

Evaluating the Effectiveness of Notations for Designing Security Aspects

Siraj Un Muneer*, Muhammad Nadeem**, Bakhtiar Kasi**, Mir Hamayoun Yousaf***

*Department of Computer Science, BUITEMS

**Department of Computer Engineering, BUITEMS

***Department of Telecom Engineering, BUTEMS

siraj.muneer@ieee.org, dr.nadeem@ieee.org, bakhtiar.kasi@gmail.com, hamayoun2@yahoo.com

Abstract— Existing software design approaches used to design security aspects lack notations that can be used to design the security requirements of a system which may result in more complex and dense designs. However, literature review has shown that there are some proposed notations to represent security aspects within a system. Other researchers have proposed evidence-based principles that provide basis for constructing Visual Notations in Software Engineering. The principles include: Graphic Economy, Perceptual Discriminability, Semiotic Clarity, Cognitive Integration, Visual Expressiveness etc.

In this research, we have evaluated a set of proposed notations in accordance with principles for designing cognitively accepted graphics by conducting a human subject study. A questionnaire was distributed among a group of graduate and post-graduate students. The goal was to validate the notations based on theory of “How Visual Notations Communicate?”. This study will help researchers in developing cognitively accepted notations and set a course for designing new notations for UMLsec that are perceptually, semantically and visually expressive.

Keywords— Visual Notations, security modelling, security designs, UMLsec, extended notation, subject-based experimentation

I. INTRODUCTION

Many researchers have contributed significantly to the objective of Model Driven Security by extending and proposing popular modelling notations to model security aspects [1]. Generally, security is focused at the finale of a project by applying patches of defensive mechanisms such as IDS, DOS and firewalls etc. *secure software engineering* is needed in order to meet the security needs of a complex software system. Secure software engineering suggests that security concerns must be the top priority and should be addressed in inception phases of any development life cycle [2]. In fact, secure software engineering should be introduced at the requirements gathering phase. The current software designing tools have notations / symbols that can be used to design the plain functionality of the system but lack the notations to design its security. For this purpose, some

researchers have proposed state chart diagrams others have extended UMLsec and so on. However, notations proposed by some researchers claim to fill the gap which they have not empirically evaluated. Therefore, in this paper we intend to conduct a human subject experiment in order to check whether those notations are acceptable or not. this study can be used to validate existing notations in literature to build artifacts representing security requirements.

A. Motivation

Since software is an integral part of Information Technology infrastructure for enterprise as well as individual level; security is the major concern of software at any level. UML supports a big collection of notations and symbols but to design the security aspects of a system, one must increase the complexity of a diagram using many state-charts and notations etc. There are few notations proposed in the literature to represent the security aspects and claim to save a lot of extra work in the design models. The proposed notations have not been evaluated based on human cognition and comprehension.

II. LITERATURE REVIEW

In this section, we have discussed the literature survey as to how many notations were proposed and some explanation of principles.

A. The Nine Principles

Moody et al. [1] proposed principles based on best evidence synthesis approach: deduced from empirical evidence and theory about cognitive effectiveness of visual representations. Altogether derived from a design theory, called the Physics of Notations as it focuses on the physical (perceptual)

properties of notations rather than their logical (semantic) properties. The principles used for designing cognitively effective visual notations are presented in the modular honeycomb structure in Figure 1.

1) Semiotic Clarity

This principle means that there should a one-to-one correspondence among symbols and their pertinent concept. For example, by looking at a musical note the reference of

songs or musical files is inferred. When the one-to-one correspondence is absent, one or more of the four ontological analysis' anomalies can occur:

- *Redundancy in Symbol* which means multiple graphical can denote the similar semantic construct.
- *Overloaded Symbol* means two separate concepts can be represented by same graphic.
- *Excess in Symbol* means the graphic doesn't correspond to any of semantic construct.
- *Deficit in Symbol* means that the concept is impossible to represent by using graphical symbols.

2) Perceptual Discriminability:

The principle states that it should be easy to distinguish between different symbols from each other.

A diagram can be accurately inferred only if its symbols are properly discriminated. Discriminability is maximized if the visual distance is increased in the design of a notation. For measuring the discriminability, the number and ranges of visual variables are considered. The more the numbers and ranges are high, the more visual distance is increased.

Visual variables are of two types shown in Figure 1 and Figure 2:

- Retinal Variables
- Planar Variables

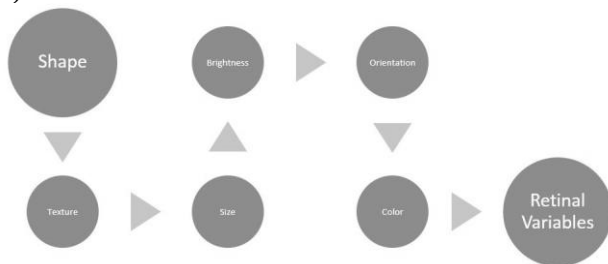


Figure 1

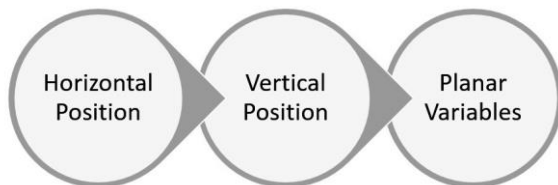


Figure 2

3) Visual Expressiveness

Notations should make use of different visual variables shown in Figure 1 effectively. In other words, absolute utilization of graphic design space is the major concern of visual expressiveness. This principle differs from the one discussed earlier, which focuses the visual distance between symbols within the same notation. Making use of color, size and brightness increase visual expressiveness which as a result increases human discrimination ability in an adaptive manner [4]

4) Semantic Transparency

The extent to which the inference of a symbol from its appearance is done is known as semantic transparency. When the appearance is suggestive of meanings of graphical representations, the principle of semantic transparency is satisfied. Some graphical representations are demographically variant.

5) Dual Coding

According to *The theory of Dual Coding*; the combined usage of graphics and text is always effective rather than using either of those [3]. In order to clarify meaning, text must be used as form of redundant coding. Pictures and texts are not foes and should never be mutually exclusive. Hybrid (text + graphic) symbols are best used in showing multiplicity in class diagrams or showing the relationship in ERDs.

6) Graphic Economy

Nordbotten et al. [4] have used the term Visual Vocabulary to calculate the graphic economy; which is the number of symbols used in a graphic representation. Novices are affected more than the experts by the graphic complexity, as they have to consciously interpret the meanings. Empirical studies have shown that novices face difficulties in interpreting SE diagrams [4], [5].

Excessive graphic complexity is dealt using three ways:

- By reducing semantic complexity
- By introducing symbol deficit
- By increasing visual expressiveness

7) Complexity Management

There must be mechanisms to control complexity within the visual vocabulary. Complexity has a great effect on cognitive effectiveness as the information depicted by a symbol is limited by perception and cognition.

- **Perceptual Limits:** Size of diagram increases the ability to discriminate between diagram elements.
- **Cognitive Limits:** Comprehension is limited by working-memory capacity, when the number of diagram elements are exceeded, the cognition overload ensues and comprehension is degraded [5].

8) Cognitive Integration

Perceptual and conceptual integration must be achieved by the mechanisms provided by a notation for its understanding. Perceptual integration allows the reader to positively navigate between diagrams. Conceptual integration allows to fit state-charts in an overall map of whole system. This principle is closely related to complexity management. This principle is of great significance in SE, where multiple diagrams are used to represent the whole system; problems are typically represented by system of diagrams rather than drawing singly [1]. Cognitive integration is equally applied to diagrams of same type:

- **Homogeneous Integration** - e.g. a set of leveled DFD
- **Heterogeneous Integration** - e.g. a suite of UML diagrams

9) Cognitive Fit

The theory of *cognitive fit* is widely accepted in Information Systems where it is used for managerial decision making and program maintenance [6]. The theory states that various representations of information are suitable for various audiences and tasks. Most SE notations manifest *Visual Monolinguisism* using single representation for various purposes [7].

Certain set of semantic constructs are represented by a visual form known as visual dialects. Multiple dialects usually exist in some notations; for example, Data Flow Diagrams (DFDs). DFDs comprise two famous dialects:

- DeMarco Dialect
- The Gane and Saron Dialect

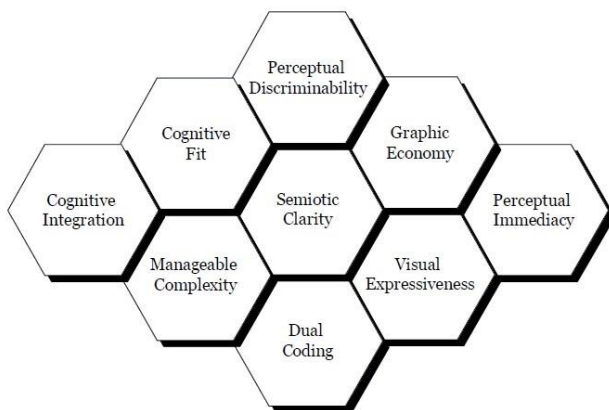


Figure 3

B. UMLsec

An extension of UML and for a quite long time now researchers have contributed quite enough in suggesting new notations, state-charts scenarios etc. UMLsec addresses four security concepts:

- Fair Exchange
- Confidentiality or secrecy
- Secure information flow
- Secure communication link

Doan et al. stated the abstraction level of UMLsec to be very high which causes problems in depiction of security requirements while designing software artifacts.

C. MAC and UML Artifacts

The notion of security at the design level has been focused adequately by the researchers. Accuracy and precision of policy definition, authorization and authentication, assurance and enforcement are the key concepts playing pivotal role at all stages of the life cycle of a software. For many years, to specify and visualize, construct and document the aspects of a software, UML has been the biggest tool. UML provides diagrams to help represent various perspectives for several stakeholders. For example:

- Use-case diagrams to show the interaction of users with the system.
- Class diagrams to draw the relationship of static classes.
- Sequence diagrams to represent the dynamic behaviour of instances of classes.

- Data Flow Diagrams to show the flow of the system.

However, UML's lacks support for defining security requirements for such diagrams and their constituent elements (e.g., systems, use-cases, classes, instances, generalizations, relationships, methods, data). Some researchers have claimed to that by incorporating Mandatory Access Control (MAC) into use case, class, and sequence diagrams, providing support for the definition of clearances and classifications for relevant UML elements have been addressed [5].

D. Model Driven Security

In the area of *Safety Analysis* fault trees were introduced to show the sequence of faults and handle them accordingly. Similar approaches came into being, namely, threat trees and attack trees. Furthermore, these trees were enhanced, and a new term was coined: attack-defence trees [4] which shows both the anticipated attacks and their countermeasures.

In the new millennium, several graphical languages were introduced to incorporate security considerations and design efforts of software systems. Commonly, these propositions are adaptations of existing modelling languages provisioning notations extensions and security aspects. Some sets of such proposals are discussed in the earlier section namely, UMLsec.

E. Security for mobile application designing

Security is of great concern when designing applications for mobile platforms as public medium (like wireless and satellite etc). There is a greater chance of theft due to the small size and mobility and power consumption issues.

III. RELATED WORK

In this section, we have discussed the only study we found which directly relates to our research.

A. Extension of UML State-chart Notations

Mohamed et al. [5] suggested another set of notations used with conventional state-charts. The thunderbolt sign is used to represent the threatened state, the shield is used to represent the defensive state and the infamous skull and bones sign is used to show the compromised state etc. The newly proposed notations [5] are depicted in the table below:

TABLE 1. SOME OTHER NOTATIONS FROM LITERATURE

Not. 1	Not. 2	Not. 3	Not. 4	Not. 5
				

An industrial survey was conducted to check whether the symbols convey clearer meaning of their semantics used in their depiction. The main idea behind conducting the survey was to respond to their two research questions [5]. First regarding the semantic transparency of the new notation and second comparison from the original notation.

A scenario drawn using the conventional notations for sending MMS Notification and the threatened state of MMS Opened. New notations were then used in a context to show

their usages in Figure 4. Same scenario is represented in Figure 5 using classical UML notations which signifies the need of new notations.

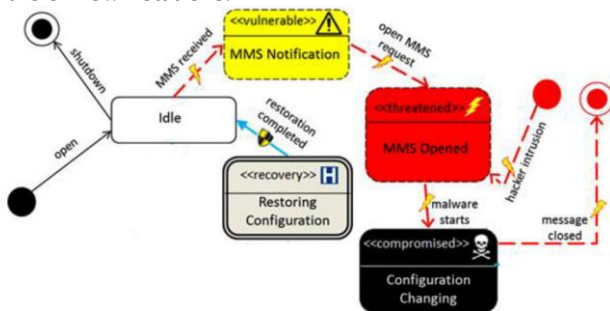


Figure 4 Scenario containing all new notations

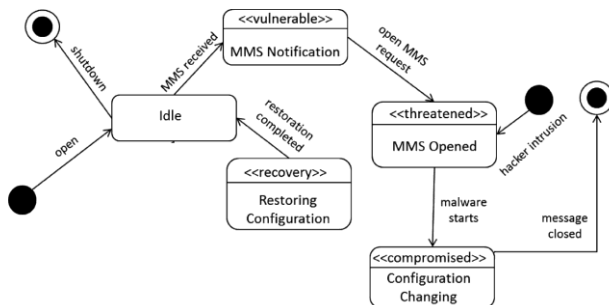


Figure 5 Scenario without new notations

IV. RESEARCH METHODOLOGY

A. How this research is organized

Qualitative data about all the notations was collected by conducting a human subject study. A questionnaire was designed that contains the visual notations used in small contexts, as to where and when it can be used. All the nine principles are mentioned having a 5-point Likert Scale. Questionnaire is distributed among several groups of students according to their academic levels. Each member of the group was given a short pre-survey quiz to analyse their comprehension of UML diagrams and relevant SE knowledge. The intention was to reduce the threats to validity of study. In this section we have discussed the collection of notations selected for evaluation and how the responses of the survey are evaluated.

B. Set of Notations for Evaluation

We have selected five notations from a proposed set of notations discussed in a master's thesis.

1) Notation for Encryption:

Encryption is an integral part of security enforcement. Yet at the design time, depicting the process for encryption is challenging. The encryption is shown when content is copied from source to destination in Figure 6.



Figure 6

2) Notation for Session Maintained Communication:

When client connects to server, most of the sessions have their pertinent session IDs. The context is illustrated where the communication between source and destination is maintained using sessions in Figure 7.



Figure 7

3) Notation for Content Verification using Hash

Cryptographic hash functions are part of mathematical algorithms. These hashing techniques are used to verify checksums which intern verifies and validates the data streams. When hash functions are to be illustrated in the architectural schema, designers are left with big challenge. So, verifying the input data values given hash values is shown in Figure 8.



Figure 8

4) Notation for Anonymous Access:

Anonymous access is usually gained on the resource. It is important to display whenever there is a risk of anonymous access in order to invoke necessary security measures. Figure 9 shows notation used.

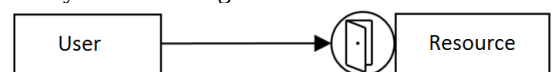


Figure 9

5) Notation for authentication using Sensory Challenge:

Sensory challenges like Captcha, biometrics etc. are used to verify that the user is a real human and not a bot. So, the sensory challenges are used to authenticate user while communication is established, or data is transferred between source or destination. Figure 10 displays the context of sensory challenges.

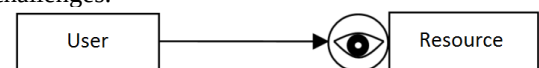


Figure 10

C. Further Explanation using Scenarios

The notations are presented within a scenario using DFD of a system. Design using the conventional notations is drawn in Figure 11.

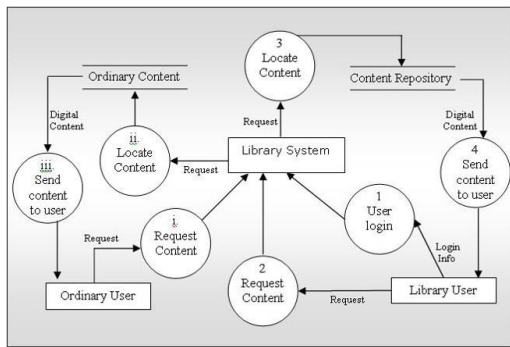


Figure 11 DFD using classical notations

It is quite visible that the complexity shown in Figure 11 is decreased using the proposed set of notations in Figure 12. Many states have been removed and just notations are serving the intended purpose in Figure 12.

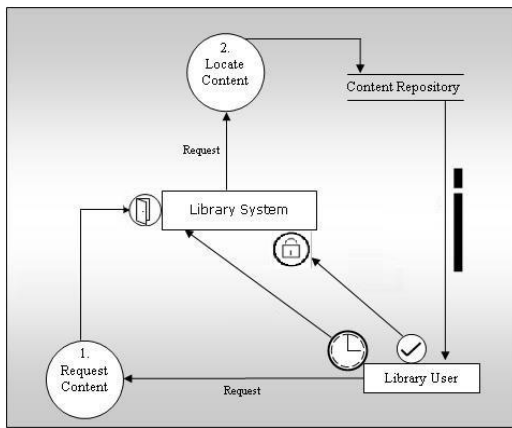


Figure 12 DFD Using proposed notations

D. Evaluation

We developed a questionnaire and distributed among four groups of students. The questionnaire included a demographic information page which helped us collecting the interests and major area of study of the participants.

Group 1: Four post-doctorate students having their Ph.D.(s) in Computer Science and Software Engineering.

Group 2: Four post-graduate students, two of them having their major in Software Engineering and rest from Computer Science background.

Group 3: Twenty graduate students, 17 out of them were of Computer Science background. So, we skipped the rest 3.

Group 4: Fifteen survey results were received from an uploaded copy of questionnaire on the SurveyMonkey and the background of the persons filling out the survey was unknown but most of them attempted the pre-survey quiz which was correct. This shows the participants were somehow related to software engineering.

We excluded 4 responses. So, in total we evaluated a total of $(4+4+17+13) = 38$ responses. The responses were then put in a statistical tool and clustered bar charts were produced presenting the data. On the CBC (Clustered Bar Chart), the vertical scale represents the principles and the horizontal scale shows the Likert Scale. Bars on the chart are labeled with an average calculation of scored scale.

V. RESULTS AND DISCUSSIONS

The result set of 38 responses was evaluated using Microsoft Excel and the CBCs (Clustered Bar Charts) were plotted using the average calculated for all responses respectively.

1) Notation for Encryption

The graph shown in the Figure 13 is for the notation used for encryption. It is quite evident that figure is perceptually discriminated and cognitively integrated. Visually the graphic is not quite expressive. Complexity of the figure is also not very well managed. Dual coding is high which signifies, if encryption notation is used, it would connote the intended meaning without the need of text representation in it.

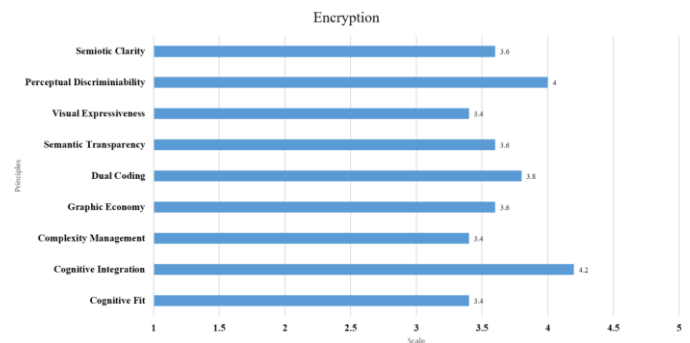


Figure 13

2) Notation for Session Maintained Communication

The Figure 14 displays the results for symbol used for session-maintained communication between source and destination. The notation's score for dual coding is significantly low. This highlights that the graphic alone is not enough; annotating the notation might have a better effect.

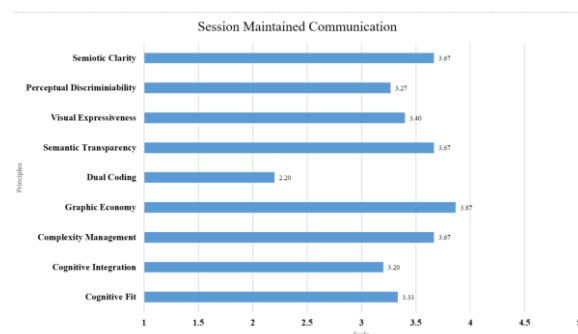


Figure 14

3) Notation for Content Verification using Hash

Hashing technique can be laid out using the proposed notation and the results after the evaluation are presented in the Figure 15. The clustered bar chart states that there is an average score for the all the principles. This means the graphic is acceptable in many aspects. Dual coding has a relatively high score which means the figure is better understood without any textual description.

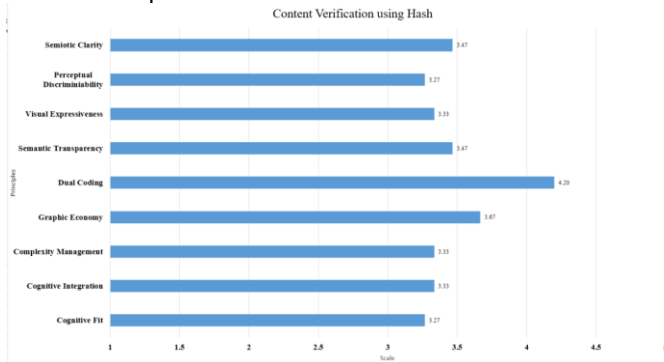


Figure 15

4) Notation for Anonymous Access

Anonymous access was evaluated and a variation in the opinions is detected shown in Figure 16. Dual coding is notably low. Textual description seems necessary. The notation doesn't give the impression to satisfy semiotics. The semantic construct appears to be good. There is higher cognitive integration which makes it clear that the notation has a comprehensible design.

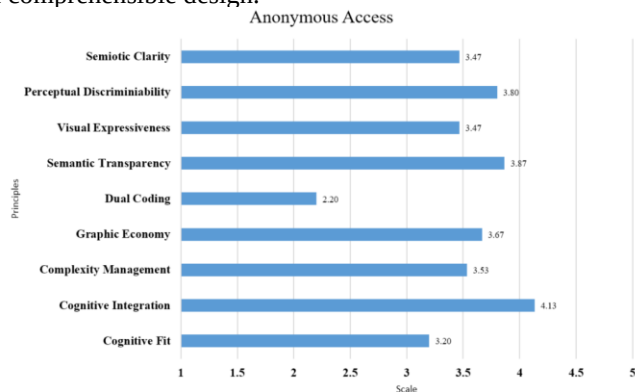


Figure 16

5) Notation for Verification using Sensory Challenge

The notation used for verification using sensory challenge satisfied the principles which is shown in Figure 17. The notation is cognitively integrated and semantically transparent. Visual expressiveness and complexity management is low. The figure exhibits less cognitive fitness and dual coding. This

shows that annotating the notation must be done to depict the true sense.

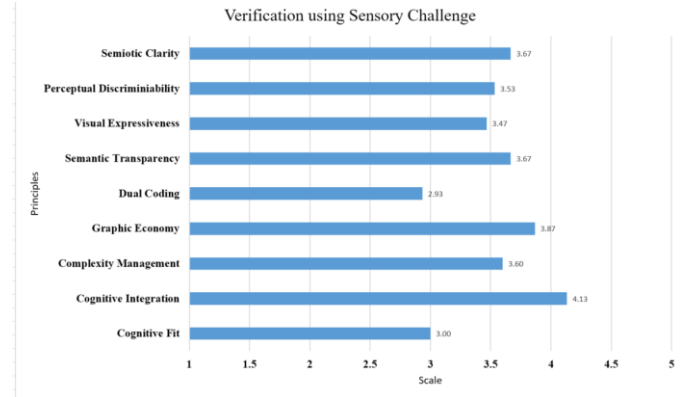


Figure 17

CONCLUSION AND FUTURE WORK

The notations have satisfied the principles with a good average. Some aspects of figures were weak though. The notations need to be redesigned considering the results of the questionnaires, in order to yield good results and achieve better cognition. In future, we intend to design new cognitively accepted notations using tailored set of principles. In addition, guidelines can be made for designing new notations using the Physics of Notations.

ACKNOWLEDGMENT

Our gratitude to all the participants who helped us in completing our survey. Special thanks to Dr. Nadeem and Dr. Bakhtiar who helped resolving the errors in Latex.

References

- [1] D. Moody, "The physics of notations: toward a scientific basis for constructing visual notations in software engineering", IEEE Transactions on Software Engineering, vol. 35, no. 6, pp. 756–779, 2009.
- [2] T. Doan, S. Demurjian, T. Ting, and A. Ketterl, "Mac and uml for secure software design," in Proceedings of the 2004 ACM workshop on Formal methods in security engineering. ACM, 2004, pp. 75–85.
- [3] A. Raghunathan, S. Ravi, S. Hattangady, and J.-J. Quisquater, "Securing mobile appliances: new challenges for the system designer," in Design, Automation and Testing Europe Conference and Exhibition, 2003. IEEE, 2003, pp. 176–181.
- [4] J. C. Nordbotten and M. E. Crosby, "The effect of graphic style on data model interpretation," Information Systems Journal, vol. 9, no. 2, pp. 139–155, 1999.
- [5] M. El-Attar, H. Luqman, P. K'arpati, G. Sindre, and A. L. Opdahl, "Extending the uml state charts notation to model security aspects," IEEE Transactions on Software Engineering, vol. 41, no. 7, pp. 661–690, 2015.

- [6] J. Jürjens, "Umlsec: Extending uml for secure systems development," International Conference on The Unified Modeling Language, pp. 412–425, 2002.
- [7] T. M. Shaft and I. Vessey, "The role of cognitive fitting the relationship between software comprehension and modification," Mis Quarterly, pp. 29–55, 2006.
- [8] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information." Psychological review, vol. 63, no. 2, p. 81, 1956.
- [9] M. Nadeem, "Notations for facilitating software security design."Faculty of Information and Communication Technology, Balochistan University of IT, Engineering, and Management Sciences, Quetta, Pakistan, 2007



Siraj Un Muneer has been associated with Balochistan University of Information Technology, Engineering, and Management Sciences since 2005. He has completed his MS and currently perusing his PhD Degree from BUIITEMS. He has secured many positions in curricular and co-curricular activities in BUIITEMS.



Dr. Muhammad Nadeem has been associated with Balochistan University of Information Technology, Engineering, and Management Sciences since 2003. He is a Fulbright alumnus and has completed his Ph.D. in Computer Science with specialization in software security from Mississippi State University. He has presented number of research papers in well reputed conferences abroad. Based on his teaching contributions, he was awarded Best University Teacher award by Higher Education

Commission (HEC) of Pakistan in 2007. He is currently looking after the Department of Computer Engineering at BUIITEMS.



Dr. Bakhtiar Kasi is the Dean of Faculty of ICT, BUIITEMS. He has completed his PhD from Lincoln University, Nebraska. He is a Fulbright Alumnus. And has published a number of research articles in various reputable journals. He has been working for Global software development with focus on coordination issues and conflict free task assignments, machine learning, big data analysis, SAT solving and information retrieval.



Dr. Mir Hamayoun Yousaf Shahwani is serving as an Assistant Professor at Telecom Engineering department. He is currently the chairperson of the department. He has completed his PhD from Sungkyunkwan University, South Korea in Electrical and Electronics. His area of expertise is resource allocation in M2M and Vehicular Networks.