Security Analyzing and Designing GUI with the Resources Model

Maryam Mehrnejad (maryam.mehrnejad@gmail.com)
Ehsan Toreini (toreini@gmail.com)
Abbas Ghaemi Bafghi (ghaemib@ferdowsi.um.ac.ir)

Ferdowsi Universirty of Mashhad

ISCISC’11
September 2011, Mashhad, Iran

Agenda

- Introduction
- Resources Model
- Security goals
- The Resources Model in Security
- Empirical Study
- Future Works
- References

Agenda

- Introduction
- Resources Model
- Security goals
- The Resources Model in Security
- Empirical Study
- Future Works
- References

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.
  - 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.

Introduction (Cont.)

- The Resources Model has been introduced as a framework to analyze and design GUI.
- 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.

Agenda

- Introduction
- Resources Model
- Security goals
- The Resources Model in Security
- Empirical Study
- Future Works
- References

Resource Model

- The Resources Model has been introduced as a framework to analyze and design GUI.
- various kinds of information can serve as resources for action
  - it defines a set of abstract information structures which can be distributed between people and technological artifacts.
- The resources model introduces the concept of interaction strategy
  - describes the way in which different interaction strategies exploit different information structures as resources for action.

Resource Model (Cont.)

- The resources model distinguishes two things in an interaction:
  - *Information Structures*
  - *Interaction Strategies*

Abstract Information Structures

- The resources model identifies six information structures that can be defined at an abstract level
- Before information can be used as a resource for action it has to be represented during the interaction.

Abstract Information Structures

- 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.

Interaction Strategies

- is concerned with how resources can be used to inform action
- 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.

Interaction Strategies (Cont.)

- These strategies are:
  - **Plan Following,**
  - **Plan Construction,**
  - **Goal Matching,**
  - **History-based Selection and Elimination**.
- we do not suppose that this is an exhaustive list.
- **Plan Following** and **History-based** Selection and Elimination strategies are used in our Empirical Study.

Plan Following

- Plan following involves the user in coordinating 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.
- A pre-computed plan is central to the plan following strategy.

History-based Selection and Elimination

- A Strategy for choosing among affordances is to eliminate those that have already been chosen.
- Interfaces that support these strategies might have some inspectable representation of history such as the go function available in many web-browsers.

- This Table summarizes the interaction strategies given above and relates these to the abstract resources that are required for their use.

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

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.
  1. as a means of comparing different interface designs.
  2. as a means of analyzing interaction scenarios.
  3. as a way of generating design alternatives and analyzing their effects on user performance.

Agenda

- Introduction
- Resources Model
- Security goals
- The Resources Model in Security
- Empirical Study
- Future Works
- References

Security Goals

- 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.
- 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 Goals (Cont.)

- 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 7 security concepts with some solutions in implementation level, which are presented in table 2.

Security Goals (Cont.)

- 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.

Security concept	Solutions in implementation
Authentication	Passwords, Tokens, Biometrics
Authorization	Access Control Lists
Confidentiality	Cryptography, 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).

Agenda

- Introduction
- Resources Model
- Security goals
- The Resources Model in Security
- Empirical Study
- Future Works
- References

The Resources Model in Security: A New Solution for Designing Secure GUI

- 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.
- ***The Idea is RAW!***

The Resources Model in Security: A New Solution for Designing Secure GUI (Cont.)

- Another approach, that we extended it here, is that considering security as constraint on functional requirements of a system.
- Haley in 2008 describes 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 harm.
- We will define some scenarios and use them for this work.

The Resources Model in Security: A New Solution for Designing Secure GUI

- 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.

Agenda

- Introduction
- Resources Model
- Security goals
- The Resources Model in Security
- Empirical Study
- Future Works
- References

Empirical Study

- 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.
- Our case study is designing new Test and Survey feature in e-learning system of Ferdowsi University of Mashhad (FUM) that we just used the Test part of it.

Empirical Study (Cont.)

- First we describe the 3 first steps
- 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).

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.

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.

Security Requirements

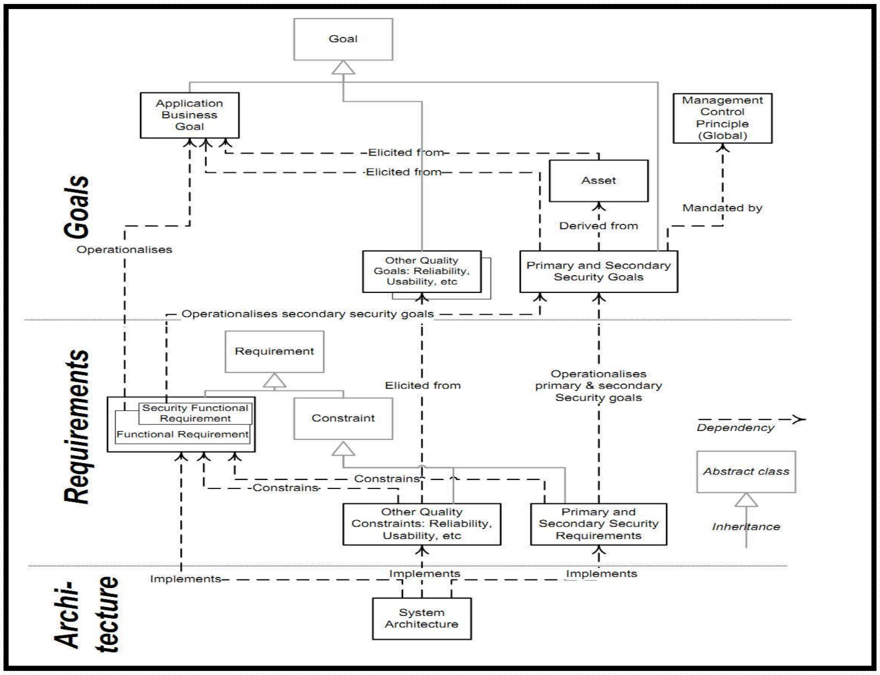
- Below we just describe the ones related to designing new test:
  - Test can be private for teacher and students of that course or it can be public.
  - Teacher can categorize students to take different tests.
  - Teacher can define a valid date and time and a certain duration time.
  - Students can take an exam anonymously.
  - Teacher could specify a valid IP range for test. Students have to take the test just in that range.
  - System should lock the content of the test during the test time.
  - System should provide a back upping feature for current test.
  - Also teacher and students can trace the steps of the exams like dates and time, answers and etc.

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.
- 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.

Scenarios : 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:
  1. **Logging Tests and Surveys part**
  2. **Creating new test**
  3. **Defining questions in repository**
  4. **Adding questions to new test**


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

Scenarios : 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:
  1. **Logging in e-learning system- authentication**
  2. **Defining test valid dates and times- authorization**
  3. **Defining test duration time - authorization**
  4. **Defining valid IP range- confidentiality and availability**
  5. **Back upping- data integrity**
  6. **Not permitting to use course content in the website during the test- authorization and confidentiality**
  7. **Categorizing students to take the test- authorization**
  8. **Anonymous test (taking by students and correcting by teacher- authorization and confidentiality**
  9. **Traceability of the test by students and teacher- accountability and non-repudiation**

Scenarios : Security Scenarios

- The general scenario covers some of the above tasks (1,2,3,4) and others (5,6,7,8) 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.

Evaluating GUI

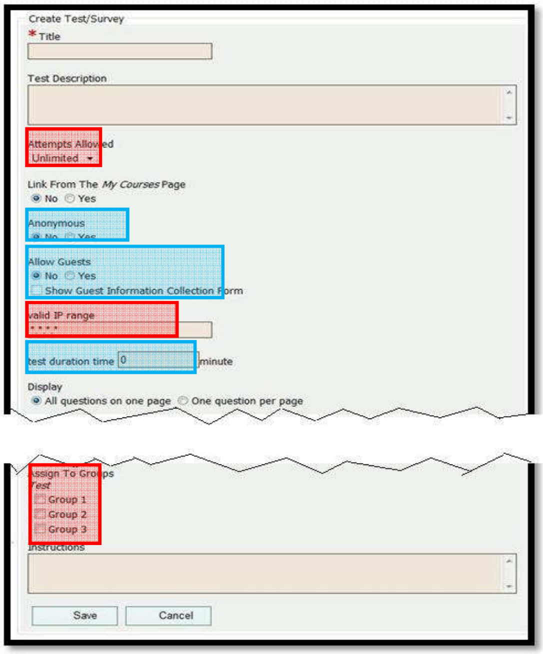
- We evaluate the GUI in 2 main parts; General Fault Designs and Security Fault Designs.
  - ***General Fault Designs***
  - ***Security Fault Designs***

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. 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.

General Fault Designs (Cont.)

- 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.



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

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 on one of the biggest problems users faced in these scenarios;
  - Categorizing students to take the test- authorization
- 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.
- 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.

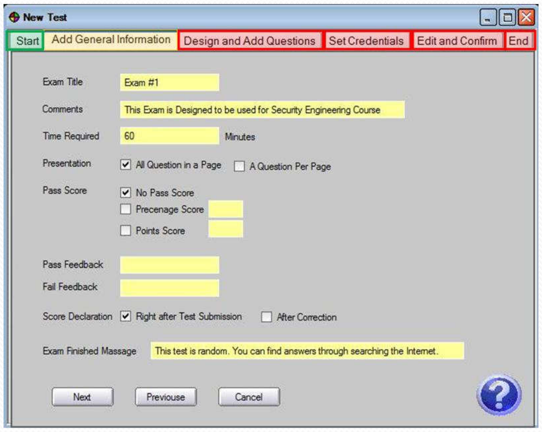
Exploring Design Alternative

- The main problem users faced, was the process of adding questions to the test and assigning proper accesses to students groups
- 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.

Exploring Design Alternative

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

Exploring Design Alternative



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

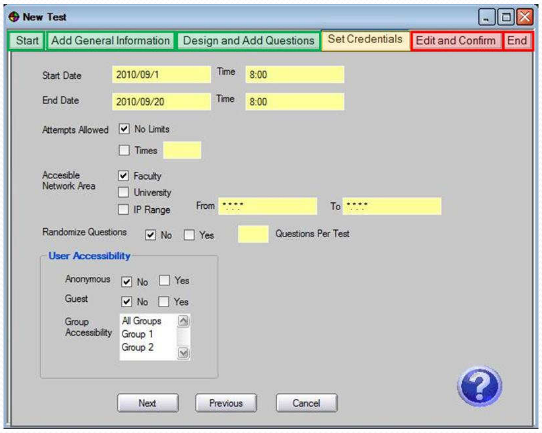
Pass Score: ☑ No Pass Score ☐ Precenage Score ☐ Points Score

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.



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

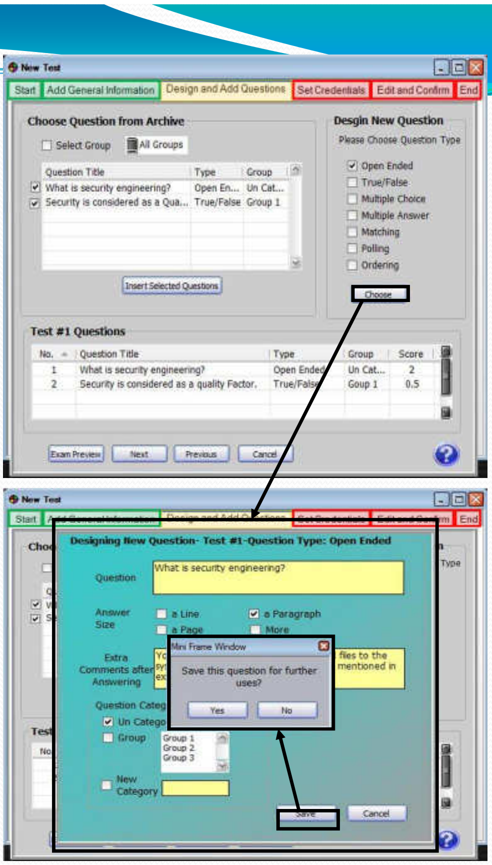
Randomize Questions: ☑ No ☐ Yes Questions Per Test

User Accessibility

Anonymous: ☑ No ☐ Yes

Guest: ☑ No ☐ Yes

Group Accessibility: All Groups, Group 1, Group 2



Choose Question from Archive

☐ Select Group All Groups

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

Desgin New Question

Please Choose Question Type

☑ Open Ended
☐ True/False
☐ Multiple Choice
☐ Multiple Answer
☐ Matching
☐ Polling
☐ Ordering

Test #1 Questions

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

Designing New Question- Test #1-Question Type: Open Ended

Question: What is security engineering?

Answer Size: ☐ a Line ☑ a Paragraph ☐ a Page ☐ More

Extra Comments after Answering: Yo... sy... ex... files to the ... mentioned in

Question Categ...: ☑ Un Catego... ☐ Group (Group 1, Group 2, Group 3)

☐ New Category

Mini Frame Window

Save this question for further uses?

Agenda

- Introduction
- Resources Model
- Security goals
- The Resources Model in Security
- Empirical Study
- Future Works
- References

Future Works

- As future works, we would like to design a framework to compare the design alternative with the old one.
- Also applying other HCI theories in the field of security would be helpful for designing more secure systems.

Agenda

- Introduction
- Resources Model
- Security goals
- The Resources Model in Security
- Empirical Study
- Future Works
- References

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/

00:00

धन्यवाद

Hindi

多謝

Traditional Chinese

ขอบคุณ

Thai

Спасибо

Russian

Gracias

Spanish

شكراً

Arabic

Thank You

Brazilian Portuguese

Grazie

Italian

Danke

German

Merci

French

நன்றி

Tamil

多谢

Simplified Chinese

Korean

متشکرم

Farsi

ありがとうございました