Security Analyzing and Designing GUI with the Resources Model

Maryam Mehrnejad
Information and Communication Security Lab
Computer Department, Ferdowsi University
Mashhad, Iran
maryam.mehrnejad@gmail.com

Ehsan Toreini
Computer Department, Engineering Faculty
Mashhad Branch, Islamic Azad University
Mashhad, Iran
toreini@gmail.com

Abbas Ghaemi Bafghi
Information and Communication Security Lab
Computer Department, Ferdowsi University
Mashhad, Iran
ghaemib@ferdowsi.um.ac.ir

***Abstract*--Recently security problems in the Graphic User Interface (GUI) of applications have become a serious threat for system security. Because much of security experts don't design the GUI from end user's point of view, users have problems to practice security. The aim of Human & Computer Interaction (HCI) and Security (HCI-Sec) is to improve the usability of security features in end user applications. In this paper we apply the resources model (a model in HCI) to analyzing and designing system GUI with a security perspective to achieve a more secure and usable system. We studied Tests part of E-learning system in Ferdowsi University of Mashhad (FUM) as our case study. And we exploited faults that slow down user co-ordination with the system and used this model to explore design alternative. We generally analyzed GUI and proposed an alternative GUI in order to solve interaction problems. Finally we analyzed the GUI with a security perspective to improve the usability of security issues in this system. The results show this model works very well in the field of security.**

***Keywords*- Externalisation, HCI-sec, Interaction Strategy, GUI, Resources Model, Security Goals.**

I. Introduction

Recently security problems in applications' GUI have become a serious threat for system security. Much of security research and practice assumes that the people using systems or tools are well acquainted with security principles and practice, and think like programmers and security practitioners. This is not often (in fact, very rarely) the case. And the average computer user simply will not (or does not know how to) practice good security. Unfortunately even if a project's HCI expert in project team notices the security issues in GUI, due to few researches in HCI and security, he/she can't embed the security features in GUI. Also as mentioned in [2]: "Younger Users (who had grown up with computers) perceived security as an obstacle they had to work around". Authors in [2] gather many works about HCI-sec. In one of the famous ones, authors claim that: "Hackers pay more attention to the human link in the security chain than security designers do" [11]. So figuring out how to make security simple would help to alleviate this problem, and applying HCI to security is an important piece of this puzzle. HCI-Sec or HCI-S is the study of interaction between human and computer, or HCI, specifically as it pertains to information security. Its aim is to improve the usability of security features in end user applications. HCI-S has being introduced in [1]: "The part of a user interface which is responsible for establishing the common ground between a user and the security features of a system. HCI-S is human and computer interaction applied in the area of computer security". HCI-S is a critical area for developing and using trustworthy systems.

Authors in [3] discuss an interesting fact: "Information assurance (IA) specialists concur that security depends on people more than on technology. Another commonplace is that employees are a far greater threat to information assurance than outsiders". And it is more highlighted in Computer Supported Cooperative Work (CSCW) [4]. Also that part of the application which is responsible for interaction with end user is called Graphic User Interface (GUI). So if we want to improve security for users, we should improve the security of GUI. Fortunately deposit of HCI-Sec, HCI has an old history and there are a lot of researches in this area. Connecting these two fields (security and HCI), improves the security in GUI level of systems as a new and strong way. Applying HCI theories into analyzing and designing GUI leads to more usable GUI. And applying these theories with a security perspective produces more secure and usable systems.

The Resources Model has been introduced in [5] as a framework to analyze and design GUI. This model is based on Distributed Cognition (DC) [6] and as they mentioned in their abstract, they "hope to provide the foundation for a program of research that extends the DC analysis of single user systems to larger units of analysis more familiar to CSCW and DC research". In our paper, we use this model to analyze GUI of a single user system (Tests part of e-learning system of FUM), exploit faults and explore design alternative. More, we extend this model on security features of the system. This is the first time that the resources model is applied in security. The results are very interesting and encourage researchers to use this model or other HCI models in security.

Section 2 introduces the resources model briefly. Section 3 discusses security goals in systems. Section 4 includes our approach; the resources model and security. In section 5 we apply the model on our case study as empirical studies. This section consists of 3 major parts: Scenarios, Evaluating GUIs and Exploring design alternative as new GUIs. Section 6 is conclusion[i].

II.RESOURCES MODEL [5]

This section is entirely taken from reference [5]. Since this is the first time this model is applied to security, there is no extended one on this model that is customized for security. So this section is a summarized version of reference [5].

The resources model takes seriously idea introduced to HCI by Suchman [7], that various kinds of information can serve as resources for action and it defines a set of abstract information structures which can be distributed between people and technological artifacts. These information structures can be combined to inform action, and in taking an action new resources are configured. The resources model also introduces the concept of interaction strategy and describes the way in which different interaction strategies exploit different information structures as resources for action.

A. *Information Structures*

The resources model distinguishes between abstract information structures on the one hand and their representation (or implementation) in an interaction on the other hand. The abstract level of analysis allows us to consider the structural characteristics of the information that are essential for them to serve as resources of various kinds. It also allows us to consider certain kinds of equivalences between different representations of information. On the other hand, at the representational level, we are concerned with the detail of how an information structure is distributed among people and the artifacts and what form and content the external aspects of the representation take.

1) Abstract Information Structures

The resources model identifies six information structures that can be defined at an abstract level. First we would describe the abstract information structures which constitute the building blocks of the resources model. Secondly, we will give examples of the different ways these abstract information structures can be represented for being applied as resources. The abstract information structures are as follows: **Plans, Goals, Affordances, History, Action-effect relations, States**. Note that it is entirely possible that more structures could be required.

2) Representing Abstract Information Structures

Before information can be used as a resource for action it has to be represented during the interaction. A given information structure may be represented externally in the interface, internally in the head of the user or more often, distributed across the two. For example **Plans** can be represented internally as memorized procedures to complete some task. They can also be represented externally as standard operating, or step by step instructions for achieving the goal. Or as an example of a **state** that is externalized, consider the Towers of Hanoi problem. The current state is represented as the positions of the disks on the pegs.

B. *Interaction Strategies*

The second part of the processing model is concerned with how resources can be used to inform action. A configuration of resources can be used in different ways. People interact with the same graphical user interface in various ways. We use the term *interaction strategy* to describe different ways in which resources can be used to make decisions about action. As an example in the case of HCI a similar distinction can be made between command line interfaces which require the user to recall appropriate commands and menu-based interfaces which support a range of interaction strategies including eliminating implausible options. Interaction strategies presuppose certain configurations of resources to make them effective and conversely a particular configuration of resources makes particular interaction strategies possible. Below we count a range of interaction strategies relevant to HCI and relate these to the information resources they require. These strategies are: **Plan Following, Plan Construction, Goal Matching and History-based Selection and Elimination**. Like abstract information structures, we do not suppose that this is an exhaustive list.

We just explain 2 Plan Following and History-based Selection and Elimination strategies that we will use in Empirical Study section of this paper. You could refer to the main reference [5] for more detailed information.

1) Plan Following

As an interaction strategy, plan following involves the user in co-ordinating a pre-computed plan with the history of action so far undertaken. In its simplest form the plan is followed by determining the next action on the list until the list is exhausted. Some plans may have conditional steps requiring the plan follower to examine the current state of the system. Plans are often followed for a particular goal but it is possible to follow a plan blindly without any knowledge of what it will accomplish. A pre-computed plan is central to the plan following strategy. Thus a plan-following interaction will have the plan as a resource either externalised in the interface, recalled by the user or recorded in some other form (as an operating procedure kept in a manual for example). In a plan following interaction, the plan and interaction history need to be maintained and co-ordination will combine them in order to keep a sense of position within the plan. Aspects of state and goal may also be co-ordinated with this position to deal with conditionals, to assess whether the goal is achieved or not.

2) History-based Selection and Elimination

HCI has paid little attention to the role of history in decisions about action, but a possible strategy for choosing among affordances is to eliminate those that have already been chosen. Alternatively, history could be used to repeat an action that had previously been taken. Interfaces that support these strategies might have some inspectable representation of history such as the go function available in many web-browsers.

The interaction strategy a user adopts will, in part, be shaped by the resources that are available to her. Table 1 summarizes the interaction strategies given above and relates these to the abstract resources that are required for their use.

TABLE1. STRATEGIES AND THE RESOURCES THEY REQUIRE.

Strategy	**Resources required**
Plan following	Plan, history, state
Plan construction	Goal, affordances, action-effects, state
Goal matching	Goal, affordances, action-effects
History-based choice	Goal, affordances, history

C. *Using the Resources Model to Analyze Interaction*

There are 3 main ways in which the resources model has proved useful in framing an analysis of interaction in terms of distributed cognition. First, as a means of comparing different interface designs. Second, as a means of analyzing interaction scenarios. Third, we demonstrate its use as a way of generating design alternatives and analyzing their effects on user performance. We will use these in our case study in section 5.

III. SECURITY GOALS

As we mentioned before, Information assurance specialists concur that security depends on human more than on technology. So the problem that how end users use the application is important in security. One source of security problems is not considering the security requirements of the complete system. As authors in [9] discussed "another source is not considering security in the application itself". More, if we present security in GUI badly, users can't use it, don't use it, ignore it, or at least perceive security as an obstacle they had to work around.

Security requirements according to [9], means "requirements that if respected, lead to a system's security goals being satisfied". Also in this paper it is mentioned that: "Security requirements have traditionally been considered to be 'non-functional' or 'quality' requirements. Like other quality requirements (e.g., performance, usability, cost to run) and they do not have simple yes/no satisfaction criteria. Instead, one must somehow determine whether a quality requirement has been satisfied (satisfied well enough)". In another part of their paper they claim: "Security requirements that express what is to happen in a given situation, as opposed to what is not ever to happen in any situation, would facilitate their analysis. Such requirements would have binary satisfaction criteria, either behaving appropriately in the given situation or not, and one can have test criteria to determine what 'appropriately' means. The cost of ensuring behavior in a given situation is easier to measure than the cost of ensuring something never happens, facilitating cost/benefit analysis".

But recently because of the importance of security, it is considered as a functional requirement. As we mentioned, Security requirements means requirements that if respected, lead to a system's security goals being satisfied. There are many various definitions for security goals. But in a general term we can say: **Those goals which protect the assets of an organization**. Authors in [9] have their approach for definition of security goals: "One set of security goals is determined by listing these threats on assets, then preventing (or avoiding) the action(s) on the asset(s) that realizes the threat", and so on .What we have proposed is considering key concepts of security in a system as security goals. There are 7 security concepts with some solutions in implementation level [10], which are presented in table 2.

TABLE2. SECURITY CONCEPTS AND IMPLEMENTATIONS IN [10]

Security concept	**Solutions in implementation**
Authentication	Passwords, Tokens, Biometrics
Authorization	Access Control Lists
Confidentiality	Cryptographظy, Steganography, Access Controls, Database Views
Data/message integrity	Hashing (MD5, SHA-1, …), Checksums (CRC…), Message Authentication Codes (MACs)
Accountability	Logging & Audit Trails
Availability	Add redundancy to remove single point of failure and Impose "limits" that legitimate users can use
Non-repudiation	Generate evidence / receipts (digitally signed statements).

These 7 security concepts can be externalized in GUI in different ways like using user/password systems GUIs or using colors as metaphors and etc.

IV. THE RESOURCES MODEL IN SECURITY: A NEW SOLUTION FOR DESIGNING SECURE GUI

From two last sections we know that the resources model as an approach in HCI can help designers to design more usable artifacts. Also there are 7 security concepts and considering them in a system, warranties system security. And we should embed security in GUI in some way that users consider it, or in other words, we should make security usable. The question is how we can do that?

It is the first time that the resources model is applied to security. Then we should somehow combine the resources model and security. Maybe it is a good idea to consider security as another abstract information structure for this model. And we know that it is possible to add more structures if it is required. This idea is totally raw and we don't have any opinion that could it work or not! Another approach, that we extended it here, is that considering security as constraint on functional requirements of a system. The idea we apply here is from [9].

In [9] authors describe a security requirement engineering framework which facilitates production security requirements. They explained that in this framework, "Primary security goals are operationalized into primary security requirements, which take the form of constraints on the functional requirements sufficient to protect the assets from identified harms". What we apply here, is different from [9].

We will define some scenarios and use them for this work. What is common in this section of our work and [9] is that we have a general scenario as [9]'s functional requirement and the security parts of it are considered as the constraints defined in [9]. Also we will define some short security scenarios as [9]'s primary and secondary requirements and other quality constraints which come from primary and secondary goals and other quality constraints directly. Figure 1 shows it more properly. For more information about this picture you can refer to [9].

The steps of our approach are described here:

1. Describe the system security goals according above 7 concepts in the field of security.
2. Describe the system functional requirements.
3. Exploit the security requirements based on security goals and functional requirements.
4. Composite some general scenarios; for each functional requirement, one scenario.
5. Specify the security parts of the general scenarios.
6. Composite enough scenarios for other security requirements.
7. Ask enough users to follow the scenarios.
8. Study users and log their interaction with the system.
9. Use the resources model to analyze these studies.

The steps are described in Empirical Study section in detail.

V. Empirical Study

E-learning system is one of the most important systems in educational environments. Today many organizations and universities present online services and in order to train their staff, they use e-learning systems. Our case study is designing new Test and Survey feature in e-learning system of Ferdowsi University of Mashhad (FUM)[1] that we just used the Test part of it. E-learning system of FUM is used for many years by students and teachers. It has several features and parts and one of them is Tests and Surveys part. By this feature a teacher can design, edit, remove, and correct a new test and also a lot of other tasks. The details of system capabilities will be discussed later.

This section describes the 9 steps of our approach. First we describe the 3 first steps and we continue the remained steps in 3 major parts: Scenarios (steps 4 to 6); General Scenario and Security Scenarios, Evaluating GUI (step 9); General fault designs and Security fault designs, and Exploring Design Alternatives (step 9).

For this system, the 3 first steps of our approach are:

1. **System security goals**: We need all 7 security concepts in nearly all application. The importance of each one differs in different application. For example the Availability concept is more highlighted in web applications. Since In this application we face different groups of students and accessibilities, the authorization is more considerable. Other concepts are all needed and we consider all 7 concepts as our security goals.
2. **System functional requirements**: These are some functional requirements of the Test part of E-learning system for a teacher; designing a new test, correcting test, calculate test statistics, informing marks to students, deleting the test and etc.
3. **Security requirements**: Each of above functional requirements causes many security requirements. Below we just describe the ones related to designing new test;
   a) Test can be private for teacher and students of that course or it can be public.
   b) Teacher can categorize students to take different tests.
   c) Teacher can define a valid date and time and a certain duration time.
   d) Students can take an exam anonymously.
   e) Teacher could specify a valid IP range for test. Students have to take the test just in that range.
   f) System should lock the content of the test during the test time.
   g) System should provide a back upping feature for current test.
   h) Also teacher and students can trace the steps of the exams like dates and time, answers and etc.

Steps 4, 5 and 6 are described in section 5.A and step 9 is described in section 5.B and 5.C. Step 7 and 8 are the human test parts.

A. *Scenarios*

We designed some scenarios and selected two software engineering graduated students as teacher and asked them to design a new test. They hadn't worked with the system before and were appropriate candidates. We asked them to follow the general scenario and security scenarios separately. In this study we used **thinking out loud** and in some cases **pair working** (user and interviewer) techniques [8]. We asked users to ask any ambiguity or question about the system. Audio files have been recorded for analysis. The average interview time is 1hour and 50 minutes per case.

[1] https://vu.um.ac.ir

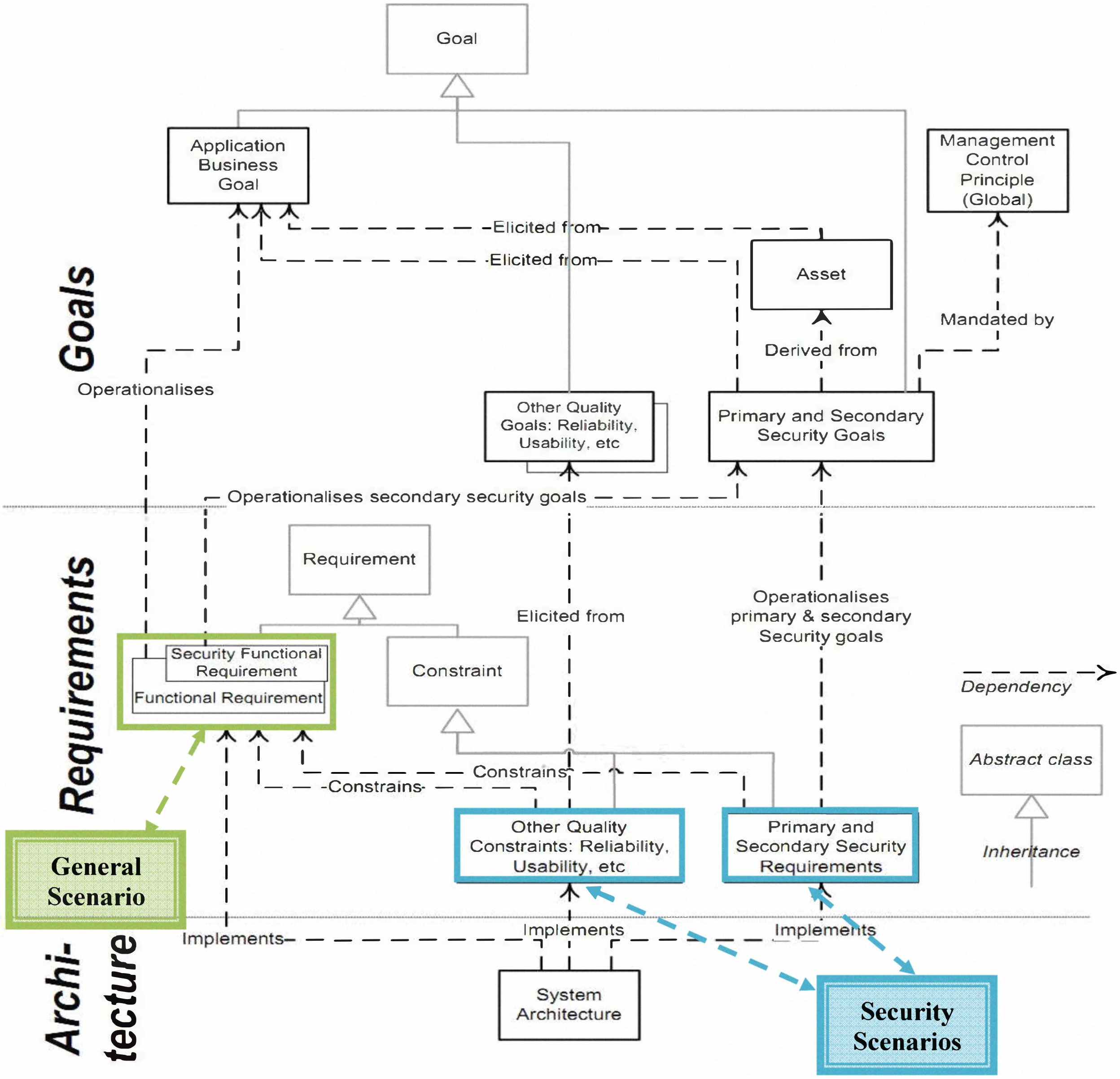

Goal
Application Business Goal
Management Control Principle (Global)
Elicited from
Elicited from
Asset
Mandated by
Derived from
Goals
Operationalises
Other Quality Goals: Reliability, Usability, etc
Primary and Secondary Security Goals
Operationalises secondary security goals
Requirement
Elicited from
Operationalises primary & secondary Security goals
Requirements
Security Functional Requirement
Functional Requirement
Constraint
Dependency
Abstract class
Constrains
Constrains
Other Quality Constraints: Reliability, Usability, etc
Primary and Secondary Security Requirements
General Scenario
Inheritance
Implements
Implements
Implements
Archi-tecture
System Architecture
Security Scenarios

Figure 1: Class diagram of security requirements core artifacts in [9]

1) *General Scenario*

Users were asked to design a new test as a general scenario. E-learning system of FUM has provided below steps for this:

a) Logging Tests and Surveys part

b) Creating new test

c) Defining questions in repository

d) Adding questions to new test

Note that steps b and c could be exchanged. In fact, defining questions in repository could be performed in any other time separately from the task of designing new test.

2) *Security Scenarios*

For this case study we defined the following tasks as security scenarios. Each one cover one or more security concepts that written in front of them:

a) Logging in e-learning system- authentication

b) Defining test valid dates and times- authorization

c) Defining test duration time: authorization

d) Defining valid IP range- confidentiality and availability

e) Back upping- data integrity

f) Not permitting to use course content in the website during the test- authorization and confidentiality

g) Categorizing students to take the test- authorization

h) Anonymous test (taking by students and correcting by teacher- authorization and confidentiality

i) Traceability of the test by students and teacher- accountability and non-repudiation

The general scenario covers some of the above tasks (a, b, c, d and e) and others (f, g, h and i) were followed by users separately. Again we asked users to do these security tasks and we recorded audio files for analysis. The average time of every interview was 1 hour and 30 minutes.

B. *Evaluating GUI*

We evaluate the GUI in 2 main parts; General Fault Designs and Security Fault Designs.

1) General Fault Designs

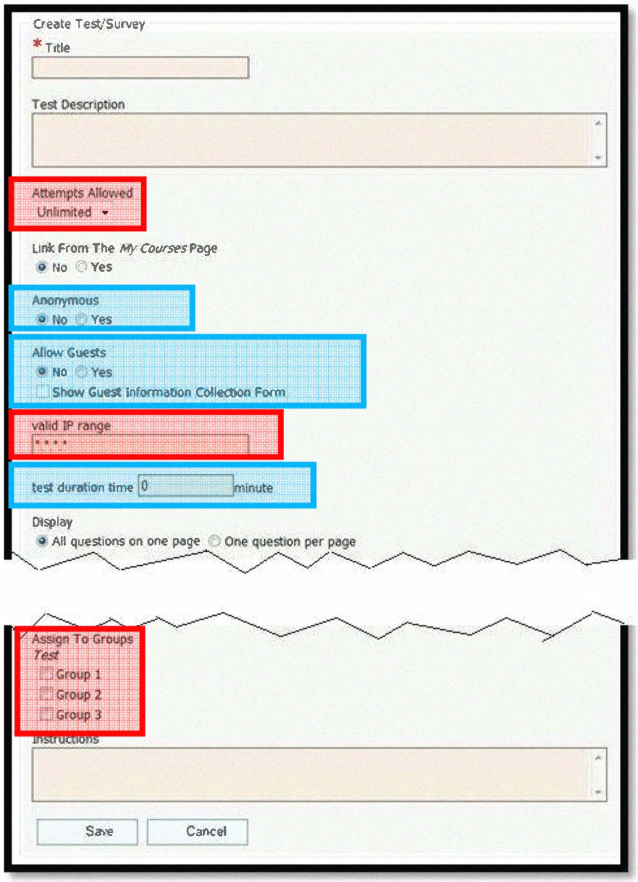
We studied all forms and pages for the scenario designing a new test. From the resources model point of view, there were a lot of faults in the current design that slow down user co-ordination with the system. Here, as an instance, we analysis the form create test (figure 2). This form is for general information of a new test. The blue parts show good externalization of the resources in this from and the red parts show the bad ones. For importing information in this section, the key resource is plan that is externalized in a form and the dominate strategy is goal matching.

Let us analyze one part of this form in detail. Notice the valid IP range. For example if we want to let students to take the test only in FUM campus, the valid IP range is from 192.168.0.1 to 192.168.254.254 and the text box should be filled with 192.168.0.0. As you see, there is no externalization for plan to compute this. But goal resource is externalized well as *.*.*.* in textbox. This supports a plan following strategy. User co-ordinate with the plan is internally in his/her mind and goal which is externally in the form. But finally our users couldn't complete the plan. A recommendation is to externalize the plan of computing IP valid range as an example in the form. Another design alternative in this case is to consider two IP valid range (from *.*.*.* to *.*.*.* that is according to user's mental model) and we explore this in our design alternative.

The biggest problem users faced in empirical studies was related to designing and adding questions to a new test. Our users were in a loop and follow wrong scenarios frequently and finally they added some questions to the test only with the help of interviewer. From the resources model point of view and with considering empirical studies, the appropriate strategy for hole task of designing a new test is plan following. As we mentioned in section the resources model, all resources which are needed for this strategy (plan, history and state) should be externalized in forms. We will discuss this later.

2) Security Fault Designs

Again we asked users to follow all security scenarios and we studied all forms related to these scenarios. Here we just focus of one the biggest problem users faced in these scenarios; Categorizing students to take the test- authorization (part g).



Create Test/Survey

* Title

Test Description

Attempts Allowed
Unlimited ▾

Link From The *My Courses* Page
● No ○ Yes

Anonymous
● No ○ Yes

Allow Guests
● No ○ Yes
□ Show Guest information Collection Form

valid IP range
..*.*

test duration time 0 minute

Display
● All questions on one page ○ One question per page

Assign To Groups
Test
□ Group 1
□ Group 2
□ Group 3

Instructions

Figure 2: Some parts of creating new test form- current GUI

However it is externalized as the history resource in the form (figure 2, *Assign to Groups*), but users didn't notice to it at all. As you see in figure 2, all security tasks in this form are in red. It is one of the more interesting parts of our study; however all security issues has been considered in the current system, but the users didn't notice them. So what is the problem? The **primary goal** for teacher is **designing new test.** And **considering security issues** is a **secondary goal**. It is the reason that users do not notice it. From HCI point of view the reason that the main task is more highlight is that the user interface is not user-friendly enough and the user can't interact with it easily. For example imagine a person who wants to make a call with her friend's cell phone. If the cell phone had a simple dialing system, the user focuses on her conversation. But if it had a complicated one, she focuses on how to dial. As a solution, we propose to separate the security parts of GUI that we will explain later.

C. *Exploring Design Alternative*

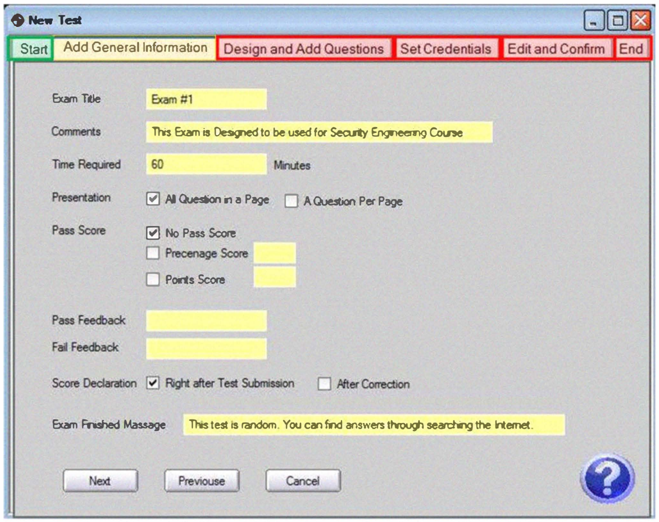
In order to use the resources model in design, we studied the user's interactions in empirical studies, to exploit the strategy used in scenarios. The main problem users faced, was the process of adding questions to the test and assigning proper accesses to students groups. In fact, our users couldn't allow just a certain group(s) of students to take the test.

Users, according to their mental model, didn't consider designing new test and designing new question as two separate parts. So they can't co-ordinate with this design. Also they consider security issues as secondary goals and ignore them in the main task. For designing new test, the key strategy user uses is plan following. Also he can use history elimination strategy and performs his task in a shorter time. The resources needed for these strategies are plan, history and state for Plan following and goal, affordance and history for History elimination.

For these two strategies, we designed the steps of task designing new test as tabs in the top of the GUI. After completing each step, the state of that step changes its color from red to green. And for solving the security problems, we separate the security parts of GUI. So the steps of task designing new test are:

- Logging Tests and Surveys part
- Creating new test
- Designing and adding questions
- Defining access authorities

Figures 3, 4 and 5 show the proposed design alternative. For designing this GUI, we used GUI Design Studio. GUI Design Studio is a common graphical user interface design tool for Microsoft Windows that you can use to rapidly create demonstration prototypes without any coding or scripting [12].



Exam Title: Exam #1

Comments: This Exam is Designed to be used for Security Engineering Course

Time Required: 60 Minutes

Presentation: ☑ All Question in a Page ☐ A Question Per Page

Pass Score: ☑ No Pass Score ☐ Precenage Score ☐ Points Score

Pass Feedback:

Fail Feedback:

Score Declaration: ☑ Right after Test Submission ☐ After Correction

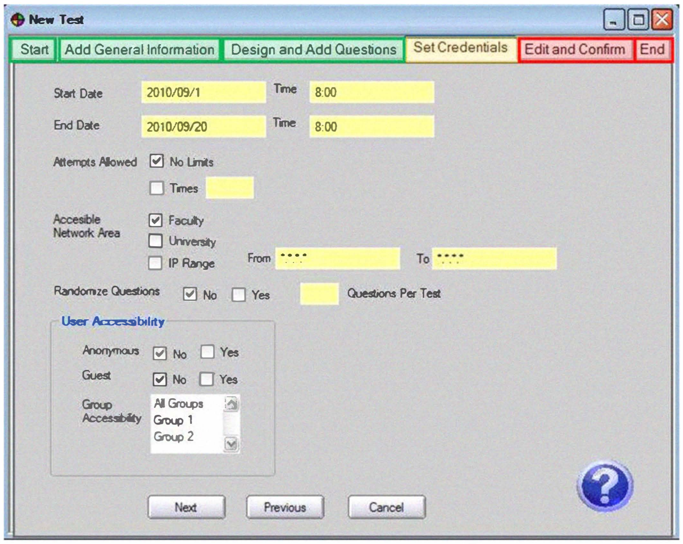
Exam Finished Massage: This test is random. You can find answers through searching the Internet.

Figure 3: Proposed design alternative

Resources of plan following and history elimination strategies have been provided in proposed design:

- **Plan**: All plans are externalized as forms in each step.
- **History**: One of the most important parts of designing new test for a user is to remember where is she/he in the task. History structure is externalized as tabs in top of the windows and we used colors red and green as metaphors which are matched with user's mental models of colors; red for uncompleted steps and green for completed steps. Another instance of history externalization is in the tab Design and Add Questions, where the previous added questions for new test are showed.
- **State**: State resources are generally presented as a final step; Edit and Confirm. Also the user can follow the state of the task by the color of tabs.
- **Goal**: The externalization of goal is in the form of changing the color of the current step in yellow. So in every stage, user can realize and distinguish the current goal from previous ones.
- **Affordance**: This resource is externalized as different links and buttons in every step. For example, in the tab Design and Add Questions for test, there are different buttons; Choose for designing new question, Insert Selected Questions for using last designed questions. Other buttons are: Cancel, Exam Preview, Previous and Next (figure 5).

Also the tab Set Credentials makes user notice security parts definitely. Figure 4 shows this tab.



Start Date: 2010/09/1 Time: 8:00

End Date: 2010/09/20 Time: 8:00

Attempts Allowed: ☑ No Limits ☐ Times

Accesible Network Area: ☑ Faculty ☐ University ☐ IP Range From: . . . . To: . . . .

Randomize Questions: ☑ No ☐ Yes Questions Per Test

Anonymous: ☑ No ☐ Yes

Guest: ☑ No ☐ Yes

Group Accessibility: All Groups, Group 1, Group 2

Figure 4: Security part of proposed GUI

VI. CONCLUSION

Applying HCI theories in the context of security of systems is a new solution to solve security problems in user side. In this paper the resources model is applied to the security features of the GUI for the first time. And Tests and Surveys part of e-learning system of FUM was the case study.

Users followed some scenarios related to security issues in the system. Their interaction with the system has been studied and design faults have been exploited. And finally design alternative for the system has been explored. We simulated proposed GUI with GUI Design Studio. In the new GUI we provided all resources needed for two strategies plan following and history elimination that were used by users in the case study. The results shows the resources model works well in security area and can force users to notice the security issues in GUI.

As future works, we would like to design a framework to compare the design alternative with the old one. Because the source code of applications is not accessible generally, implementing proposed GUI and then comparing is not sensible. And it is more rational first we prove the efficiency of the new GUI and then implement it. So we should evaluate it in some other way. One way is to draw the user navigation graph in each system and compute the length of paths for doing different tasks. The shorter one is the better one. It could be implemented as a framework for comparing the design alternative with the current GUI. Also applying other HCI theories in the field of security would be helpful for designing more secure systems.

ACKNOWLEDGMENT

The authors would like to thank Dr. Mohsen Kahani from *Ferdowsi University of Mashhad,* for his great support to let us the needed credentials to use the *Virtual E-Learning System of Ferdowsi University.* Also we thank Prof. Matt Bishop from *University of California at Davis,* for his comments about the importance of working in the field of HCI-Sec and his help towards our better understanding of HCI-Sec concepts. We feel lucky and delighted to have the opportunity to speak to Prof. Bishop and thank him for his friendly encouragement.

REFERENCES

[1] J. Johnston, J. Eloff, and L. Labuschagne, "Security and human computer interfaces,"Computers & Security, Vol. 22, No. 8, pp: 675–684, 2003.

[2] S. Smith, "Humans in the Loop: Human-Computer Interaction and Security", IEEE Security and Privacy, Pages: 75-79. IEEE. May 2003.

[3] M. Kabay, "Using Social Psychology to Implement Security Plicies", Computer Security Handbook, Chapter 35, 4th edition, John Wiley & Sons Prees, 2002.

[4] M. S. Ackerman ,"The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility", Human-Computer Interaction, Volume 15, pages 179 – 203, September 2000.

[5] P. Wright, B. Fields, and M. Harrison, "Analyzing human-computer interaction as distributed cognition: the resources model. Human-Computer Interaction", ACM Digital Library, Journal Human and Comouter Interaction, Valume 15, Issue 1, 2000.

[6] Y., Rogers, "Distributed Cognition and Communication", In The Encyclopedia of Language and Linguistics 2nd Edition. Edited by Keith Brown Elsevier: Oxford. 181-202, 2006.

[7] L.A., Suchman, "Plans and situated actions: The problem of human computer interaction", Cambridge University Press, 1987.

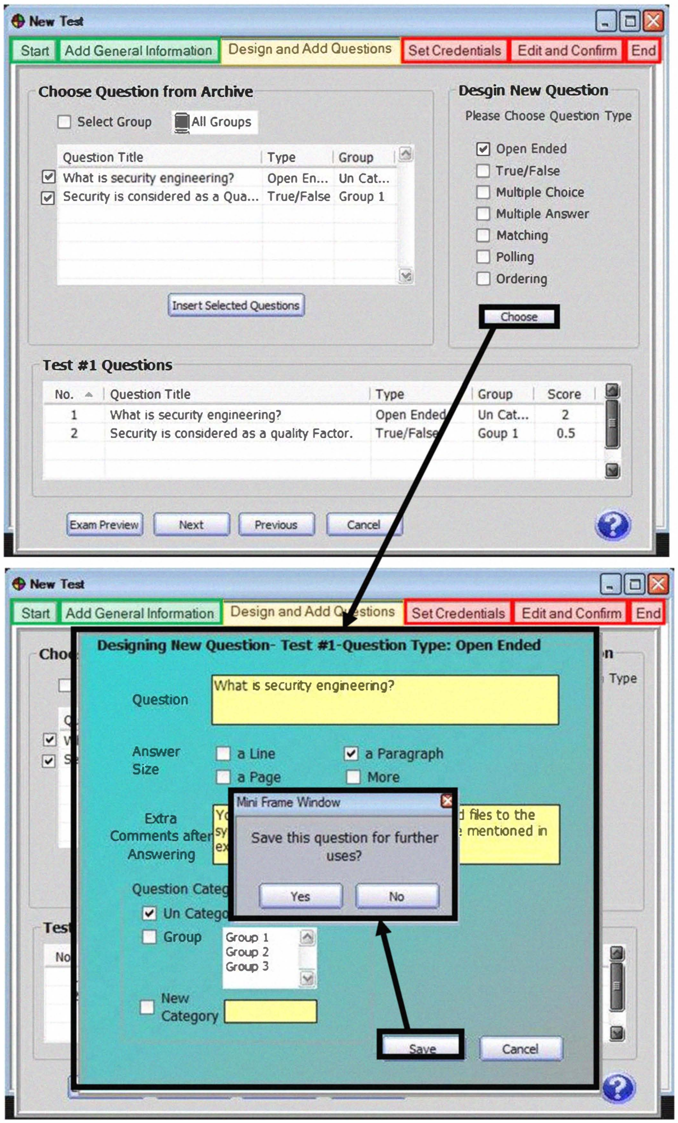
[8] G. Hachman, T, Ferratt, F, Kerckaert, "An experiential approach to teaching students about usability and HCI", SIGCHI Bullettin, ACM, Valume 26, Number1, pp: 56-59, January 1994.

[9] C. Haley, L. Robin, J. Moffett and B. Nusibeh, "Security Requirements Engineering: A Framework for Representing and analysis", IEEE transaction on software engineering, Volume 34, Number 1, 2008.

[10] N. Daswani, C. Kern, and A. Kesavan, "Foundations of Security: What Every Programmer Needs To Know", ACM Digital Library, Apress Berkely, CA, USA, 2007.

[11] A. Adams and M.A. Sasse, "Users are Not the Enemy: Why Users Compromise Security Mechanisms and How to Take Remedial Measures", Comm. ACM, vol. 42, no.12, pp. 41–46, 1999.

[12] GUI Design Studio Help, Available on: http://www.carettasoftware.com/guidesignstudio/



	Question Title	Type	Group
☑	What is security engineering?	Open En...	Un Cat...
☑	Security is considered as a Qua...	True/False	Group 1

No.	Question Title	Type	Group	Score
1	What is security engineering?	Open Ended	Un Cat...	2
2	Security is considered as a quality Factor.	True/False	Goup 1	0.5

Question: What is security engineering?

Save this question for further uses?

Figure 5: Designing and adding questions for new test

[i] Please note that the figures in this paper are colorful and the colors are used as metaphors. So please read the soft version of this paper.