Usability of User Interface Styles for Learning a Graphical Software Application

Andisheh Feizi[1], Chui Yin Wong[2]
[1]Universal Usability and Interaction Design (UUID) SIG
[2]Interface Design Department, Faculty of Creative Multimedia,
Multimedia University, 63100 Cyberjaya, Malaysia.
feiziandisheh@yahoo.com, cywong@mmu.edu.my

***Abstract*-This paper examines usability of user interface styles for learning a graphical software application, namely Adobe Flash CS4. An empirical study was performed to investigate the usability attributes of effectiveness, efficiency and satisfaction scores for learning Adobe Flash CS4. There were 32 participants recruited whom consist of interface designers and software developers. A set of 10 tasks were designed to compare the different effects of user interface styles between graphical user interface (GUI) and command line interface (CLI). Performance variables (effectiveness, efficiency, duration, number of errors and number of helps) were measured for tasks performed by all the participants in the test. Satisfaction score was measured using QUIS (Questionnaire for User Interface Satisfaction) tool. The result revealed that the average effectiveness score higher than 75% (96.88%), and average efficiency (7.43) for all the 10 tasks given. The satisfaction score also showed above average (>3.5) for majority of the user interface satisfaction attributes of software regardless of users' background.**

Keywords: ***usability, user interface styles, graphical user interface (GUI), command line interface (CLI), graphical software application***

I. INTRODUCTION

User Interfaces (UI) have been around since the invention of computers, even before the field of Human-Computer Interaction (HCI) was initiated [1]. Since UI design is an important component of HCI system, software designers have been tasked to create interfaces that effectively predict and interpret the operator's needs besides allowing the user to perform tasks in natural ways [2]. Thus, good usability a software user interface will bring satisfactory user experience.

Throughout the last four decades, programming has evolved from platforms from reading and debugging a programme using command language strategies, and move into Graphical User Interface (GUI) [3]. In the context of programming, user interface design plays an important role. User interface is the medium for users interact with computers [4].

The development of GUI software applications has been one of the noteworthy improvements in programming field that reduce the difficulty of remembering syntax and semantics with the guidance of menu-based interactive properties it delivers [5]. However, there are still circumstances that require the users to use command-line interface (CLI) since CLIs often afford more options than their equivalent GUIs, leading to greater flexibility available for users or one can perform a task by using command that its function is not supported by its GUI counterpart [16]. For the experts who are familiar with the software application, he/she prefers to opt for command line interfaces (CLI) rather than graphical user interfaces (GUI) for task execution, whilst novice computer users prefer menus [5]. CLI is quite inconvenient environment for new generation of users since they are substantially used to GUIs [30]. Hence, it is important to examine usability level of different user interface styles of using GUI and/or CLI in learning a graphical software application.

In regards of the issue of users learning a software application to produce a new knowledge, usability is considered as an essential attribute for quality of software design. It is important when a large number of people use software packages either at work or for their personal tasks. Mainly because most of them are not regular users and they do not work with software several hours a day for a long period of time [8].

Having analyzed the above-mentioned issues, this paper aims to examine the usability of user interface design styles comparing GUI and CLI for learning a graphical software application by interface designers and software developers. The followings are the objectives of this research:

(i) To evaluate usability attributes in terms of performance measure (effectiveness, efficiency, time duration, number of errors, and number of helps) for a graphical software application;
(ii) To evaluate user satisfaction for user interface of a graphical software application;
(iii) To examine usability of different user interface styles (i.e. GUI and CLI) for learning a graphical software application.

II. LITERATURE REVIEW

In Human-Computer Interaction (HCI) discipline, researchers and professionals in the field mainly focus on five 'E' of usability, which propose an interactive system must be 'effective, efficient, engaging, error tolerance, and easy to learn'. The goal is to provide users with information systems (i.e. software interface) and work environments in which they can do their tasks efficiently.

Software interface is an essential medium for information transmission between users and computers for successfully performing various tasks, besides designing new software products, and is also one of the significant concerns in HCI field [6]. On the other hand, assimilating HCI with Software Engineering (SE) processes is a desire, increasingly reported. User interface is where HCI and SE

978-1-4673-1938-6/12/$31.00 ©2012 IEEE

link up. The only distinction between them is their perspectives. Conventionally, HCI deals with "social, cognitive and interactive phenomena". However, the fundamental purpose of SE is "the designing of software, the process of software construction, structural design, reliable functioning and design for convenience" [6]. During the design process of user interface, HCI concentrates on the design attributes of interface such as ease of use (or usability) and supplementing attractiveness, inspiration and affective responses. While SE emphasizes on how functional features of the software will be provided for users [6].

A. Usability of Software Design

Based on research from theoretical and practical perspectives in software field, usability is defined as *"the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component"* [25]. ISO/DIS [29] demonstrates usability as *"the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"*.

Bevan [9] proposed a detailed description for the term of usability, which considers effectiveness, efficiency, and satisfaction as quality factors for usability. Subsequently, ISO 9126-1 explains usability in terms of learnability, operability, understandability, and attractiveness [7].

Nielsen [8] proposed one of the popular definitions for usability involving the learnability and memorability of a software program, its capacity to avoid and control user errors, its efficiency of use and user satisfaction. Technically speaking, "efficient use of the computer is intrinsic to usability" [9]. To the sense that in assessing the usability of a software product it is required to examine user performance in addition to considering the amount of effort a user puts in applying the software. Therefore, a system is not usable if it requires high amount of effort in order to complete a task with a high performance [9]. The most prevalent perspective in the field of software engineering (SE) is that usability is intertwined with the user interface [10]. Shneiderman and Plaisant [11] determines features of user interface design based on evaluation of several human factors such as length of time to learn, learner's rate of errors, pace of performance, user's satisfaction, and retention over time. However, despite the significance of usability in software development, it is still unsatisfactory in majority of software programs [10].

Measuring software usability is a significant indicator of the deficiency level of software application, and software testing is the foundation for software usability enhancement [26]. Software usability is not directly measurable; it can be simply evaluated indirectly through observing measures, such as effectiveness, user's satisfaction and performance assessment [27].

B. User Interface Design (UID)

User interface is defined as "features that facilitate communications between a user and an information system" [12]. User Interface Design (UID) refers to the "overall process of designing how a user will be able to interact with a system" [13]. UID concerns about 'facilitating clear and accurate information exchanges, efficient transactions, and high-quality collaborative work' [14]. User interface design is a fundamental concern for the usability of a software product [15].

C. Categories of User Interface Styles

According to IBM [4], user interfaces can be categorized into three fundamental groups:

(i) Command line user interface that is a fully-text display mode on a computer screen controlled by a keyboard, in which users type in data, commands or instructions notifying the computer to do a task. A common example of a Command Line Interface (CLI) is UNIX that text is only shown on the entire screen [16].

(ii) Menu-Driven user interfaces, "in which a user is provided with a hierarchically organized set of choices" [4]. Robertson et al. [17] mention that users fail to correctly perform a task on a menu when structure of the menu is complex. In a menu-based environment, a user clicks on a command from pre-defined array of commands exhibited in menus. If command names on the menus are understandable and well organized, users can easily perform their tasks since discovering a command in a menu is equivalent to recognition instead of recall [18]. This type of interface is ideal for novice learners as they support error handling; however, they can appeal to expert users if arrangement and selection processes are quick enough as well as convenient shortcuts are provided [18]. On the other hand, possibly menus are slow for regular users besides the fact that numerous numbers of menus may result in overload and too much complexity.

(iii) Graphical User Interfaces (GUIs), which is an interactive human-computer interface that makes use of widgets including windows, icons, menus, buttons, dialog boxes and etc. It is often directly manipulated by a computer mouse, and to a limited extent by keyboard [16]. The widgets are basic visual blocks combined in an application that hold all the data processed by the application and the available interactions required to achieve goals of the user. Users can interact with information by manipulating visual widgets provided; according to the kind of data they hold [19]. GUIs are direct manipulation systems currently familiar to users in the Windows environment.

According to Shneiderman [5], users can track down information more quickly with GUIs compared to CLIs, besides that, user's understanding and satisfaction is higher for GUI applications. Another investigation on the influence of interface styles on perceived ease of use and usefulness came to the conclusion that menu-based interface was more beneficial rather than command-based interface [32]. However, Davis et al. [32] compared DMI and CLI styles. Their results indicated no significant distinction on perceived ease of use. Wiedenbeck et al. [21] examined DMI (direct manipulation interfaces where GUIs are included), menu-driven and CLI. Their outcome showed that interface style did not affect participants' perception

towards the usefulness of the system, however DMI style was considered easier to use by participants. Having said this, we would like to find out which UI styles are more conversant for designers and programmers dealing with a software interface.

III. RESEARCH METHODOLOGY

We conducted a usability testing to evaluate the usability attributes of comparing different user interface design (i.e. GUI and CLI) of learning a graphical software application using Adobe Flash CS4. The research method is experimental study with a mixture of observation, user interface satisfaction questionnaire (QUIS) and user testing for data collection. The QUIS questionnaire was adapted for this research purpose since it is a validated instrument for conducting comparative evaluations for software applications [20].

A. Rationale of Graphical Software Applications

There have been several popular graphical software applications in the market that integrate graphical user interface and programming scripting functions all-in-one for interface designer and software developer to work seamlessly for a software development work. Adobe Flash CS4 is selected for the study because it is a software specifically designed by Adobe™ to bridge the development platform for interface designers and developers/ programmers. It combines GUI and CLI in the software itself and enables the developers and designers to work apart more effectively without losing each other's work in a software development process.

Generally, Adobe Flash is an authoring tool that can be utilized to design and create presentations, software applications that act in response to user's interactions. Flash projects can contain animation, video content and complex user interfaces. On one hand, Adobe Flash allows software developer to use Action Script (a scripting language) to create functions and determine how the elements in the application act, and the code also allows adding interactivity and logic in a project.

B. Apparatus and Testing Facility

The study was conducted at User Interface Lab. The apparatus use for the user testing is a laptop (with a 14 inch monitor, 4 GB RAM, 2.20 GHZ CPU having 1280 * 800 display resolution) as a workstation, Windows Vista Home version as operating system, and Adobe Flash® CS4 as a graphical software tool to carry out the task sets, and Camtasia Studio 3 was employed to record the screen capture for data analysis.

C. Participants and Tasks

32 participants were recruited with the background of interface design or programming for the study. The participants were tested individually. The user test took around 1 to 1.5 hours. Upon arrival, all participants were given a consent form before the experiment commenced. They were then asked to complete a demographic and software products experience questionnaire. Then, the participants were randomly given 10-minute trainings to learn the basic conventions of Adobe Flash® CS4. The participants are also given an average of 7 minutes to practice on their own and gain confidence before taking the tests. Subsequently, all participants were given a set of 10 tasks (Table 1) to perform in the software application. If the participants were unable to complete a task, they were free to proceed without task completion. Figure 1 below shows an example of screen shot for Task 8 (create mouse over event for an image).

TABLE I
TASKS DURING USER TESTING

Adobe Flash CS4		
Task No.	**Task Type**	**Task Explanation**
1	GUI	Set background color
2	GUI	Create text
3	GUI	Set font, size, color and make it bold
4	GUI + CLI	Create animation for text
5	GUI	Import image to the file
6	GUI	Place image on screen, resize it and make it symbol
7	GUI	Name image
8	CLI	Create Mouse Over event for image
9	CLI	Create Mouse Out event for image
10	CLI	Change image transparency

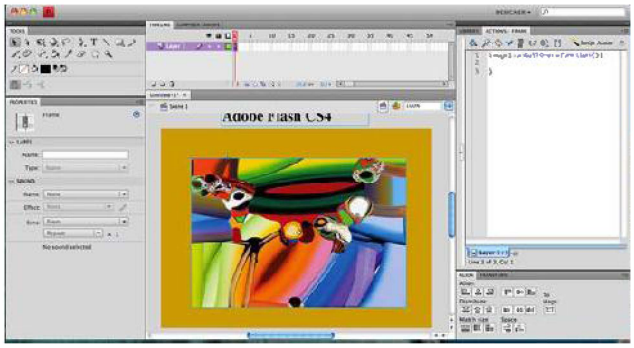

Figure 1. A screen shot example of Task 8 for creating a mouse over event for an image in Adobe Flash CS4.

D. Usability Metrics

The User Performance Variables for usability as follows:

- Effectiveness score: Percentage of successful completion for each task.
- Time taken: The total time spent to complete each task.
- Efficiency Rate: It is calculated by dividing effectiveness score by the time taken to do the task.
- Error: Any error made during performing each task.
- Help: Any help received during performing each task.

For subjective satisfaction, QUIS questionnaire measures users' subjective satisfaction with the interface of software applications on a 7-point semantic differential scale. The questionnaire covered items such as overall reaction, screen, terminology and software feedback, learning, and software capabilities.

IV. RESEARCH RESULT

The result is analyzed using statistical analysis software, SPSS 16. The data gathered from the user testing were analyzed using descriptive analysis with mean and standard deviation. The rationale was to compare effects of two different GUI styles and CLI styles on learning Adobe Flash CS4. The effectiveness score of each task is calculated for all users in the original test.

A. User Performance Analysis

Among all 32 users, 17 (53.13%) participants were designers with interface design background while 15 (46.88%) participants were programmers from IT background. Users were from both genders (22 male and 10 female). Users were asked about their knowledge and usage of Adobe Flash, 28 (87.5%) individuals already had training for Adobe Flash while 4 individuals did not have (12.50%).

One-Sample Kolmogorov-Smirnov test was conducted to examine normal distribution of effectiveness scores for all 10 tasks. Table 2 shows the results of user performance measure:

TABLE 2
A SUMMARY OF USABILITY TESTING RESULTS FOR ADOBE FLASH CS4

Task no.	**Effectiveness (%)**	**Time Duration (min)**	**Efficiency**	**Error**	**Help**
1	100	10.84	14.08	0.19	0.13
2	100	15.13	7.86	0.0	0.0
3	100	31.63	3.93	0.06	0.0
4	92.97	54.38	2.23	0.44	0.31
5	100	12.34	9.12	0.0	0.06
6	100	39.94	3.01	0.16	0.06
7	100	6.34	19.21	0.03	0.06
8	85.94	47.47	2.74	0.25	0.63
9	98.44	18.59	6.28	0.03	0.03
10	91.41	21.63	5.85	0.22	0.34
Mean*	96.88	25.83	7.43	0.14	0.16
SD**	5.01	16.62	5.48	0.14	0.20

Mean* indicates average for total 10 tasks.
SD*=standard deviation

The result in Table 2 shows that the effectiveness scores higher than 75% for all the 10 tasks (Mean for total task-96.88%, SD=5.01). However, the minimal effectiveness score is associated with CLI (Task 8 of creating Mouse Over event) and all GUI tasks are linked with maximum effectiveness score. On one hand, the minimum time duration taken to complete the tasks is 6.34 min (Task 7 of naming an image), which also means the task is simple to achieve. On another hand, the maximum time duration is 54.38 min (for Task 4), which indicates the task is more complex and challenging to complete in terms of creating animation using GUI and CLI technique. The average time for 10 tasks is 25.83 minutes.

In terms of efficiency, the total tasks score for 7.43 in average. Task 7 again achieves the highest efficiency rate of 19.2; however, the least efficient task to complete is Task 4 (2.23). This indicates that a task that combines GUI and CLI will take longer time for task completion. Apart from this, the average error rate for an overall task accounts for 0.14, which is considered minimal, and Task 4 shows the highest score of making mistakes. The number of help accounts for 0.16 for the total task, which is acceptable for task performance. The highest number of seeking help is Task 8, which shows the users are not familiar with creating a mouse over event for an image using CLI approach.

Table 3 gives a summary of comparing GUI and CLI in terms of user performance metrics. In this case, GUI tasks only include Task 1 to Task 7 except Task 4 because Task 4 combines GUI and CLI together, which makes comparison difficult. CLI task covers Task 8 to Task 10 only. In summary, GUI task is more effective (100%), more efficient (8.48) and taking less time (24.37 min) with less error rate (0.13) and number of help (0.09) to perform as compared to CLI tasks of effectiveness (91.93%), efficiency (4.96), longer time duration (29.33 min), higher error rate (0.17), and more help required (0.33) for Adobe Flash CS4.

TABLE 3
COMPARING GUI AND CLI USABILITY TESTING RESULT

Task Type/ no.	Effectiveness (%)	Time Duration (min)	Efficiency	Error	Help
GUI (Task 1-7, except 4)	100	24.37	8.49	0.13	0.09
CLI (Task 8-10)	91.93	29.23	4.96	0.17	0.33

B. QUIS Questionnaire Analysis

For subjective user satisfaction, data collected through QUIS questionnaire at the end of the test is summarized below (see Table 4). Users' satisfaction for Adobe Flash CS4 software was measured on a 7-point semantic differential scale. Software application was ranked by users from different aspects (i.e. overall software performance, screen, terminology and software feedback, learning and software capabilities).

TABLE 4
QUIS QUESTIONNAIRE RESULT ANALYSIS

Items	**Mean**	**Standard deviation**
Category: Overall software performance		
TERRIBLE-WONDERFUL	5.16	1.194
DIFFICULT-EASY	4.63	1.338
INEFFICIENT-EFFICIENT	5.34	1.125
UNFRIENDLY-FRIENDLY	4.69	1.469
FRUSTRATING-SATISFYING	4.56	1.413
INEFFECTIVE-EFFECTIVE	5.59	1.073
RIGID-FLEXIBLE	4.87	1.432
Category: Screen		
ONSCREEN INFORMATION (Inadequate-Adequate)	4.91	1.634
USER INTERFACE ARRANGEMENT (Not organized-Organized)	5.34	1.285
EASY TO FIND FUNCTIONS (Never-Always)	4.38	1.476
READING CHARACTERS	5.38	1.100

(Difficult-Easy)		
SCREEN BACKTRACK (Difficult-Easy)	5.81	1.355
CREATING NEW PROJECT (Confusing-Very clear)	6.19	1.091
TOOLBAR ACCESS (Difficult-Easy)	5.91	1.279
Category: Terminology and software feedback		
SIMPLE AND NATURAL DIALOGUE (Never-Always)	4.74	1.29
TERMS USED IN THE SOFTWARE (Inconsistent-Consistent)	5.5	1.28
POSITION OF WINDOWS DIALOG BOX (Inconsistent-Consistent)	5.31	1.281
INFORMS ABOUT WORK PROGRESS (Never-Always)	4.48	1.71
ERROR MESSAGES	3.97*	1.974
PROMPT FOR DOING SCRIPTING INPUT (Confusing-Clear)	3.35**	1.644
Category: Learning		
SOFTWARE LEARNING (Difficult-Easy)	4.47	1.704
EXPLORING BY TRIAL AND ERROR (Difficult-Easy)	3.65	1.704
REMEMBERING COMMANDS (Difficult-Easy)	4.13	1.561
PERFORMING TASKS IS SIMPLE (Never-Always)	4.50	1.191
HELP ACCESS OR DOCUMENT (Difficult-Easy)	4.25	1.8
HELP MESSAGES ON SCREEN (Unhelpful-Helpful)	4.31	1.966
Category: Software capabilities		
CORRECTING MISTAKES (Difficult-Easy)	4.13	2.060
DESIGNED FOR ALL LEVELS OF USERS (Never-Always)	3.19**	1.554
IMPORT AND EXPORT PROJECT IN AND OUT OF SOFTWARE (Difficult-Easy)	5.38	1.408
SOFTWARE RELIABILITY (Unreliable-Reliable)	5.37	1.273

**<3.5; *just above 3.5

The average user satisfaction for the overall software performance, screen, terminology and software feedback, learning and software capabilities are 4.99, 5.41, 4.72, 4.51 and 4.62 respectively. In general, the overall subjective user satisfaction for Adobe Flash CS4 scores above average of >3.5 for all the above-mentioned 5 categories. However, under the category of 'terminology and software feedback', 'prompt for doing scripting input (Confusing-Clear)' rates only 3.35 score, and designing for all levels of users accounts for 3.19, which is lower than the average acceptance level (3.5). This user feedback is consistent and proven by the CLI performance result of having more difficulties of performing Task 8 to 10 that requires prompts for doing scripting input.

V. DISCUSSIONS

This research measures the usability aspects in terms of user performance and satisfaction towards a graphical software application, Adobe Flash CS4 among user interface designers and software programmers. Users' performance in terms of effectiveness, efficiency, task duration, errors and number of help on working with GUI and CLI during the initial learning of Adobe Flash CS4 were investigated in the usability testing. It showed that users could easily pick up the interfaces' functionalities when some training was given. Tasks which needed to be carried out via GUI (Graphical User Interface) showed better results in terms of effectiveness, duration, efficiency rate and number of helps comparing to tasks via CLI (Command Line Interface). This result is consistent with Wiedenbeck et al. [21] considering Direct Manipulation Interface (DMI) as an easier to use as compared to menu-driven and CLIs. In addition to Shneiderman [5] claiming that users could track down information more quickly with GUI as compared to CLI. The result is also consistent with Gunderloy [22] stating that learning and using GUI software is easy and effortless, Schneiderman's [23] study on DMI interfaces and another investigation on menu-based interface benefits rather than command-based interface [24].

In terms of performance variables for the usability test of Adobe Flash CS4, tasks performed via CLI styles indicated very poor scores (in terms of effectiveness, number of errors and number of help) as opposed to the test result performed via GUI styles. It is consistent with Wisher et al. [33] claiming that failing to remember just one of the essential facts may leave some tasks unachievable. The finding of QUIS questionnaires showed that Adobe Flash CS4 in general gained high satisfaction scores in terms of overall software performance, terminology and software feedback, learning and software capabilities except 'prompt for doing scripting input' in CLI task. The application reaches its highest convenience in terms of screen due to the familiar and clear menu labels, ease of toolbar access, organized interface arrangement and ease of reading characters. Furthermore, it obtains its lowest satisfaction in terms of software learning due to the difficulty of exploring by trial and error, remembering commands and help material.

Having examined relationship of participants' background of study in Adobe Flash CS4 test, the results showed no significant difference for interface designers and programmers on performance variables. In other words, performing tasks in either GUI or CLI did not have significant difference among programmers and interface designers.

VI. CONCLUSION

The goal of this research was to compare the impacts of different interface styles (GUI vs. CLI) in a software application for developers and interface designers. CLI were found to be more difficult to learn and less ease of use, even for developers as well as designers. However, GUI was

perceived to be simpler to learn for both groups; however when it comes down to unfamiliar menu labels or icons difficult to find, users can easily make the mistake of selecting wrong menu items. Moreover, for procedural tasks with higher number of levels, users more likely forget the series of steps due to the dependency to recall issues. For the software application to be usable for end-users, employment of familiar menu labels and toolbars, less number of steps necessary for accomplishment of tasks, and providing GUI equivalent for CLI tasks are highly recommended. Besides experience of software will be more satisfying if it is designed for all levels of users. Utilizing commands without input prompting yields difficulties in usability of software. Nevertheless, we still witness development and implementation of many new software applications by reputable companies undergoing the same flaws prevalent in the past.

All in all, software applications are basic tools for developers and interface designers to create new software. Therefore, it is important to ensure usability of software applications in terms of meeting the needs of their users. Since user interface is the basis for all interactions between users and applications. Thus, it is important that software applications have to be usable and provide more enjoyable experience that put users in control of interface and reduce their memory load. Our Future work will touch upon a comparison analysis of retention study on the similar software, and compare with user prior knowledge and experience on learning a graphical software application.

ACKNOWLEDGMENT

We would like to thank all the participants that involved in this research study, particularly from Interface Design Department.

REFERENCES

[1] Jorgensen, A. H. and Myers, B. A. "User interface history". *Proceeding CHI '08 extended abstracts on Human factors in computing systems*, 2008, pp. 2415-2418.

[2] Frankish, C., Morgan, P. and Hull, R. "Recognition accuracy and usability of pen-based interfaces." IEE Savoy Place, vol. 126, pp. 7/1-7/6, 1996.

[3] Jacko, J. A., and Sears, A. *The Human-Computer interaction handbook: Fundamentals, evolving technologies and emerging applications (2nd Ed.).* New York: Taylor & Francis Group, 2002.

[4] International Business Machines (IBM). "The library for system solutions End User Interface reference", 1994 [Online]. Available: http://www.redbooks.ibm.com/redbooks/pdfs/gg244107.pdf [2011, May 13]

[5] Shneiderman, B. "Designing the user interface: strategies for effective human-computer-interaction". Reading, MA: Addison Wesley, 1998.

[6] Jamil, U., Mustafa, T., Sattar, A. R., Shafia, and Shahzad, F. "Cognitive analysis of software interfaces." *European Journal of Scientific Research,* vol. 1, pp. 99-108, 2010.

[7] ISO/IEC 9126-1. *Software engineering product quality-part 1: quality model (1st ed.).* USA: International Organization for Standardization, 2001.

[8] Nielsen, J. *Usability engineering.* Academic Press. Boston; London, 1993.

[9] Bevan, N., and Curson, I. "Methods for Measuring Usability." *Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction,* 672-673, 1997.

[10] Seffah, A., and Metzker, E. "The obstacles and myths of usability and software engineering." *Communications of the ACM*, vol. 47, pp. 71–76, 2004.

[11] Shneiderman, B., Plaisant, C. *Designing the user interface, strategies for effective human–computer interaction (4th ed.).* Reading, MA: Addison Wesley, 2004.

[12] Turban, E., McLean, E., & Wetherbe, J. *Information technology for management: Transforming business in the digital economy (3rd edition).* CA: John Wiley & Sons, 2002.

[13] Foviance. Glossary, 2010. [Online]. Available: http://www.foviance.com/glossary/u/user-interface-design/ [2010, December 20]

[14] Burgoon, J. K., Bonito, J. A., Bengtsson, B., Cederberg, C., Lundeberg, M., and Allspach, L. "Interactivity in human–computer interaction: a study of credibility, understanding, and influence". *Elsevier Science Publishers,* vol. 16, pp. 553-574, 2000.

[15] Oppermann, R. "User-interface design", 2002. [Online]. Available: http://fit.fraunhofer.de/~oppi/publications/UserInterfaceLearningSystems.pdf [2010, September 6]

[16] The Linux Information Project, "GUI Definition", 2004. [Online]. Available: http://www.linfo.org/gui.html [2010, September 6]

[17] Robertson, O., McCracken, D., and Newell, A. "The ZOG Approach to Man-machine Communication". *International Journal of Human-Computer Studies*, vol. 51, pp. 279-306, 1999.

[18] Soegaard, M. "Interaction styles", 2010. [Online].Available: http://www.interaction-design.org/encyclopedia/interactionstyles.html [2010, Novemeber 18] November 2010 from Interaction-Design.org.

[19] Higgins, D. "Widget", 2005. [Online]. Available: http://whatis.techtarget.com/definition/0,,sid9_gci213364,00.html [2010,September 6]

[20] Chin, J. P., Diehl, V. A., and Norman, K. L. "Development of an instrument measuring user satisfaction of the human-computer interface." In J. J. O'hare (Ed.), *CHI '88 Conference Proceedings: Human Factors in Computing Systems.* New York: Association for Computing Machinery, pp. 213-218, 1988.

[21] Wiedenbeck, S., and Davis, S. "The influence of interaction style and experience on user perceptions of software packages". *International Journal of Human–Computer Studies*, vol. 46, pp. 563–588, 1997.

[22] Gunderloy, M. *Developer to designer: GUI design for the busy developer,* Alameda, CA: Joel Fugazzotto, 2005.

[23] Shneiderman, B. "Direct manipulation: A step beyond programming languages". *Computer,* vol. 16 , pp. 57-69, 1983.

[24] Hasan, B., and Ahmed, M. U. "Effects of interface style on user perceptions and behavioral intention to use computer systems". *Computers in Human Behavior*, vol. 23, pp. 3025–3037, 2007.

[25] McCall, J. A., Richards, P. K. and Walters, G. F. "Factors in Software Quality". Springfield, VA, National Technical Information Service, 1977.

[26] Zhou, P. and Fang, X. "Analysis of Cognitive Behavior in Software Interactive Interface". IEEE, 2008, pp. 113-116.

[27] Raza, A., Capretz, L. F. and Ahmed, F. "Users' perception of open source usability: an empirical study". Engineering with Computers, vol, 28, 2011, pp. 109-121.

[28] IEEE Std. 1061. "IEEE standard for a software quality metrics methodology*"*. New York, IEEE Computer Society Press, 1992.

[29] ISO/DIS 9241-11. "Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on usability", 1998.

[30] Ajayi, A. O., Olajubu, E. A., Ninan, D. F., Akinboro, S. A., and Soriyan, H. A. "Development and Testing of a Graphical FORTRAN Learning Tool for Novice Programmers". *Interdisciplinary Journal of Information, Knowledge, and Management, 5*, 277-289,2010.

[31] Davis, F. D. "User acceptance of information technology: System characteristics, user perceptions and behavioral impacts". *International Journal of Man–Machine Studies, 38*(3), pp. 457–487,1993.

[32] Davis, S. A., and Bostrom, R. P. "An experimental investigation of the roles of the computer interface and individual characteristics in learning of computer systems". *International Journal of Human–Computer Interaction, 4*, pp. 143–172,1992.

[33] Wisher, R. A., Sabol, M. A., and Ellis, J. A. "*Staying sharp: Retention of military knowledge and skills* (ARI Special Report 39)". Alexandria, VA: U.S. Army Research Institute for the Social and Behavioral Sciences, 1999.