Designing a Trade-off between Usability and Security: A Metrics Based-Model

Christina Braz[1], Ahmed Seffah[2] and David M'Raihi[3]

[1] Department of Computer Science, University of Quebec at Montreal, 201, President-Kennedy Avenue, room PK-4918, Montreal, QC H2X 3Y7 Canada
braz.christina@courrier.uqam.ca
[2] Department of Computer Science and Software Engineering, Concordia University, 1515, St. Catherine West, Montreal, QC H3G 1M8 Canada
seffah@cs.concordia.ca
[3] Innovation Group, VeriSign Inc.,
685, East Middlefield Road, Mountain View, CA 94043 United States
dmraihi@verisign.com

Abstract. The design of usable yet secure systems raises crucial questions when it comes to balancing properly security and usability. Finding the right tradeoff between these two quality attributes is not an easy endeavor. In this paper, we introduce an original design model based on a novel usability inspection method. This new method, named Security Usability Symmetry (SUS), exploits automata machines theory and introduces the concept of an advanced Multifunction Teller Machine (MTM). We demonstrate, via case study, how to use this model during the design of secure, usable interactive systems.

Keywords: Usability, Security, User-Centered Design, Critical Systems, Automata Machines, Metrics.

1 Introduction

Security has always been an important quality factor in many types of interactive systems including banking software such as Multifunction Teller Machines (MTM). Usability also is required in such systems. However, there is also a common (but false) belief that security is only related to the software systems functionality and that it can be designed independently from usability which only relates to the UI component. In fact, the term UI and the way usability is defined are perhaps major underlying obstacles that explain such erroneous conceptions. Indeed, it gives the impression that the UI is a thin layer sitting on top of the "real" system and that usability can be conceived independently from the other quality factors.

Usability has been defined differently in several standards [1], [2], [3]. Each of these standards emphasizes somewhat different sets of usability factors, such as effectiveness, efficiency, learnability, or user satisfaction. Thus, a more comprehensive model of usability should include both process-related and product-related usability characteristics such as effectiveness, efficiency, satisfaction, security,

and learnability. Moreover, Usability is a generally relative measure of whether a software product enables a particular set of users to achieve specified goals in a specified context of use [4].

On the other hand, "Security Usability" according to [5] deals with how security information should be handled in the UI. Both usability and security can vary depending on the context of use that includes user profiles (i.e., who are the users), task characteristics, hardware (including network equipment), software, and physical or organizational environments [6]. Usability is imperative from the user's perspective (e.g., complete a task correctly without errors that could result in a security problem), from the developer's perspective (e.g., success or breakdown of a system), and from management's perspective (e.g., software with weak security support can be a major constraint to the usability of the system).

A limited amount of work has been done on the usability of security systems and in particular on the intimate relationship that exists between usability and security. The design of usable yet secure systems raises crucial questions concerning how to solve conflicts between security and usability goals. The fundamental question is therefore how to ensure usability without compromising security and vice-versa. Building an acceptable tradeoff is not an easy endeavor. We propose a novel design model named Security Usability Symmetry (SUS) inspection method and the utilization of the Quality in Use Integrated Measurement Model (QUIM) [6] as a model for usability measurement and classification of our advanced MTMs for dealing with this issue in the automata machines domain. We also show, via a case study, how to apply this model during design.

2 Usability and Security: How to Design a Trade-Off?

Some researchers, as well as other standards organizations, have identified other viewpoints on usability, and have included another usability characteristic, *security* (Table 1). Table 1 lists some of the standards where security is included within their usability model. These standards consider that good usability is a significant condition for human security in critical systems, such as medical apparatus or nuclear power stations [4]. Within our model, we adopted this perception of security.

Table 1. Security as a usability characteristic.

Tasks	***Usability***	***Security***
ITSEC: Information Technology *Security* Evaluation Criteria.	IEC 300 [7]	It presents software as *security*-critical.
International Standards Organization (ISO)	ISO 13407 [8]	It describes human-centered design as a multidisciplinary activity incorporating human factors and ergonomic and technical knowledge with the objective of raising efficiency and effectiveness, improving human working conditions, and opposing possible unfavorable effects of use on human health, *security* and performance.
	ISO/IEC 9126 [2]	It defines *security,* which is a sub-characteristic, as a set of software attributes which relates to its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data.
Federal Aviation Administration (FAA) [9]	FAA, 1998	*Security* is a characteristic of the CHI, which is particularly important in an industrial context.

What exactly usability specialists and software designers have to bear in mind when designing usability for secure systems? Figure 1 models the relationship between usability and security. The key characteristics of the usability problem – represented via a usability scenario – related to security are briefly described next:

i) It is important to make sure that the users understand what they should do well enough to avoid making potentially high risk mistakes; this is especially important for security mechanisms, since if a secret is left unprotected, even for a short moment, there is no way to ensure it has not been compromised;

ii) Security is a secondary goal for many users, a necessary step on the way of achieving their primary goals such as checking their account balance or their email; therefore developers should not optimistically assume that users are motivated to read manuals or look for security rules and regulations;

iii) Security concepts might seem self evident to the security developer but are very unintuitive to many users; developers therefore need to put extra effort into understanding the users' mental models and be sure to use concepts the users are familiar with;

iv) Feed-back is often used to prevent users from making mistakes, but this is difficult in security since the state of a security mechanism is often very complex and hard to explain to the user;

v) Security is only as strong as its weakest component. Therefore users need guidance to attend all security aspects in their usage of the mechanism [10].

vi) Attackers can abuse a system that it is "too" usable but not very secure. For example, European Commission (EC) regulations were fundamentally different from usual banking practices. They forbade fees when converting national currencies to euros (fees would otherwise deter users from adopting the euro); this created a unique fraud context where money could be made by taking advantage of the EC's official rounding rules. A method to protect against such attacks is detailed in [11].

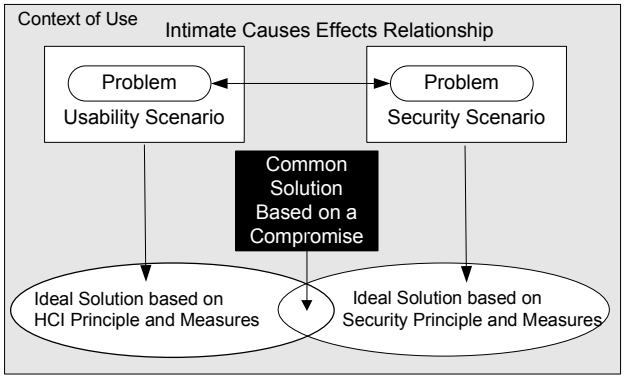

Fig. 1 Usability and Security trade-off: A common solution based on a compromise.

3 Automata Machines as a Tool for Modeling the Relationship

The banking industry pioneered the first wide-scale deployment of self-service automata machines or the Automated Teller Machines (ATMs). A Traditional Automated Teller Machine (ATM) is a device employed by mostly bank users to carry out financial transactions. Usually, a user inserts into the ATM a special plastic card that is encoded with information on a magnetic strip. The magnetic stripe contains an identification code that is transmitted to the bank's central computer by modem. To prevent unauthorized transactions, a personal identification number (PIN) must also be entered by the user using a keypad. The computer then permits the ATM to complete the transaction; most machines can perform limited transactions including for example receiving cash, making deposits, checking account balances, printing checks, printing statements, and other functions: dispense cash, accept deposits, transfer funds, and provide information on account balances. ATMs usually include one or more internal processors which carry out software instructions and enable operation of the machine. Presently most ATM software is proprietary to the particular machine manufacturer. As a result the software which causes one manufacturer's automated teller machine to operate will not operate another manufacturer's automated teller machine. Recently organizations have begun to develop standards related to devices commonly found in ATMs. These standards provide a generally uniform set of instructions for operating each particular type of device which is likely to be found in an ATM (e.g., WOSA-XFS or XFS standard).

As the technology has continued to evolve, a new generation of self service automata has emerged, known as a Multifunction Teller Machines (MTMs), which consists of the best-of-breed of traditional ATMs and the most advanced financial and purchase transactions offered at remote terminals and over the Internet. Where ATMs are really in disadvantage concerning key design factors in relation to MTM's? First, MTMs are conceived to optimize the perceptual and physical needs of the consumer for example features like smaller footprint and graphically enjoyable UIs capitalize on

the appeal and adoption of MTMs. Second, time-saving efficiencies and high tech designs can make MTM's a fashionable option for consumers and deployers as well. And finally, the integration of the self-service with human service if needed in a way that optimizes both transactional efficiencies and the user experience.

The good old ATM cannot defeat the new Multifunction Teller Machine (MTM) in several aspects especially in the "transaction" factor. Being capable to perform up to 150 kinds of transactions ranging from straightforward cash withdrawals and deposits, to fund transfer to trading in stocks to purchasing mutual funds or to cash a check using check imaging to something as ordinary as processing the payment of electricity bills, booking air-tickets, purchasing concert tickets and making hotel reservations. A MTM is in effect the next generation of the old ATM, fully integrated cross-bank MTM network (e.g.: Multibanco in Portugal) providing numerous functionalities which are not straightforwardly associated to the management of one's own bank account, such as loading monetary value into pre-paid cards (e.g., cell phones, tolls, service and shopping payments, etc.). In short, it is really all about survival in the marketplace. Any bank or organization that does not provide intelligent self-service choices to their users will be at a competitive disadvantage in its industry, from a cost structure and a user stickiness and loyalty standpoint [12]. A MTM endows users with the capability to control their end-user experience pretty much in terms of the pace and outcome of this experience. ATMs transactions on the contrary are fundamentally static and linear. MTMs on the other hand dynamically reconfigure transactions "on the fly" to coordinate the actions of users, applications, and devices in real-time.

4 A Case Study: The Credit's Lyonnais French Bank

Within a Credit Lyonnais, French bank branch located at the neighboring of Paris 6 in France, the counter clerk who greeted the local clients have disappeared three years ago. The clerk was replaced by an automaton machine able to carry out 70% of current banking transactions such as money withdrawing, income statements, funds transfers, bill payments, cards orders, profile updating, automaton machine's access with mobile phone, check deposit without envelopes, and other advanced features. From now on, the clients themselves perform their banking operations on the machine.

4.1 Usability and Security Scenarios: What Scenarios Are All About?

Although the HCI literature contains plenty definitions of what a task and usability scenarios are, we would like to refine it and also introduce a new definition for a Security scenario according to below (See also Figure 1):

- **Task Scenario.** A task scenario refers to a description of the task at hand including its context of use. According to [1], the Context of Use (CoU) analysis refers to a broad technique to determine the characteristics of the User, Tasks, and their Environments. The application of the CoU analysis mostly is used as a support to data gather requirements to build the basic components at the early

development stages of the application, and also to establish if the end results which consist of effectiveness, efficiency and satisfaction.

- **Usability Scenario.** A usability scenario details a user problem when doing a task in a certain context. Therefore a usability scenario is a problem related to a task scenario, but it should be well known meaning defined in a usability model, standard or evaluation method.
- **Security Scenario.** A security scenario refers to a description of a task scenario which includes the use of a particular security mechanism. A Security Scenario can be tangible or intangible. A Tangible Security Scenario (TSS) includes physical infrastructure such as controlling user's access to buildings and facilities using Biometrics, or sending a silent alarm in response to a threat at a MTM, etc. An Intangible Security Scenario (ISS) includes data or other digital information, for example, a user who enter sensitive information at registration in order to purchase a concert ticket at a MTM. A Security Scenario might be (or not) a combination of TSS and ISS.

According to the Security Scenario, we classify them as indicated by the overall impact of the security risks of the security mechanisms related to the system's owner such as *High Security Impact, Moderate Security Impact, and Low Security Impact*. A *High Security Impact* refers to the confidentiality, integrity, or availability of the security mechanisms, and it may cause severe or catastrophic loss to the owner's system (e.g., authentication credentials like private cryptographic keys, and hardware tokens); a *Moderate Security Impact* refers to the confidentiality, integrity, or availability of the security mechanisms, and it may cause a moderate loss to the owner's system (e.g., data on internal file shares for internal business use only); and finally a *Low Security Impact refers to the i*mpact on the confidentiality, integrity, or availability of the security mechanisms, and it may not cause any significant financial loss, legal or regulatory problems, operational disruptions, etc. to the owner's system (e.g., public cryptographic keys).

4.2 Tasks, Usability, and Security Scenarios for MTMs

In this section, we introduce a sample of Task, Usability and Security Scenarios related to our case study according to the table 2. A complete and detailed description of the tasks, security and usability scenarios can be viewed at the following Website: http://www.labunix.uqam.ca/~ha991381/TasksScenarios.pdf

Table 2. A sample of one of the Task, Usability and Security Scenarios: Authenticate a user to a Multifunction Teller Machine.

Tasks	*Usability*	*Security*
1. Name: Authenticate yourself to a Multifunction Teller Machine (MTM). ***Scenario:*** User must authenticate her/ himself through a multipurpose contactless smart card token-based authentication (i.e., PIN) in order to have access to different systems. ***Required Features:*** To authenticate yourself to the following systems: • MTM: Card + card slot and card reader + PIN; • Medical Institution: card + card slot and card reader (desktop computer) + PIN; • Facility: bring card close to the card reader (physical access); • Electronic Purse: swipe the card in a card reader + PIN	***Problem:*** *Minimal Action (User Convenience: dealing with multipurpose VS. one purpose smart cards).* ***Scenario:*** The card just mentioned improves user convenience since the user doesn't need to carry several cards and usually memorizing different PIN codes. However, it raises the risk if the card is lost or gets stolen, and also if the card is forgotten by the user in the reader of the MTM. Using a one purpose card is more secure, but means the user will need to carry one card for each application which is not so convenient.	***Problem:*** *Storage of Information.* ***Scenario:*** A multipurpose contactless smart card however puts more sensitive information on the card (i.e., all applications in one card are clearly less secure than one card with one application), and also requires more complex organizational coordination. The risk involved if the wrong person gets access to the card, is much higher. Moreover, contactless smart cards open the door to attacks which exploit over-the-air communication channels in an unsolicited way such as eavesdropping, interruption of operations, covert transactions, and denial of service [13].

4 A Usability and Security Design Model for MTM

We propose the use of the Quality in Use Integrated Measurement (QUIM) [6] as a model for usability measurement and classification of MTMs. QUIM brings the best of breed of existing usability standards and models for evaluating usability in a single consolidated, hierarchical model of usability measurement. QUIM is hierarchical model in that it decomposes usability into *factors*, then into *criteria*, and finally into specific *metrics*. In this sense, QUIM follows the IEEE 1061 (1998) standard (Software Quality Metrics Methodology), which outlines methods for establishing quality requirements as well as identifying, implementing, analyzing, and validating both process and product quality metrics [14], [15]. Also included in QUIM are some emerging usability factors, those identified only quite recently as being important considerations in some contexts. For example, the inclusion of trustfulness and accessibility as usability factors among others.

4.1 Usability Factors

The proposed model includes 9 usability factors among the 10 existing in QUIM briefly described as follows: **Efficiency.** The capability of the software product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use; **Satisfaction.** The subjective response of user while using a software product (i.e., is the user satisfied?); **Productivity.** The level of effectiveness achieved in relation to the resources (i.e. time to complete tasks, user efforts, materials or financial cost of usage) consumed by the users and the system; **Learnability.** The features required for achieving particular goals can be mastered; **Safety.** Whether a software product limits the risk of harm to people or other resources, such as hardware or stored information; **Trustfulness.** The faithfulness a software product offers to its users; **Accessibility.** The capability of a software product to be used by persons with some type of disability (e.g., visual, hearing, psychomotor); **Universality.** Whether a software product accommodates a diversity of users with different cultural backgrounds (e.g., local culture is considered); and finally, **Usefulness.** Whether a software product enables users to solve real problems in an acceptable way.

4.2 Measurable Criteria

Each factor is broken down into measurable criteria (sub-factors). A criterion is directly measurable via at least one specific metric. Definitions of 7 of the 26 existing criteria in QUIM are presented below. These definitions all assume a particular context of use or stated conditions for an application feature. Minimal Action. Capability of the application to help users achieve their tasks in a minimum number of steps; Minimal Memory Load. Whether a user is required to keep minimal amount of information in mind in order to achieve a specified task [6]; Operability. Amount of effort necessary to operate and control an application; Privacy. Whether users' personal information is appropriately protected; Security. Capability of the application to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access [16]; Load Time. Time required for the application to load (i.e., how fast it responds to the user); and finally, Resource Safety. Whether resources (including people) are handled properly without any hazard.

The relations between the 9 usability factors and the 7 usability criteria are described in Table 3. For example, in our MTM application, the Efficiency factor is assumed to correspond to criteria such as Minimal Action, Operability, Privacy, Resource Safety, and Minimal Memory Load. That is, we hypothesize that efficiency can be measured with metrics associated with the criteria listed for this factor in the table. Looking at Table 3 from the perspective of the criteria, we hypothesize that Minimal Action, for example, is related to at six different factors, including Efficiency, Satisfaction, Learnability, and Accessibility.

Table 3. The 9 Usability Factors and Usability Criteria.

Task Number	Task Scenario	Security Problem	Usability Criteria	Usability Factors								
				Efficiency	Satisfaction	Productivity	Learnability	Safety	Trustfulness	Accessibility	Universality	Usefulness
1	Authenticate yourself	Storage of Information	Minimal Action	●	●		●			●		
2	Transfer funds to an intl bank account	Access Control	Operability	●	●				●	●		●
3	Buy a ticket concert	Sensitive Information	Privacy	●	●				●	●		●
			Minimal Action	●	●		●			●		
4	Access your MTM with your cell phone	Credentials across several channels	Security					●	●			●
5	Deposit your check using checking image	Encryption	Loading Time	●		●					●	●
6	Send a silent alarm	User physical safety	Minimal Memory Load	●	●		●			●	●	●
			Resource Safety					●				

4.3 Usability Metrics

A measure or metric is basically a mapping function that assigns a number or symbol to an attribute of some entity [17]. Some metrics are basically functions that are defined in terms of a formula, but others are just simple countable data. Examples of countable metrics include the percentage of a task completed, the ratio of task successes to failures, the frequency of program help usage, the time spent dealing with program errors, and the number of on-screen UI elements. Whereas calculable metrics are the results of mathematical calculations, algorithms, or heuristics based on raw observational data or countable metrics. For example, a formula by [18] for calculating task effectiveness is TE = Quantity × Quality/100 where Quantity is the proportion of the task completed and Quality is the proportion of the goal achieved. The proportions just mentioned are the countable metrics that make up the calculable TE metric. Listed in Table 4 are calculable metrics which may be used in the MTM environment. All these metrics are detailed in QUIM model [6].

Table 4. The proposed calculable metrics for a MTM environment.

Task #	Task Scenario	Usability Criteria	Metrics
1	Authenticate yourself	Minimal Action	Efficiency: Monetary cost of performing the task. Layout Appropriateness LA=100xC_optimal/C_designed[1]
2	Transfer funds to an international bank account	Operability	Layout Appropriateness LA=100xC_optimal/C_designed
3	Buy a concert ticket	Privacy	Data sharing privacy: size of smallest group that share the same identifiable properties, e.g., k-anonymity; Communication Privacy: Probability of identifying correctly the participants of a communication; Pseudonymity: Probability of possibility to link pseudonym with user identity.
		Minimal Action	Same as task 1.
4	Access your MTM with your cell phone	Security	Efficiency: Time to learn; Satisfaction: rating scale for user versus technological control of task.
5	Deposit your check using checking image	Loading Time	Essential Efficiency: EE=100xS_essential/S_enacted
6	Send a silent alarm	Minimal Memory Load	Effectiveness: Workload Task Visibility $TV = 100 \times (1 / S\_total \times \Sigma\ Vi)\ \forall\ i$
		Resource Safety	Effectiveness: Task Concordance TC=100 x D/P Task effectiveness (TE) TE=Quantity x Quality/100

4.4 The Security Usability Symmetry (SUS)

We also propose a new usability and security inspection method called Security Usability Symmetry (SUS), a variant of the *Heuristic Evaluation* method [19]. It aims to help usability specialists and security designers to design/inspect/evaluate an interactive system to identify any usability and security user problems and check for conformance with its corresponding usability criteria and security aspects mentioned previously for Multifunction Teller Machines (MTM). These usability criteria and security aspects can be used to *guide a design decision* or to *assess a design* that has already been created.

According to [19], usability specialists were much better than those without usability expertise at finding usability problems by heuristic evaluation. Moreover,

[1] $C=\sum Pi,j \times Di,j\ \forall\ i \neq j$; Pi,j=Frequency of transition between visual components i and j;

usability specialists with specific expertise (e.g., security) did much better than regular usability specialists without such expertise, especially with regard to certain usability problems that were unique to that kind of interface. Thus, SUS is developed as a security usability inspection method for evaluators who have knowledge of usability AND also computer security. In SUS, a solely usability specialist can also work in pair with a solely security specialist.

The SUS can help also to develop a MTM profile that will impact whether or how usability and security aspects will be implemented in the system. A MTM profile might present the MTM profile that is used by systems designers to determine their specific characteristics and needs.

Prior to evolving into the iterative design phase whereby a product is designed, modified, and tested repeatedly, it is critical that usability specialists and security designers understands its own specific requirements and goals for the MTM. Toward that end, we have provided the SUS that will guide you to the most suitable MTM for your organization. It focuses on the following key areas: Usability and Security requirements, Interoperability, System Application, Technology, and Resources.

The output of the SUS' inspection method. A list of usability and security problems in the interface with references to those usability principles and security aspects that were violated by the design in each case in the opinion of the evaluator is provided as the main output of the method.

Rating severity of the identified usability problems. In SUS, the rating severity is based on three factors: Frequency with which the problem occurs (i.e., is it common or rare?), Consequence of the problem if it occurs (i.e., will it be easy or difficult for the users to overcome?, and finally Persistence of the problem (i.e., is it a one-time problem that users can overcome once they know about it or will users repeatedly be bothered by the problem?).

Rating severity representation. The following Minor to Major rating scale may be used to rate the severity of usability problems: Minor= Minor usability problem; fixing this should be set low priority; Intermediate=Medium usability problem; important to fix as soon as possible. Major=important usability problem; it should be set high priority.

Rating severity of the identified security problems. In SUS, the rating severity is based on six aspects: Authentication (i.e., user identity proofing and verification); Confidentiality (i.e. information is not made available or disclosed to unauthorized individuals, entities or processes); Integrity (i.e., data has not been modified or destroyed in an unauthorised manner); Non-repudiation (i.e., the author of a document cannot later claim not to be the author; the "document" may be an e-mail message, or a credit-card order, or anything that might be sent over a network); Access Control (i.e., granting access to data or performing an action; an authentication method is used to check a user login, then the access control mechanism grants and revokes privileges based on predefined rules); Availability (i.e., a computer system asset must be available to authorized parties when needed).

Rating severity representation. The following Low to High risk level rating scale may be used to rate the severity of security problems: Low=Minor security risk problem; fixing this should be set low priority; Medium=intermediate security problem;

important to fix as soon as possible. High=important security risk problem; it should be set high priority.

The usability problems can greatly be eliminated or reduced through the severity rates where we are able to identify those problems that should be tackled and fixed. The ratings also aid in the allowance of resources for treating the UI problems. According to [Nielsen, 1992], severity consists a combination of three elements: frequency ranges (i.e. from common problems to unusual ones), impact (i.e., establishes the ease or difficulty with which a user gets over a problem), and persistence (i.e., ranges from just one problem that might be surmount to the problem that constantly replicate itself becoming annoying to the user).

A sample of the SUS Review Check-List is presented in Table 4. First, we specify our *Usability Criterion*; second, we define it, and then we present the *Usability Review* questions with their respective *Occurrence*s; afterwards, the *Security Review* with their respective level of security *Risk*. It is important to noting that the dotted squares were used in order to facilitate the information visualization. When the evaluator starts to review the check-list, all cases are filled out. It means that during the evaluation the dotted squares are taken off, and at the end the evaluator is able to well visualize whether the trade-off of usability and security are reached or not. A complete and detailed description of the SUS model can be viewed at: http://www.labunix.uqam.ca/~ha991381/SUSmodel.pdf

Table 4. A sample of the SUS review check-list for MTMs regarding the Usability Criterion: Minimal Memory Load.

Security Usability Symmetry Check-List								
6. Usability Criterion: MINIMAL MEMORY LOAD								
Whether a user is required to keep minimal amount of information in mind in order to achieve a specified task.								
#	Usability Review	Occurrence			Security Review	Occurrence		
		Y	N	NA		Y	N	NA
6.1	Is the memory load on the user minimized (i.e., no memorization of long data lists, complicated procedures, or undertake complex cognitive activities)?				If the task at hand is complex, are the procedures or steps broken down into sub-steps to facilitate securely its understanding and execution?			
6.2	Are the entries short (i.e., short term memory capacity is limited[2], so the shorter the entries, the smaller errors and reading times)?				Does the system provide displayed feedback for all user actions during data entry[3]?			

[2] The capacity of short term memory is normally limited to 7+ 2 items (e.g. letters, digits, words, etc.).

[3] i.e., for reasons of data protection, it may not be desirable to display passwords and other secure entries.

6. Usability Criterion: MINIMAL MEMORY LOAD								
Whether a user is required to keep minimal amount of information in mind in order to achieve a specified task.								
#	Usability Review	Y	N	NA	Security Review	Y	N	NA
6.3	Are short PINs used such as four digits or less (i.e., they are easier to memorize and fast to type)?				Is PIN able to be used in conjunction with a hardware device (2-factor authentication) providing stronger security?			
6.4	Is a *non-user selected* PIN avoided (i.e. more difficult to memorize since it has no meaning and can not be pronounced)?				Can a *non-user-selected* PIN be combined with a variable such as current date and time, at each login to eliminate the risk of replay?			
6.5	Is the MTM's application based on recognition of visual items for authentication (i.e., to avoid unaided recall)?				If recognition of visual items for authentication is used, can the users also associate a phrase to an image to enhance security?			

7 A Concluding Remark

As highlighted in this paper, a very few research has been done on the intimate relationship between usability and security. To be able to build reliable, effective and usable security systems, we need specific guidelines that take into account the specific constraints of usability mechanisms and their potential consequences on security. In this paper, we proposed a design model - a novel usability inspection method - named Security Usability Symmetry (SUS) for dealing with this issue using automata machines more specifically our advanced Multifunction Teller Machine (MTM). We also showed via a case study how to apply this model during design.

References

[1] International Organization for Standardization (1998) ISO 9241-11: "Ergonomic requirements for office work with visual display terminals (VDTs - Part 11: Guidance on Usability".

[2] International Organization for Standardization: ISO/IEC 9126-1:2001 Edition 1; Software product Evaluation – Quality Characteristics and Guidelines for the User, Geneva (2001).

[3] Institute of Electrical and Electronics Engineers (IEEE): 1061-1998 IEEE Standard for a Software Quality Metrics Methodology (1998).

[4] Abran A., Khelifi A., Suryn W., Seffah A.: Usability Meanings and Interpretations in ISO Standards. Software Quality Journal, Kluwer Academic Press. Volume 11, Issue 4 (2003).

[5] Jøsang, A. & Patton, M.: UI Requirements for Authentication of Communication, White Paper, Distributed Systems Technology Centre, QUT, Brisbane, Australia (2001).

[6] Seffah A., Donyaee M., Kline R., Padda H.K.: Usability Metrics: A Roadmap for a Consolidated Model. Journal of Software Quality, Volume 14, Number 2 (2006).

[7] Commission of the European Communities: Information Technology Security Evaluation Criteria (ITSEC), Standard EIC 300 Version 1.2 (1991).

[8] International Organization for Standardization, 1999. ISO 13407: Processes for Interactive Systems, Geneva, Author.

[9] Federal Aviation Administration (FAA): Standard Terminal Automation Replacement System, Human Factors Team Report of the Computer–Human Interface Re-Evaluation (1998).

[10] Cranor, L.F. & Garfinkel, S.: Security and Usability: Designing Secure Systems that People Can Use.O'Reilly (2005).

[11] David M'Raïhi, David Naccache & Michael Tunstall: Asymmetric Currency Rounding, Proceedings of Financial Cryptography 2000, LNCS 1962, Springer Verlag, pp. 192-201, 2001.

[12] NCR Self-Service Universe conference, Washington, D.C. , U.S.A. (2006).

[13] Handschuh, H.: Contactless Technology Security Issues. Security Technologies Department, Gemplus. Information Security Bulletin, Vol. 9, page 95 (2004).

[14] Schneidewind N. F.: Methodology for validating software metrics, IEEE Software Engineering, 18: 410-422 (1992.

[15] Yamada, S., Hong, J.K. & Sugita, S.: Development and Evaluation of Hypermedia for Museum Education: Validation of Metrics. In ACM Transactions on Computer-Human Interaction (TOCHI), 2 (4) p. 284-307 (1995).

[16] International Organization for Standardization/International Electro echnical Commission: ISO/IEC 12207, Information Technology, Software Life Cycle Processes Geneva (1995) Author.

[17] Fenton, N.E. & Pfleeger, L.: Software metrics, 2nd ed., International Thompson Publishing Company (1997).

[18] Bevan, N.: Measuring usability as quality of use, Software Quality Journal, 4, 115 130 (1995).

[19] Nielsen, J.: Finding Usability Problems through Heuristic Evaluation. In the Proceedings of ACM Computer Human Interaction 1992 (CHI'92) Monterey, CA, (US) 3–7 May 1992).