called Office Assistant, known to most users as "Clippy" (or "that (*)#@^ animated paper clip") but actually named Clippit. Almost from the beginning, the intrusively obsequious dancing paper clip has been a controversial user interface feature. My own informal audience surveys over the decade since its introduction in Office '97 suggest that the Office Assistant is widely reviled, almost universally regarded as useless or at least largely ineffective, and frequently the subject of violent fantasies, particularly among those who were unable to get rid of it. In operation, the Office Assistant tries to guess what the user is doing and then offer appropriate help. However, it almost invariably guesses wrong and the proffered help is rarely if ever useful. Over time, most users come to see most such agents as arrogant and decidedly not benevolent.

Even where users are highly dependent on pseudo-intelligent agents, such as in automatic email spam filters, there remains a high level of distrust, particularly as

false positives, in which legitimate and potentially important messages get classified as spam, require manual review and recovery by the user.

7 Paternalism and User Control

To some extent, many modern user interface design practices are experienced by users as arrogant, manifesting an attitude of “we know better than you.” Modern information systems often hoard information and hide it from the user. They bury controls under layers of wizards and dialogs, menus and property sheets, all in the name of protecting users from themselves.

Truly benevolent design is not paternalistic. It is built on trust and grounded in respect, respect for the integrity and the ability of the user. It manifests itself in not only the architecture of interaction but also small details of design and programming practices, such as never discarding user input. Nothing erodes trust more quickly than for users to click on the back button within a browser only to discover a form newly cleared of all their personal data so carefully entered. To treat with cavalier disrespect what the user has worked hard to create not only violates trust but obviously contributes to inefficiency, inaccurate or incomplete data, and even lost customers.

Trust is reciprocal. If users are to trust systems, system designers need to learn how to trust users, to return control to users and allow them to make their own decisions about what to do, where to do it, and how. This is not a call for a return to the intimidating and confusing *tabula rasa* of the command-line interface of yesteryear. Rather it is a suggestion that designers and developers may have become too arrogant in assuming that users are stupid and incapable, and that “we” know best.

In truth, all too often users have been made stupid and incapable by the ill-conceived tools we have given them. Rather than “dumbing down” interfaces with so-called wizards that work no magic or arrogating user prerogatives by embedding in software more so-called intelligence that is at best profoundly naïve and at worst maliciously misguided, we should instead be designing better tools. Tools do not do the work of the user. Good tools serve the user—and thereby earn the user’s trust—by allowing and enabling activity, use, performance. To this end, to design trustworthy interaction, we must better understand and apply knowledge of how tools are used in the activities of users.

References

1. Arrow, K. J.: *The Limits of Organization*. Norton, New York (1974)
2. Norman, D., and Draper, S.: *User Centered System Design: New Perspectives on Human-Computer Interaction*. Erlbaum (1986)
3. Constantine, L. L.: *Beyond User-Centered Design and User Experience*. *Cutter IT Journal*, Vol. 17 No. 2, (2004)
4. Norman D.: *Human-Centered Design Considered Harmful*. *Interactions*, Vol. 12, No. 4, (2005) 14-19
5. Constantine, L. L., and Lockwood, L. A. D.: *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. Addison-Wesley Reading, MA (1999)

6. McKnight, D. H., Chervany, N. L.: What Trust Means in E-Commerce Customer Relationships: An Interdisciplinary Conceptual Typology. *International Journal of Electronic Commerce*, Vol. 6, No. 2, (2001) 35–59
7. Tschannen-Moran, M. and Hoy, W. K.: A conceptual and empirical analysis of trust in schools. *Review of Educational Research*, Vol. 71 (2000) 547-593.
8. Urban, G. L. The Trust Imperative. MIT Sloan School of Management Working Paper 4302-03 (2003)
9. Constantine, L. L., and Lockwood, L. A. D.: Instructive Interaction: Making Innovative Interfaces Self-Teaching. *User Experience*, Vol. 1, No. 3, Winter (2002)
10. Butler, J. K. & Cantrell, R.S.: A Behavioral Decision Theory Approach to Modeling Dyadic Trust in Superiors and Subordinates. *Psychological Reports*, Vol. 55 (1984), 81-105
11. Hosmer, L.T.: Trust: The Connecting Link Between Organizational Theory and Philosophical Ethics. *Academy of Management Review*, Vol. 20, (1995) 379–403.
12. Norman, D.: *The Psychology of Everyday Things*.
13. Rousseau, D., Sitkin, S. B., Burt, R., and Camerer, C.: Not So Different After All: A Cross-Discipline View of Trust. *The Academy of Management Review*, Vol. 23, No. 3 (1998) 393-404
14. Kim, J. Towards the Construction of Customer Interfaces for Cyber-Shopping Malls. *Electronic Markets*, Vol. 7, No. 2 (1997) 12-15
15. Egger, F. N.: Consumer Trust in E-Commerce: From Psychology to Interaction Design. In J. Prins (ed.) *Trust in Electronic Commerce*. Kluwer International (2002)
16. Zhou, X., and Liu, X.: Effective User Interface Design for Consumer Trust: Two Case Studies. Masters Thesis. Lulea University of Technology, Lulea, Sweden (2005)
17. Johnson, A.: Hotels Take 'Know Your Customer' to New Level. *The Wall Street Journal*, February 8, D1 (2006)