A Social-Feedback Enriched Interface for Software Download

Gianluca Dini, Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa, Italy

Pierfrancesco Foglia, Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa, Italy

C. Antonio Prete, Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa, Italy

Michele Zanda, Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa, Italy

ABSTRACT

Software downloading over the Internet is a major solution for publishers to deliver their software products. In this context, user interfaces for software downloading must be designed carefully. They should provide usable interactions and support users when deciding whether to accept the software product or not. This work proposes to enrich a common browser interface for software downloading with a reputation system - a mechanism for collecting and presenting user feedback. The reputation system is assessed with a usability study. The authors' results show that positive user rankings are effective in increasing user download acceptances for well-known publishers and common name publishers, as well as in increasing acceptance motivations related to trust aspects. In addition, the presence of the reputation system reduces incoherent user behaviors.

Keywords: *Downloading over Internet, Reputation System, Software Download, Usability, Web Interfaces*

INTRODUCTION

User interfaces for software download on the Internet are the front end of software publishers for remote users. The design of these interfaces is critical for users who may make unconscious or misguided decisions if interfaces are hard to understand and use (Brustoloni & Brustoloni, 2005; Kormann & Rubin, 2000). Problems may also arise for online publishers if hard tasks and complex interfaces reduce user trust, satisfaction and system usability at large. Software downloading also has security implications, and Brustoloni and Villamarin (2007) show that bad interface design can cause *unjustified risks* (breaches in the policy adopted to classify risks).

Addressing similar issues, our aim is to investigate the users' behavior while dealing with software download interfaces. In line with Brustoloni and Villamarin (2007), we observed *incoherent behaviors* in users interacting with the software download interface (Dini, Foglia, Prete, & Zanda, 2006, 2007). We define an *incoherent behavior* as a download decision which is not coherent with the motivation given

DOI: 10.4018/joeuc.2013010102

a-posteriori. In Dini et al. (2006) we observed that names of well-known (WK) publishers have a very strong brand effect that increases users' downloads, even if users claimed they wished to download free software only. From this, we inferred that users could have problems in understanding the information in the interface: they accepted software downloads according to the publisher's name, not the item cost. The interface was also problematic for publishers with no brand effect who had low acceptance rates. Hence, we proposed in Dini et al. (2007) a 3-step wizard interface asking for an explicit input on software cost. We observed improvements in user behaviors, with significant reductions of *incoherent behaviors*. However, the wizard forced users to be aware of cost and as a consequence users mainly accepted free software and refused software which entailed a charge.

This paper proposes and assesses the inclusion of a Reputation System (RS) in the common browser download interface. The RS was adopted to investigate whether it can mitigate the incoherent behaviors and increase trust in publishers with little or no brand effect.

In online service or product provisioning the client party often has little information on the service or product provider. Internet RSs collect, distribute, and aggregate feedback about past performances of a service or product provider. These systems have enjoyed widespread diffusion proving valuable in providing users with relevant information from previous users: RSs support the user in deciding who to trust, foster trustworthy behaviors, and deter malicious parties. "*Reputation systems seek to establish the shadow of the future to each transaction by creating an expectation that other people will look back on it*" (Resnick, Kuwabara, Zeckhauser, & Friedman, 2000). Because of such positive effects they have been adopted with success in many e-commerce systems (Chia, Heiner, & Asokan, 2010; Dellarocas & Resnick, 2003; Chen & Liu, 2011).

In our test the RS shows previous users' rankings upon deciding whether to download a piece of software or not. Our results show that positive rankings increase download acceptances for well-known publishers (49.75% vs. 34%) and impact also publishers with a common name (29.75% vs. 24%); positive rankings increase software acceptance motivations related to trust for common name publishers (16% vs. 6.9%). A further positive effect is that *incoherent user behaviors* are reduced for each type of software publisher (9.6% vs. 5.3%). Overall, the RS has an impact on user acceptances and motivations. This impact does not appear as considerable as regards the general acceptance/refusal averages, in contrast to that observed by Bolton, Katok, and Ockenfels (2004). Bolton et al. (2004) reported a significant improvement in transaction efficiency when the online marketplace is enriched with an RS. Their experiment was, however, different from ours: they followed a game theory approach wherein each user was a buyer half the time and seller half the time, trading undefined items.

The paper is structured as follows. The next two sections present the problem in detail and report works related to dialog-based interfaces for software download. The subsequent section describes the proposed reputation system. We report the usability study concerning the effects of adding a reputation system to the download interface. The experimental results and a discussion follow. Finally the conclusions of our work are given.

BACKGROUND AND HYPOTHESES

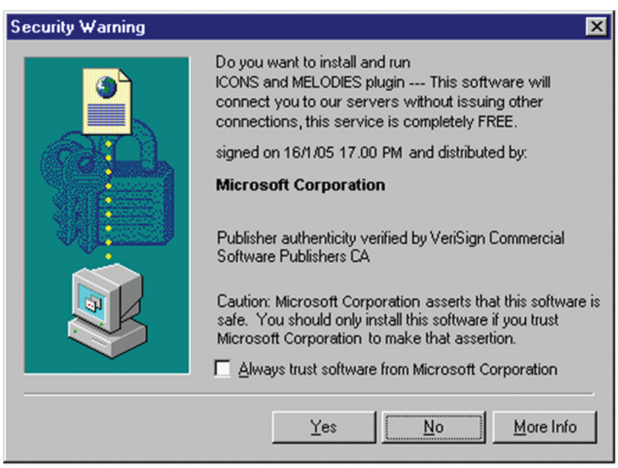
Issues in Online Software Delivery

Web interfaces for software download are increasingly relevant, as the Internet becomes a relevant means of software delivery (Gaudeul, 2010; Schonfeld, 2008). Online software delivery, like common e-commerce, must solve concerns especially regarding authentication and trust, in order to make users confident that they are being not cheated (CommerceNet, 2000; Corritore, Kracher, & Wiedenbeck, 2003). Complex Web interfaces and fraudulent software houses have caused serious problems such as spreading dialers, spywares and other threats (Shukla & Nah, 2005). Users cannot relate with a merchant and understand whether the software is in line with their expectations. In an ordinary shop, clients relate with a merchant and establish a trust relationship, and a similar relationship should be established online as "*without trust, development of e-commerce cannot reach its potential*" (Corritore et al., 2003; Cheskin Research and Studio Archetype/Sapient, 1999). Designing an usable and trustworthy interface for software download is important, since easy-to-use interfaces and complete information are highly correlated to user trust (Nilsson, Adams, & Herd, 2005; Laberge & Caird, 2000; Lanford, 2006), and a trustworthy interface is more likely to make users trust a vendor on the Internet (Fogg et al., 2001). Security symbols and trustworthy brand names proved to have a positive, lasting effect on trusting beliefs (Stoeklin-Serino & Paradice, 2009), and our studies have also confirmed this finding (Dini et al., 2006, 2007).

A trustworthy, usable interface would not solve two relevant questions for users dealing with software download from the Internet: first, the need to know who really published the file offered, second, supporting the user in understanding what to expect from the executable file. As a solution to the first problem (authentication), major software vendors have developed frameworks for code signing (Jansen, 2000; Thawte, 2007), such as the Microsoft Authenticode. By signing code, publishers seek to build a relationship of trust with users, satisfying the matter of accountability at the same time. Software publishers sign the piece of software they are releasing. The publisher's certificate, the Certification Authority (CA) certificate, and the signed code are then packaged together. At the client side, the browser verifies the publisher's signature on the code, and, if verifications are successful, the Security Warning dialog box (SWDB) is presented to the user (Figure 1). If some problems arise, the user is notified with a different dialog box. The SWDB presents the CA name, the software publisher name and the software name, as well as other details such as the cost and other optional information pieces. This information is shared by means of the different Authenticode versions that have been released over the years. Many frameworks can be used for code signing, but in this paper we have chosen investigate Authenticode due to its widespread diffusion on the Internet: 53% of the market share (BMS, 2012).

The further issue to be faced in software downloading concerns the user's expectations of the file. The Authenticode framework, like other code signing frameworks, ties a pub-

Figure 1. The security warning dialog box (SWDB)



lisher to a file via its CA, but it does not help the user understand what to expect from the file. Predictability in this scenario would be important as it influences trust (Corritore et al., 2003). Current software download frameworks do not give any safe information about what and how the executable file will behave once run on a computer system. Given that predictability cannot be fully ensured with the above approaches, in recent years Reputation Systems (RS) have emerged as a method for fostering trust amongst strangers cooperating online by gathering, distributing, and aggregating the feedback of previous consumers (Resnick et al., 2000).

Interaction Paradigms for Software Download

Deciding whether to accept or refuse a software download is a context-dependent security decision: "*In such situations, an application usually needs user input, because the application cannot determine automatically all the context relevant to the security decision*" (Brustoloni & Villamarin, 2007). For instance, receiving a software from a dear friend should be allowed if the sender is a friend, and should be denied if the sender is a malicious user. The authors classify the interaction paradigms that support such decisions in four classes: *warn-and-continue* (W&C), *no warning*, *no dialog*, and *Context Sensitive Guidance (CSG)*. A CSG interface asks the user to provide context information necessary for a security decision. In particular, in the *warn-and-continue* (W&C) approach the application warns the user of the risk and asks whether the user wants to accept it or not. Interfaces for software download (e.g., Authenticode security warning dialog box) follow the W&C approach.

In a previous paper (Dini et al., 2006), we observed that names of Well-Known (WK) publishers had a very strong brand effect that increased user downloads. When polled at the end of the tests, users claimed that they wished to download free software only. However, they largely accepted software from WK publishers, showing that the brand effect was more important than cost. We determined, thus, that users did not pay full attention to the whole interface, as they accepted or refused software downloads according to the publisher name, not the item cost. We also observed that such an interface can prevent Common Name (CN) publishers with little or null brand name from having their software downloaded. Both Brustoloni and Villamarin (2007) and our papers (Dini et al., 2006, 2007) show the problems deriving from the W&C approach: users click "continue" automatically, click through warnings due to habituation (Chia et al., 2010), do not pay attention to the whole information in the interface, and run into *incoherent behaviors* in Dini et al. (2007) or *unjustified risks,* defined as breaches in the policy adopted to classify risks, in Brustoloni and Villamarin (2007).

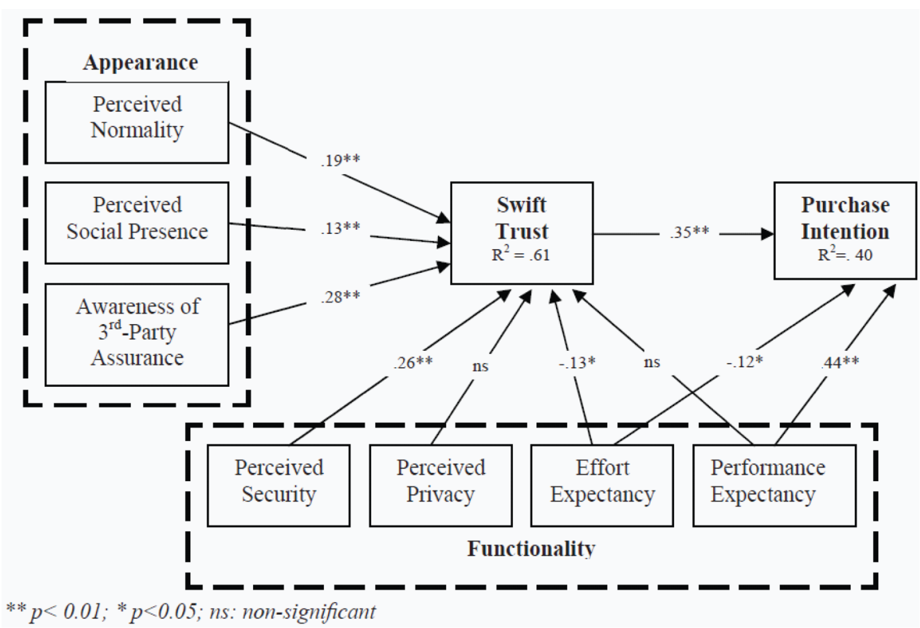
As a countermeasure to these problems, in Dini et al. (2007) we designed a 3-Step Wizard (3SW) interface with the purpose of increasing user attention to cost aspects. The 3SW is a CSG interface, as it requires input on software cost to ensure that the user is aware of it. Users interacting with the 3SW paid attention to cost, and based their decisions on cost by mainly accepting free software and refusing software which entails a charge. Brustoloni and Villamarin (2007) propose the adoption of polymorphic dialogs that change the form of the required user inputs, forcing users to pay attention to security decisions. Such a design proved effective in reducing *unjustified risks* with respect to other interaction paradigms. In line with Brustoloni and Villamarin, who observed a reduction of *unjustified risks*, we observed a reduction of *incoherent behaviors* (Dini et al., 2007) highlighting the same problems in the interaction paradigm. A behavior is incoherent when the motivation given for accepting or refusing a piece of software does not match with the software features, e.g., downloading software which entails a charge with motivation "free of charge." Our wizard, however, does not solve the (no-)branding effect for small and emerging publishers highlighted in previous works, but makes users focus mainly on cost.

Trust and Reputation Systems

Trust for Mayer, Davis, and Schoorman (1995) is the willingness of a trustor to be vulnerable to the actions of a trustee based on the expectation that the trustee will perform a particular action important to the trustor, irrespective of the ability to monitor or control the trustee.

Trust is driven by many variables, and upon deciding whether to accept or refuse a software download, the user should be able to predict if the software meets with his/her expectations, and predictability influences trust: "*predictability is a trustor's expectation that an object of trust will act consistently based on past experience*" (Corritore et al., 2003). To better understand and solve trust issues, Egger (2000) proposed a Model of Trust for Electronic Commerce (MoTEC), which includes prepurchase knowledge, interface properties such as usability and familiarity, informational content such as risk, transparency, and cooperation between consumers. Cooperation is relevant also for Olson (2000), who states that trust deals with behavior: "*people learn to trust others by noting their behavior.*" In the online domain, the end user has limited possibilities to fully try a software product or a service before buying, while the software publisher knows what he gets as long as he receives money. "*The inefficiencies resulting from this information asymmetry can be mitigated through trust and reputation*" (Josang, Ismail, & Boyd, 2007). Reputation Systems help end users in predicting whether a product can be trusted or not. Enforcing a trust belief in the end users towards a remote software provider is a subjective phenomenon influenced by a set of factors, and in absence of personal experience, trust must be based on third party reviews. For Josang et al. (2007), reputation is a collective measure of trustworthiness based on reviews from members in a community. Lin, Rong, and Thatcher (2009) validate a model wherein swift trust is influenced also by Perceived Social Presence (Figure 2), wherein swift trust refers to trust formed quickly in new or transitory relationships (Meyerson, Weick, & Kramer, 1996). According to Lin, "*social presence refers to consumers' perception that there is personal, sociable, and sensitive human*

Figure 2. Model of swift trust in web vendor (Lin et al., 2009)



contact and/or peer community on the Web site." This type of reassurance for online entities is similar to the reassurance provided by brick and mortar businesses (Gefen & Straub, 2004).

These claims support the adoption of Reputation Systems, as aggregators of reputation scores from a community, also to improve the efficiency in the process of software downloading. Resnick et al. (2000) state that these mechanisms can help people make decisions about who to trust and provide incentive for honest behavior, while deterring dishonest parties from participating. Despite many problems in theory (Resnick et al., 2000), in practice RSs proved to work rather well as a means to provide relevant information on the quality of merchants and products in auction sites. Successful examples are represented by iTunes and its reputation system for both desktop and mobile users, by the E-Bay auction site or online book sellers like Amazon (Chia et al., 2010; Dellarocas & Resnick, 2003; Chen & Liu, 2011). RSs have been proposed as a means to identify inauthentic content in file sharing applications (Walsh & Sirer, 2006), and similarly they could be effective in helping users to identify deceptive software publishers.

Reputation Systems proved to improve markets efficiency and minimize risks for customers, hence our research question is: *do Reputation Systems positively support users in the process of software downloading?* In order to replicate an everyday scenario, we differentiate the reviews in the RS into three classes of software publishers: Well-Known (WK) publishers, publishers with a Common Name (CN) and publishers with Deceptive Names (DN), each class having different scoring patterns. Hence the research question is investigated by testing two hypotheses that compare the RS enriched SWDB interface to the plain SWDB interface:

H1: *Positive rankings in an RS embedded in the SWDB interface increase the acceptance rate for WK and CN publishers compared to the plain SWDB.*

H2: *Negative rankings or no rankings in an RS embedded in the SWDB reduce the acceptances of software by DN publishers compared to the plain SWDB.*

The third hypothesis to investigate the research question concerns how users are influenced by different rankings shown in the RS boxes:

H3: *Positive rankings in an RS embedded in the SWDB interface increase the acceptance rate for WK and CN publishers compared to no rankings or negative rankings.*

In order to better understand user behaviors, in addition to acceptances and refusal rates, we also investigated user motivations concerning either a software download or a refusal, and analyzed the answers to a final questionnaire about the whole experiment.

Unlike other studies which investigated RSs with agent-based approaches (Fullam et al., 2005; Schlosser, Voss, & Bruckner, 2006; Boella & Remondino, 2010) or online forums statistical analyses (Garcin, Faltings, Jurca, & Joswig, 2009; Talwar, Jurca, & Faltings, 2007), we assessed the RS with a usability test recruiting a group of participants similar to Chia et al. (2010). Chia et al. (2010) designed a system architecture to support personal social feedback comments and show them to the user while deciding whether to download software on a smart phone. The designed architecture reduced unsafe actions, with users being slowed down with habituation-breaking mechanisms which made them focus on the social comments.

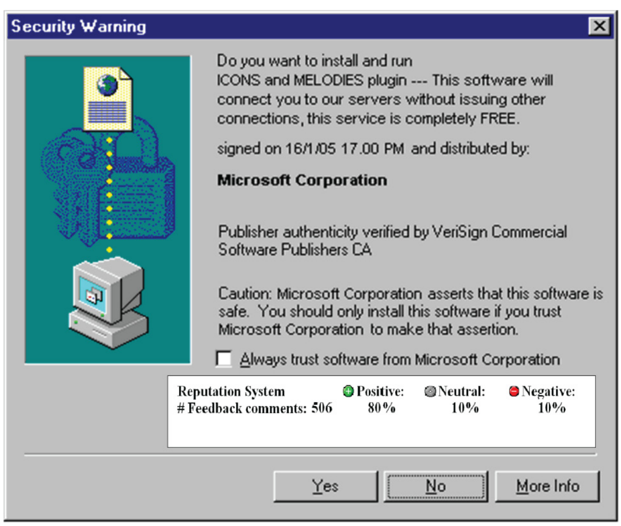
OUR PROPOSAL: REPUTATION SYSTEM IN THE DOWNLOAD INTERFACE

This paper proposes a context sensitive interface that embeds a Reputation System showing aggregate rankings regarding the software to be downloaded. Our solution can be added to current code signing frameworks, like that of Authenticode. We embedded the Reputation System (RS) into the software download interface of the Authenticode framework (Figure 3).

As shown in Figure 3, we placed the RS at the bottom of the dialog box, showing the total number of feedback reviews, as well as the rate of positive, neutral, and negative reviews. According to the classification of RS ranking aggregation mechanisms proposed by Josang et al. (2007), our interface, like that of the EBay Feedback Forum, is a simple summation system. In a simple summation RS, the system accumulates all given ratings to get the overall reputation. We did not include a link connected to an external Web page with user feedback comments given that in our previous works (Dini et al., 2006, 2007), we observed that participants little use links in the download dialog box under test, and presenting the user a lot of information can cause information overload and incoherent behaviors. Thus, we designed a rather simple RS, readable at a glance, providing a concise view of the distribution of users' feedback. Our work differs from (Brustoloni & Villamarin, 2007) in that our interface is context dependent but does not require user input, and it is different from (Chia et al., 2010) in that we do not focus on the mobile environment and are not modifying the code-signing architecture.

We do not address the implementation details of the RS here because we are more interested in investigating whether and how the RS can support the users interacting with the downloading interface. If the RS is actually adopted in the context of software downloading, it can be deployed preferring a centralized approach (Josang et al., 2007), implemented by means of an object-oriented distributed technology (Bechini, Foglia, & Prete, 2002). In fact, distributed RS are more suitable for peer-to-peer transactions, while we are considering a scenario wherein users download software from the publisher or a central repository. Concerning a deployment in the real world, user reviews can be requested by the downloaded software itself after a number of uses or by sending a

Figure 3. Security warning dialog box (SWDB) with the Reputation System (RS)



feedback request to users' email addresses like in Tripadvisor.com, with reviews collected by a third party and presented in the download interface.

METHOD AND EVALUATION

Populating the Reputation System

With the purpose to investigate if and how the presence of a Reputation System influences user behaviors, we performed a within-groups experiment by proposing each participant two interfaces: the classic SWDB interface and the SWDB with the RS. The RS was inspired by the E-Bay Feedback Forum also described by Resnick et al. (2000), with the purpose to modify user trust beliefs about software publishers while dealing with a software download interface. The RS effect is assessed by looking at the software download acceptances rates and by considering the motivations given for each acceptance or refusal.

The RS must be populated properly, in order to simulate an everyday scenario for all three classes of publishers tested in the experiment, following our three major hypotheses (*H1*, *H2*, *H3*). Visiting the E-Bay Feedback Forum, we observed that usually well-known merchants have a high and positive reputation ranking, unless they decide to have no ranking at all as their brand effect (e.g., Microsoft) is strong enough, even without the RS support. On the other hand, deceptive merchants who may provide an RS, present few comments which are mostly negative. Finally, common name merchants tend to support the RS as they want to achieve visibility and, contextually, be trusted by end users. Thus, in our experiments, we do not simulate the population of the RS, but instead we choose to assume that the RS is already populated in line with the above observations. In particular: well-known publishers have a *high and positive* RS, or *do not support it*; deceptive publishers have a *low and negative* RS, or *do not support it*; publishers with a common name have either *high and positive* or *low and negative* RS ranking. The reason was twofold: reducing the number of test cases for each user, and simulating real life scenarios by associating appropriate reputation ranks to the three classes of publishers, while optimizing the time to complete the experiment.

In detail, the actual rankings that we had in the experiments are shown in Table 1: a high ranking means that the number of reviews is greater than 400, with 80% positives, 10% neutrals, 10% negatives; a low ranking means that the number of feedback is smaller than 40, with 30% positives, 30% neutrals, 40% negatives; and finally the publisher can decide to not support the reputation system.

Test Procedure

The study included the following independent variables.

- *Software publisher name*. Software publishers were varied according to three classes: (i) well-known software publishers (WK), (ii) common name publishers (CN), and (iii) companies whose name was deceptive (DN).
- *Software cost.* It was varied between *free* and *charge entailed* (above 20 €).
- *Reputation System ranking.* The RS was varied with the following scheme:

Table 1. The three rankings associated to the software publishers in the usability test

RS Ranking	Number of reviews	Positive	Neutral	Negative
High and positive	> 400	80%	10%	10%
Low and negative	< 40	30%	30%	40%
Not Supported	-	-	-	-

    - *High and positive rankings* for WK publishers and CN publishers;
    - *Low and negative ranking*s for CN publishers and DN publishers;
    - *Not supported*: the publisher decides not to support a Reputation System, for WK publishers and DN publishers.
- *Interface for software downloading.* Participants interacted with two interfaces to perform differential analyses: the original SWDB and the SWDB enriched with the RS.

Each user was shown a complete permutation of the independent variables. Each participant completed 18 test cases, obtained from the independent variables: (i) with the SWDB interface, publisher name (3 values) and cost details (2 values); (ii) with the SWDB + RS interface, publisher name (3 values), cost details (2 values) and RS ranking (2 values). The user test consists of a sequence of test cases, where each test case consists of three stages (Figure 4). In the first stage we display a web page that proposes a software download. In the second stage, the user decides whether to accept or not the piece of software, by using one of the interfaces. In the third stage, the user is required to motivate their choice with a *motivation radio-button*. A user, following their decision of code acceptance or code refusal, must choose, with a radio-button, the *major motivation* that drove their decision. The given motivations, together with the acceptance or refusal of a piece of software, are dependent variables in our experiments.

The motivation radio-button presents a set of options for participants to choose. The content of this window is described in Table 3 (column Motivation). If the user downloads a piece of software, they are shown a list of motivations mainly related to interest (I), trust (T), free cost (F) and other minor motivations (O). If the user refuses the download, they are shown another list of motivations such as no interest (NI), distrust (DT), high cost (C) and other minor ones (O).

A gender balanced group of 43 participants was recruited, with ages ranging from 18 to 40. 34 participants were high school students or

Figure 4. Test procedure

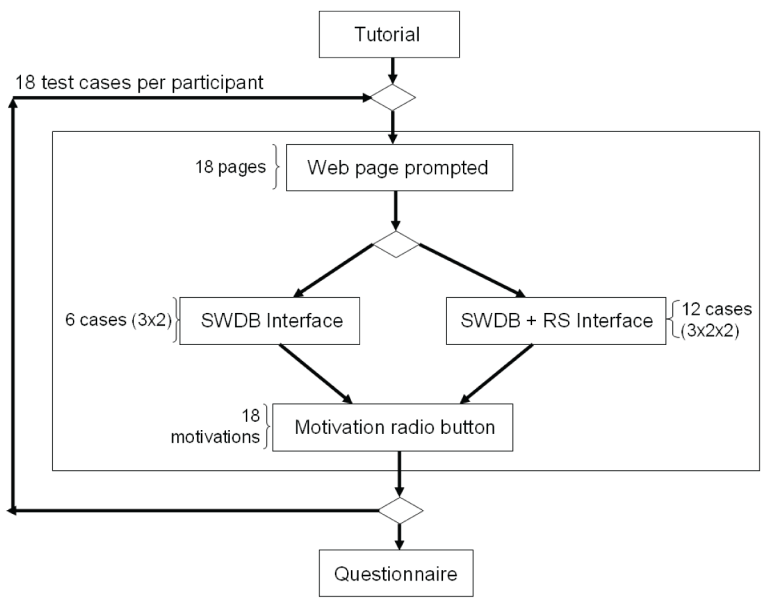


Table 2. Sample characteristics

Question	Answers			
How often do you use the Internet?	Rarely (-)	Once a month (-)	Once a week (4)	Every day (36)
Why do you use the Internet?	Work (5)	Study (31)	Entertainment (4)	Other (-)
Do you usually download software?	Never (-)	At times (13)	Frequently (27)	
As a Internet user, have you ever seen the "Security Warning" dialog box?	Yes (40)	No (-)	Don't remember (-)	
How do you usually behave when confronted with a "Security Warning" dialog box?	Click Yes (8)	Click No (5)	It depends (27)	
Do you usually pay attention to what is written in the "Security Warning" dialog box?	Yes (15)	No (11)	At times (14)	

undergraduate students, while 6 participants were employed in public administration. 3 students were dropped in the initial stages. The study participants attended the tests in person in our labs or in their office (rather than remotely via web). The recruited group had good Internet experience, as shown in Table 2.

At the end of the 18 test cases, participants answered a multiple-choice questionnaire, with questions on the participant and on the relevant interface (questionnaire in the Appendix). Given that each user had to complete 18 test cases, we minimized learning effects by not giving participants feedback on their actions during the experiment.

In our previous work (Dini et al., 2006), we observed that user intentions and proper behaviors for each single test case could not be derived by just looking at the final questionnaire answers, as each test case needed a

Table 3. Motivation options to be chosen in each test case, and related incoherent motivations. Labels identify specific motivation options, those labeled with O were chosen in minor quantity by participants.

Decision	Motivation	Label	Incoherent, if in Combination With
Acceptance	I was very interested in the product	I	
	I was not very interested, but I trust the code publisher	T	*a deceptive publisher name*
	I was not very interested, but the cost was fine	O	*a free code*
	I was not very interested, but it was free	F	*a code entailing a charge*
	I didn't want to accept it	O	*(explicit error)*
	Other motivations	O	
Refusal	I was not interested in it at all	NI	
	I could be interested, but I didn't trust the code publisher	DT	
	I could be interested, but it was expensive	C	*free code*
	I could be interested, but even if it was free, I refused	O	*a code entailing a charge*
	I didn't want to refuse it	O	*(explicit error)*
	Other motivations	O	

precise motivation. For this reason, we let users choose the most suitable motivation out of a set (Table 3), after either accepting or refusing a downloading (Dini et al., 2007). By doing this, users can motivate their decisions with precision. In addition, we used user motivations to clearly identify participants' coherent and incoherent behaviors. In fact, in the cited previous works, we observed that users interacted with interfaces incoherently at times: a participant behaves incoherently if their motivation does not match with the features of a piece of software downloaded or refused. In Table 3, column "*Incoherent, if in combination with*" shows how motivations not matching actual software features identify incoherent behaviors. An example of an incoherent behavior would be if a participant refuses a download picking the motivation "*I could be interested, but it was expensive,*" while the software is free of charge.

Data Analysis

All user actions were logged on a database for subsequent analyses. Results presented in the paper are given reporting raw percentages, or statistical analyses such as ANOVA or Chi-Square (χ^2). ANOVA (ANalysis Of VAriance) was applied to analyze paired groups of users, while Chi-square was adopted to compare two groups with different cardinalities. Each statistical test provides a p-value that indicates the probability that the observed difference is due to chance. Usually $p<0.05$ is commonly accepted to indicate a statistically significant difference in the sample (Glantz, 2005).

RESULTS

Acceptance/Refusal Analysis

This section reports whether the three main hypotheses were confirmed or not by the experiments. Detailed results are reported including details on such hypotheses and further results deriving from the statistical analysis of user behaviors in the usability study.

H1: Positive rankings in an RS embedded in the SWDB interface increase the acceptance rate for WK and CN publishers compared to the plain SWDB. - *Confirmed for WK publishers; Not confirmed for CN publishers*.

H2: Negative rankings or no rankings in an RS embedded in the SWDB reduce the acceptances of software by DN publishers compared to the plain SWDB. - *Confirmed*.

H3: Positive rankings in an RS embedded in the SWDB interface increase the acceptance rate for WK and CN publishers compared to no rankings or negative rankings. - *Confirmed.*

Table 4 lists results for the classic interface (SWDB) with no Reputation System, while Table 5 lists results for the classic interface (SWDB) with the RS. Table 5 shows the acceptance and refusal percentages free software and software entailing a charge, with results detailed for publisher class (WK, CN, DN) and for reputation rankings ("*RS:*" labeled rows).

The RS rankings influenced user behaviors. The RS has two major effects when embedded into the SWDB: the RS ranking influences download acceptances and refusals (Table 5) and reduces incoherent behaviors (Table 6: *average* from 9.6% to 5.3%).

First of all, by looking at average acceptances and refusals with the SWDB and the SWDB + RS (Table 4 and Table 5), we observe minimal changes in user behaviors while dealing with WK and CN publishers in the two interfaces. The same does not hold for DN publishers: either a low or absent RS ranking for these publishers reduced (31.25% in Table 4, vs. 20.75% in Table 5) user download acceptances significantly (χ^2 (1,40): p=0.039). The Reputation System influenced participants not only in relation to DN publishers. Table 5 shows that acceptances (and dually refusals) are influenced by different review ratings. The acceptances of WK publishers offering free pieces of software with a high reputation rank in the SWDB + RS are significantly higher (F(1,40): p=0.013) than free WK products with

Table 4. Acceptance and refusal rates with the plain Security Warning dialog box (SWDB). Cells in grey are data analyzed in the text; percentages underlined red are major drawbacks of SWDB without the Reputation System.

	SWDB					
Interface	**Acceptances (%)**			**Refusals (%)**		
Code type	Free	Charge entailed	Sum	Free	Charge entailed	Sum
WK	40	23.75	63.75	10	26.25	36.25
CN	30	6.25	36.25	20	43.75	63.75
DN	22.5	8.75	31.25	27.5	41.25	68.75
Average	31	13	-	19	37	-

Table 5. Acceptance and refusal rates for the interface with the Reputation System. Results are given for each code publisher name type, detailing for the reputation system scores. Cells in grey are data analyzed in the text; percentages underlined green are major benefits of the Reputation System.

	SWDB + RS									
Interface	**Acceptances (%)**					**Refusals (%)**				
Code type	Free		Charge entailed		Sum	Free		Charge entailed		Sum
WK	41.88		19.38		61.26	8.12		30.62		38.74
RS: high | n.p.	*49,75*	*34*	*20*	*18,75*	-	*0,25*	*16*	*30*	*31,25*	-
CN	26.88		5		31.88	23.12		45		68.12
RS: high | low	*29,75*	*24*	*8*	*3*	-	*20,25*	*26*	*41,5*	*48,5*	-
DN	15.75		5		20.75	34.25		45		79.25
RS: n.p. | low	*18,5*	*13*	*5*	*5*	-	*31,5*	*37*	*45*	*45*	-
Average	28.2		9.8		-	21.8		40.2		-

Table 6. Incoherent behaviors with the two interfaces tested in the study

	Incoherent behaviors	
Interface →	**SWDB**	**SWDB + RS**
WK	6.25%	5.25%
CN	6.25%	3.13%
DN	16.25%	7.5%
Average	9.6%	5.3%

the SWDB interface (40% vs. 49.75%). A high reputation ranking correlates to higher acceptances (F(1,40): p=0.047) than a low reputation ranking in CN publishers (29%, 24% vs. 8%, 3%).

RS rankings in WK publishers (high RS or absent RS) influence software acceptances significantly (χ^2 (1,40): p=0.034) depending on whether there are positive rankings or no rankings (49.75% vs. 34%). Well-Known publishers at times did not present feedback comments as we observed that famous publishers rely on their brand effect. However, despite the power of their brand with respect to the other classes of publishers, having a positive reputation system is beneficial also for this class of publishers.

The RS rankings of DN publishers (low RS or absent RS) affected user acceptances significantly (χ^2 (1,40): p=0.043) as well (31.25% vs. 20.75%). This is not unexpected: the experiment was designed in order to support WK publishers with positive rankings or no RS (brand effect), disadvantage DN publishers with negative ranks or no RS, differentiate CN publishers with either positive or negative rankings. Thus, reducing acceptances of pieces of software by DN publishers shows that users have paid attention to the information provided by the Reputation System box in the SWDB interface. Finally, the acceptances averages show that the RS does not have a huge impact on the general trends among all three classes of publishers.

The presence of feedback rankings created a higher interactive interface that made users focus on the Reputation System and on other pieces of information of the SWDB as well. Table 6 gives the incoherent behavior rates that were observed in the two interfaces: the RS presence causes less incoherent behaviors than the SWDB. Such a beneficial effect of an interactive interface is in line with the decision-tree based interface tested by Likarish, Dunbar, Hourcade, and Jung (2009).

To further analyze such incoherent behaviors, we also asked in the questionnaire "*how much are you willing to spend to use a piece of software?*" with a predefined set of answers from zero to other amounts of money. 82.5% of the participants chose the answer "*Zero, I don't want to spend money.*" Hence, we looked into the logs of these users, to see if they had accepted software which entailed a charge, making a decision that according to our classification could not be considered an incoherent decision. These would have been low aware behaviors, because it is likely that the interfaces were not communicating all the information effectively to the participants. In addition, we observed that 50% of the participants who do not want to spend money accepted at least once code which entailed a charge while dealing with the SWDB, 34% while dealing with the SWDB + RS.

Motivations

Table 7 reports the acceptance and refusal motivations with the basic SWDB and the SWDB with the Reputation System, showing that participants were influenced by the reviews from previous users.

The reported motivations if the user downloads the software are Interest (I), Trust in publisher (T), Free of charge (F), with percentages reported for the three classes of publishers (WK, CN, DN). The same structure holds for motivations for software refusal, which vary among Not Interested (NI), Distrust (DT), Cost (C). Columns labeled "Os" show further minor motivations (Table 3).

Feedback reviews in the SWDB interface have a beneficial effect for Common-Name publishers, reducing their gap in trust (T) with Well-Known publishers: from 23.5%-6.9% to 17%-16% (Table 7). The RS lowers acceptances (χ^2 (1,40): p=0.026) based on Trust for pieces of software by DN publishers (20% vs. 9%), which must be considered a beneficial effect (Table 7). However, we observe that the RS has little effect on refusal based on distrust (61.8% vs. 58%) for DN publishers (Table 7).

With the RS, the gap between WK publishers and CN publishers is large where DT refusal motivations are concerned (11% vs. 30%), but in this case if we investigate DT refusals for CN publishers it can observed that 70% of

Table 7. Motivations for acceptance/refusal with the SWDB interface and the SWDB + RS, respectively. Cells in grey are data analyzed in the text; percentages in bold and underlined are major benefits of the Reputation System.

		Interface							
Decision		**SWDB**				**SWDB + RS**			
Accept	Motivation →	I	T	F	Os	I	T	F	Os
	WK	49%	23.5%	25.5%	2%	61%	17%	20%	2%
	CN	72.4%	6.9%	20.7%	0%	49%	**16%**	29%	6%
	DN	52%	20%	20%	8%	53%	**9%**	32%	6%
	Average	58%	17%	22%	3%	54%	14%	27%	4%
Refuse	Motivation →	NI	DT	C	Os	NI	DT	C	Os
	WK	58.6%	13.8%	13.8%	13.8%	32%	11%	45%	12%
	CN	23.5%	39.2%	21.6%	15.7%	34%	30%	29%	7%
	DN	16.4%	61.8%	10.9%	10.9%	20%	58%	14%	8%
	Average	33%	38%	15%	13%	29%	33%	29%	9%

DT motivations were due to a low ranking in the Reputation System. This means that users paid attention to the RS rankings and were influenced by such extra information: only CN publishers had two rankings to either increase (high rank) or decrease (low rank) acceptances, while WK publishers had rankings supposed to increase acceptances and DN ones had rankings to decrease acceptances. We observed no significant changes in the motivations for WK and DN publishers with the two rankings tested for each class of publisher.

The RS presence influenced participants' motivations as shown on Table 7 also driving higher refusals based on cost (C): on average from 15% to 29%, and this is highly significant (χ^2 (1,40): p=0.015) for WK publishers (13.8% vs. 45%). Through participant observation and interviews we infer that users paid attention to the RS but also increased the attention to the overall interface: reading user feedback reviews, they looked at the whole information in the dialog box. User involvement increased and users reduced the acceptances of pieces of software where a charge was entailed. In line with our previous studies (Dini et al., 2006, 2007), participants in this experiment also wanted to avoid pieces of software where a charge was entailed.

Questionnaire

Answering the questionnaire 85% of the participants already knew what a reputation system and feedback reviews were. On a 5-point Likert scale, recruited participants felt safer with the reputation system at 3.7, and in addition they stated that showing the exact number of reviews was important at 4.0.

Results Validity

There were three ranking categories shown in the Reputation System in our test: high and positive rankings, low and negative rankings, and rankings not supported. Although we observed that rankings tend to be classified mainly in line with these three categories, further studies should investigate how users behave while dealing with other reputation patterns.

We minimized learning effects by not giving the users any feedback on each single test cases (e.g., free acceptances/acceptances entailing a charge, incoherent motivations). We observed that the actions taken by the users with tested interfaces were very conservative: they tried to minimize possible risks (even if they were aware that the experiment could not harm their terminal). Thus, we believe that the test environment had little impact on our results.

Discussion

To sum up the results, we observed that positive reputation rankings increase download acceptances for Well-Known publishers and publishers with a Common Name (*hypothesis H3, and H1 for WK publishers*), and negative rankings reduce acceptances for deceptive publishers (*hypothesis H2*). Positive reputation rankings increase software acceptances based on trust for publishers with a Common Name. The higher interactivity of the RS interface reduces users' incoherent behaviors for each type of software publisher. However, the RS influences user decisions, but it seems not to have a huge impact on the general acceptance/refusal trends observed with the classic download interface. Considering previous studies, the impact of the RS rankings in our experiment was not as relevant as that observed by Bolton et al. (2004). Bolton et al. arranged a game theory experiment wherein an RS enriched marketplace improved efficiency, trust and trustworthiness of transactions significantly with respect to an RS-less marketplace. However, such experiment did not investigate publisher name brand effects, with participants having the only goal of maximizing their profit by selling or buying undefined items.

Looking at our experimental findings, we recommend a broader adoption of reputation systems on the Internet, extending their reach also to browser interfaces devoted to software download, in line with the market success that reputation systems have had in other areas (auction sites, online bookstores, hotel booking services, …). Clearly, we highlight that having an RS is only beneficial for publishers that offer a good product and are thus able to receive positive reviews from the users. From a theoretical point of view, this study supports all studies which reported beneficial effects of RSs for online product delivery or service provisioning, and it supports the inclusion of social interactions (user feedback, reviews, rankings) to establish trust between remote parties.

CONCLUSION

Our work investigates the impact of embedding a Reputation System (RS) into a browser interface for software downloading. Such interfaces are critical for relating software providers to their customers, given that the Web is a large marketplace for distributing software products. In our previous study (Dini et al., 2006) users mainly accepted or refused software downloads according to the publisher name, as they were following the publisher brand effect. However, in the questionnaire users claimed they wished to refuse software which entailed a charge, showing discrepancies between their actions and their claims. We defined incoherent behaviors and almost nullified them in another study (Dini et al., 2007), wherein we tested a wizard interface requiring an input on software cost. The wizard proved over effective, making cost by far the major motivation for either accepting or refusing software downloading. In this work, we tested the adoption of a Reputation System to better support users in deciding whether to trust or distrust a software publisher, also reducing the incoherencies. As a starting point, due to its widespread diffusion, we took the design of the Authenticode interface for software downloading and enriched it with a Reputation System showing feedback reviews from previous users.

We had RS rankings for three classes of software publishers: Well-Known publishers, Common-Name publishers and Deceptive-Name publishers. Each single class had RS rankings that reproduced real life scenarios: WK publishers had positive rankings or did not support at all a RS (they have the brand effect), CN publishers had negative rankings or did not support the RS, while CN publishers had both positive and negative rankings.

The different rankings in the Reputation System proved effective in significantly influencing software acceptances/refusals as well as user motivations. When the RS is present, positive rankings increase user download acceptances for both WK publishers and CN publishers. Having an RS showing positive reviews, most likely due to a good product, is beneficial also for WK publishers that increase user downloads with respect to their acceptances based only on their brand effect. The RS does not increase distrust refusal motivations for deceptive publishers, while the RS and its feedback comments reduce users' incoherent behaviors for all classes of publishers.

REFERENCES

Bechini, A., Foglia, P., & Prete, C. A. (2002). Use of a CORBA/RMI gateway: Characterization of communication overhead. In *Proceedings of the ACM Workshop on Software and Performance*, Roma, Italy (pp. 150-157).

Boella, G., & Remondino, M. (2010). Collaboration and human factor as drivers for reputation systems effectiveness. In J. Cordeiro & J. Filipe (Eds.), *Proceedings of the 5th International Conference on Web Information Systems and Technologies* (LNBIP 45, pp. 3-16).

Bolton, G. E., Katok, E., & Ockenfels, A. (2004). How effective are electronic reputation systems? An experimental investigation. *Management Science*, *50*(11), 1587–1602. doi:10.1287/mnsc.1030.0199

Browser Market Share (BMS). (2012). *Desktop top browser share trend*. Retrieved from http://marketshare.hitslink.com/browser-market-share.aspx?qprid=1&qpcustomb=0

Brustoloni, J. C., & Villamarin-Salomon, R. (2007). Improving security decisions with polymorphic and audited dialogs. In *Proceedings of the Symposium on Usable Privacy and Security* (pp. 76-85).

Brustoloni, X., & Brustoloni, J. C. (2005). Hardening Web browsers against man-in-the-middle and eavesdropping attacks. In *Proceedings of the 14th Conference on World Wide Web* (pp. 489-498).

Chen, M., & Liu, X. (2011). Predicting popularity of online distributed applications: iTunes app store case analysis. In *Proceedings of the ACM iConference* (pp. 661-663).

Cheskin Research and Studio Archetype/Sapient. (1999). *Ecommerce trust study*. Retrieved from http://www.cheskin.com

Chia, P. H., Heiner, A., & Asokan, N. (2010). *The wisdom of cliques: Use of personalized social rating for trustworthy application installation* (Tech. Rep. No. NRC-TR-2010-001). Helsinki, Finland: Nokia.

CommerceNet. (2000). *Barriers to electronic commerce 2000 study*. Retrieved from http://www.commerce.net

Corritore, C. L., Kracher, B., & Wiedenbeck, S. (2003). On-line trust: Concepts, evolving themes, a model. *International Journal of Human-Computer Studies*, *58*, 737–758. doi:10.1016/S1071-5819(03)00041-7

Dellarocas, C., & Resnick, P. (2003). *Online reputation mechanisms a roadmap to future research: Summary report of the first Interdisciplinary Symposium on Online Reputation Mechanisms*. Cambridge, MA: MIT.

Dini, G., Foglia, P., Prete, C. A., & Zanda, M. (2006). An analysis of user interface factors influencing the acceptance of code download. In *Proceedings of the 28th IEEE Information Technology Interfaces Conference* (pp. 233-238).

Dini, G., Foglia, P., Prete, C. A., & Zanda, M. (2007). Effects of increasing users' attention on cost in software download interfaces on the Internet. In *Proceedings of the 29th IEEE International Conference on Information Technology Interfaces* (pp. 227-232).

Egger, F. N. (2000). "Trust me, I'm an online vendor": Towards a model of trust for e-commerce system design. In *Proceedings of the Extended Abstracts on Human Factors in Computing Systems* (pp. 101-102).

Fogg, B. J., Mashall, J., Laraki, O., Osipovic, A., Varma, C., & Fang, N. …Treinen, M. (2001). What makes Web sites credible? A report on a large scale quantitative study. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (pp. 61-68).

Fullam, K. K., Klos, T. B., Muller, G., Sabater, J., Schlosser, A., & Topol, Z. …Voss, M. (2005). A specification of the agent reputation and trust (ART) testbed: Experimentation and competition for trust in agent societies. In *Proceedings of the International Conference on Autonomous Agents and Multi Agent Systems* (pp. 512-518).

Garcin, F., Faltings, B., Jurca, R., & Joswig, N. (2009). Rating aggregation in collaborative filtering systems. In *Proceedings of the Third ACM Conference on Recommender Systems* (pp. 349-352).

Gaudeul, A. (2010). Software marketing on the Internet: The use of samples and repositories. *Economics of Innovation and New Technology*, *19*(3), 259–281. doi:10.1080/10438590802472572

Gefen, D., & Straub, D. (2004). Consumer trust in B2C e-commerce and the importance of social presence: Experiments in e-products and e-services. *Omega*, *32*, 407–424. doi:10.1016/j.omega.2004.01.006

Glantz, S. A. (2005). *Primer of biostatistics*. New York, NY: McGraw-Hill.

Jansen, W. A. (2000). Countermeasures for mobile agent security. *Computer Communications*, *23*(17), 1667–1676. doi:10.1016/S0140-3664(00)00253-X

Josang, A., Ismail, R., & Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision Support Systems – Emerging Issues in Collaborative Commerce, 43*(2), 618-644.

Kormann, D. P., & Rubin, A. D. (2000). Risks of the passport single sign-on protocol. *Computer Networks, 33*(1-6), 51-58.

Laberge, J., & Caird, J. K. (2000). Trusting the online banking interface: Development of a conceptual model relevant to e-commerce transactions. In *Proceedings of the Designing Interactive Systems for 1-to-1 e-Commerce Workshop at the ACM SIGCHI Conference*, The Hague, The Netherlands.

Lanford, P. (2006). E-commerce: A trust perspective. In *Proceedings of the International Conference on Semantic Web and Web Services, the World Congress in Computer Science, Computer Engineering, and Applied Computing*.

Likarish, P., Dunbar, D., Hourcade, J. P., & Jung, E. (2009). Bayeshield: Conversational anti-phishing user interface. In *Proceedings of the Symposium on Usable Privacy and Security* (article 26).

Lin, X., Rong, G., & Thatcher, J. B. (2009). Swift trust in web vendors: The role of appearance and functionality. *Journal of Organizational and End User Computing*, *21*(1), 88–108. doi:10.4018/joeuc.2009092205

Mayer, R., Davis, J., & Schoorman, F. (1995). An integrative model of organizational trust. *Academy of Management Review*, *20*(3), 709–734.

Meyerson, D., Weick, K., & Kramer, R. (1996). Swift trust and temporary groups. In Kramer, R. M., & Tyler, T. R. (Eds.), *Trust in organizations: Frontiers of theory and research* (pp. 166–195). Thousand Oaks, CA: Sage. doi:10.4135/9781452243610.n9

Nilsson, M., Adams, A., & Herd, S. (2005). Building security and trust in online banking. In *Proceedings of the Extended Abstracts on Human Factors in Computing Systems* (pp. 1701-1704).

Olson, J. S., & Olson, G. M. (2000). i2i trust in ecommerce. *Communications of the ACM*, *43*(12), 41–44. doi:10.1145/355112.355121

Resnick, P., Kuwabara, K., Zeckhauser, R., & Friedman, E. (2000). Reputation systems. *Communications of the ACM*, *43*(12), 45–48. doi:10.1145/355112.355122

Schlosser, A., Voss, M., & Bruckner, L. (2006). On the simulation of global reputation systems. *Journal of Artificial Societies and Social Simulation*, *9*(1), 1–23.

Schonfeld, E. (2008, November 13). The death of packaged software: Microsoft store opens on the Web for U.S. shoppers. *TechCrunch*.

Shukla, S., & Nah, F. F. H. (2005). Web browsing and spyware intrusion. *Communications of the ACM*, *48*(8), 85–90. doi:10.1145/1076211.1076245

Stoecklin-Serino, C., & Paradice, D. (2009). An examination of the impacts of brand equity, security, and personalization on trust processes in an e commerce environment. *Journal of Organizational and End User Computing*, *21*(1), 1–36. doi:10.4018/joeuc.2009010101

Talwar, A., Jurca, R., & Faltings, B. (2007). Understanding user behavior in online feedback reporting. In *Proceedings of the 8th ACM Conference on Electronic Commerce* (pp. 134-142).

Thawte. (2007). *Certify your software integrity with Thawte code signing certificates* (Tech. Rep.). Cape Town, South Africa: Thawte.

Walsh, K., & Sirer, E. G. (2006). Experience with a distributed object reputation system for peer-to-peer filesharing. In *Proceedings of the USENIX Symposium on Networked System Design and Implementation* (Vol. 3).

Gianluca Dini received his "Laurea" degree in Electronics Engineering from the University of Pisa, Pisa, Italy, in 1990, and his PhD in Computer Engineering from Scuola Superiore "S. Anna," Pisa, Italy, in 1995. From 1993 to 1994 he was Research Fellow at the Department of Computer Science of the University of Twente, The Netherlands. From 1993 to 1999 he was Assistant Professor at the Dipartimento di Ingegneria della Informazione, Elettronica, Informatica, Telecomunicazioni of the University of Pisa, where he is now Associate Professor and Vice-Director. His research interests are in the field of distributed computing systems, with particular reference to security. Currently he is working on network security and security in networked embedded systems. He has published more than eighty papers in international conferences, books and journals and has served as coordinator or principal investigator in many projects including those funded by the Commission of the European Community, the Italian Government and private companies.

Pierfrancesco Foglia is assistant professor at the Dipartimento di Ingegneria dell'Informazione of the Università di Pisa, Italy, where he teaches Informatics and Industrial informatics. His research interests include computer architectures, embedded systems and usability in computer systems, and he has published more than 50 papers in conferences, journals and book chapters on the topics. He served as General Chair for the HIPEAC 2010 Conference, and he participated with different roles in many research project founded by EU, national and/or private organizations. He is member of the IEEE, ACM, and HIPEAC NoE.

C. Antonio Prete is full professor at the Dipartimento di Ingegneria dell'Informazione of the Università di Pisa, where he teaches Embedded Systems and Design of Computer Systems and Applications. He was dean of the MS degree in Computer Engineering at the University of Pisa, and he is chairman of Consorzio Pisa Ricerche and member of the scientific board for Computer Science and Engineering at the IMT Lucca. His research interests include computer architecture and usability of computer systems, and he published more than 70 papers at major international conferences and journals.

Michele Zanda received his "Laurea" degree in Computer Engineering from the University of Pisa, Pisa, Italy, in 2004, and his PhD in Computer Science and Engineering from IMT Advanced Studies in Lucca, Italy, in 2008. Currently he is post-doctoral fellow at the University of Pisa and consultant at Consorzio Pisa Ricerche, in Italy. His research interests include human machine interfaces and the design of embedded systems. Results of his studies were published on conferences and journals.

APPENDIX

Final Questionnaire

1. How often do you use the Internet? (Rarely, Once a month, once a week, Every day)
2. Why do you use the Internet? (Work, Study, Entertainment, Other)
3. Do you usually download software? (Never, At times, Frequently)
4. As a Internet user, have you ever seen the “Security Warning” dialog box? (Yes, No, Don’t remember)
5. How do you usually behave when confronted with a “Security Warning” dialog box? (Click Yes, Click No, It depends)
6. Do you usually pay attention to what is written in the “Security Warning” dialog box? (Yes, No, At times)
7. While deciding whether to accept or refuse a download, the publisher name matters to me. (Agree completely, Agree somewhat, Yes/no, Disagree somewhat, Disagree completely)
8. Do you trust more well-known publishers? (A lot, Not much, Yes/no, Little, Very little)
9. Did you notice publisher names similar to famous ones? (Yes, No)
10. How did you behave when dealing with publishers with a name similar to those of well-known ones? (Always accepted, Always refused, It depends)
11. Do you know what a premium rate connection is? (Yes, No)
12. Is it a risk for you that software download entails a charge? (A lot, Not much, Yes/no, Little, Very little)
13. Is it a risk for you that software usage implies a premium rate connection? (A lot, Not much, Yes/no, Little, Very little)
14. How much are you willing to spend to use a piece of software? (0 - I don’t want to spend, 3€, 10€, 30€ for full download, more than 30€ for full download)
15. Would you feel safer if software which entails a charge were guaranteed by well-known entities? (A lot, Not much, Yes/no, Little, Very little)
16. Before this test, had you already dealt with a Reputation System and feedback systems? (Yes, No)
17. How much were you influenced by the extra information provided by the reputation system? (A lot, Not much, Yes/no, Little, Very little)
18. How much do you trust a publisher with no reputation system? (A lot, Not much, Yes/no, Little, Very little)
19. Do you consider it useful to know the precise number of users who gave feedback comments? (A lot, Not much, Yes/no, Little, Very little)
20. Do you feel safer having a reputation system in the dialog box? (A lot, Not much, Yes/no, Little, Very little)