Usability and Information access challenges in complex simulation models

Erick Dube[1] and Shahen Naidoo[2]
Email addresses

[1]School of Computer Science, Univeristy of KwaZulu Natal
Durban 4001, South Africa
Dubee2@ukzn.ac.za

[2]Defence Peace Safety & Security (DPSS), Council for Scientific and Industrial Research
Pretoria, South Africa
SNaidoo3@csir.co.za

ABSTRACT

There has been a paradigm shift in recent years from efficient techniques of accessing data to effective information access. The challenge facing researchers working in information access is how to help users easily locate the information needed. One of the most affected groups of software classes are simulation modeling tools. This problem is further compounded by the fact that the model developers are not necessarily the model users and as such can hardly perceive how the users will interact and react to the operational model. Developers have in mind effectiveness, efficiency, and user satisfaction when developing the software and seldom address human–computer interaction. This means that very good software models then fail to deliver to their capacities due to challenges in the complexity of the model interfaces. In this paper we review the information access challenges by applying the software usability metrics to operational simulation tool currently being used and constantly being modified at the DPSS for simulating defence systems.

KEY WORDS

Software, usability, quality metrics, modelling, simulation

1. Introduction

The purpose of using simulation to model defence operations is to obtain insights into the consequences of using different techniques and strategies and thus helping the planner in making the most advantageous decisions. However the lack of information as presented by the user interface may result in some operations not being visible to the user although available in the simulation tool. The challenge has been to develop simulation tools that can be used by the users with minimal effort required both to operate the tool and interpret the result of an experiment. A systems modeling and simulation team at the Council for Scientific and Industrial Research is involved in the development of a number of Defense System simulation tools. We will apply software usability metrics to evaluate the information access challenges imposed by use of one such tool.

The Ground Based Air Defense System (GBADS) acquisition program of the South African Army is making use of a systems modelling and simulation capability for acquisition decision support. The Virtual GBADS Demonstrator (VGD) was developed for this purpose VGD is an in-house suite of software that provides for the deployment, simulation and analysis of virtual entities within a defined scenario to observe the behavior and interaction between the various operators and their related subsystems within a GBADS deployment. The VGD architecture supports the distributed simulation of many-on-many engagements. The behaviour of equipment and operators is modeled, as well as the interaction between these entities.

In this paper we discuss the various interfaces of the VGD with respect to the usability of the software and subsequent information access challenges. Our approach will include a careful application of usability metrics to expose the software usability challenges and opportunities thus establish how best information could be relayed with minimal effort of both use and understanding.

2. Application (model) description

VGD can function as both a virtual and constructive simulator. For virtual simulations Operator in the Loop (OIL) consoles allow human operators, from battery-level to detachment-level, to interact with the real-time simulation in order to evaluate various doctrinal concepts from within the virtual environment. An example of the Battery Fire Controller Human Machine Interface (HMI) is shown in Figure 1. Constructive simulation is used for statistical analysis and evaluation of emergent behavior at battery level.

In real-time mode the architecture supports the integration of VGD with operational systems via gateways (shown in

Figure 1) allowing live data to be imported into the simulation. It thus allows a virtual environment to be integrated within a live exercise.

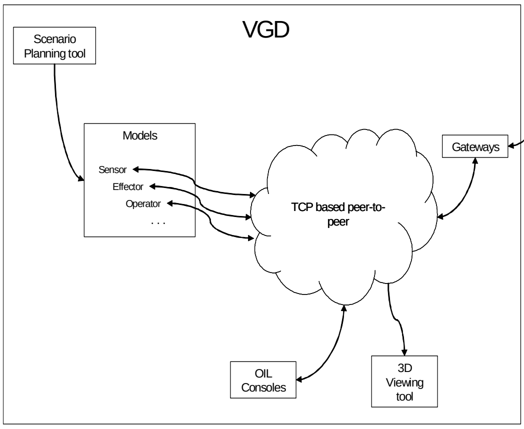


Figure 1 Architectural Overview of VGD

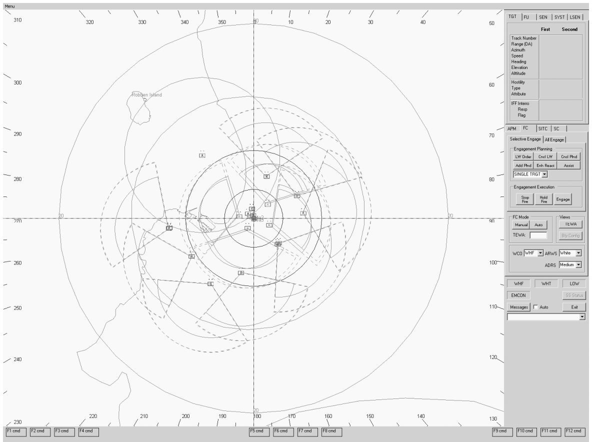


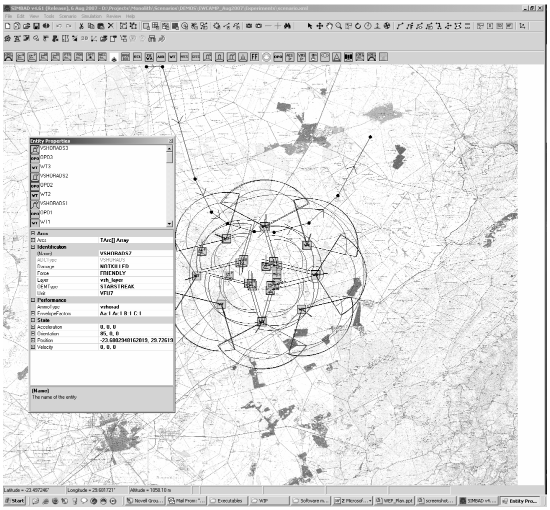
Figure 2 Screenshot of an OIL Console

A scenario-planning tool is used to create scenarios consisting of each of the GBAD entities to be simulated. An example deployment of a battery is shown in Figure 3

3D visualization is used for real-time viewing of a scenario, as well as after action review. 3D visualization, specifically for complex scenarios allows for the detailed analysis of events and object interaction. A 3D view of a threat scenario is shown in Figure 4

2.1 Scenario Planning

Any scenario to be simulated must first be planned and configured using the scenario planning tool Fig 3. A scenario will consist of various entities that will be used to create instances of the models that must run within the simulation environment.



Arcs	
Arcs	TArc[] Array
Identification	
(Name)	VSHORADS7
ADCType	VSHORADS
Damage	NOTKILLED
Force	FRIENDLY
Layer	vsh_layer
OEMType	STARSTREAK
Unit	VFU7
Performance	
AmmoType	vshorad
EnvelopeFactors	Aa:1 Ar:1 B:1 C:1
State	
Acceleration	0, 0, 0
Orientation	85, 0, 0
Position	-23.6802948162019, 29.72619
Velocity	0, 0, 0

(Name)
The name of the entity

Latitude = -23.497246°
Longitude = 29.681721°
Altitude = 1058.10 m

Figure 3 Screenshot of the Scenario-planning tool

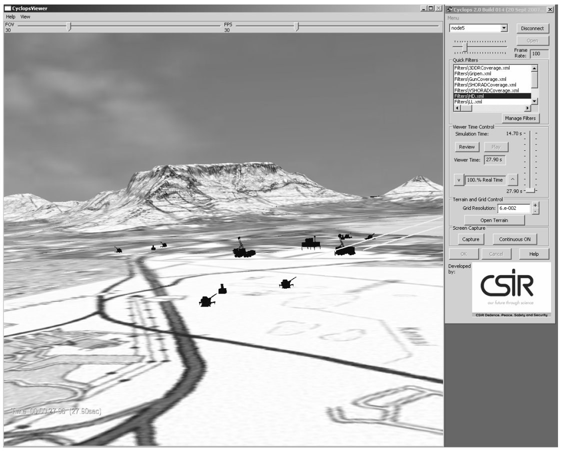


Figure 4 Screenshot of 3D visualization tool

It is therefore necessary to also include the various parameters that are required to create each instance of a model within the simulation environment. A scenario planning tool was developed to perform this function of configuring scenarios. It consists of a graphical interface that allows the user to drag and drop entities onto a map position and configure their parameters. The scenario planning tool then saves the configured scenario as a file using an XML format to store the data. Because VGD uses a distributed simulation architecture, the tool also allows the user to specify where each model will run. In this case the user will, in addition to the scenario, also specify a list of IP addresses and the models that must run on each PC. Some models may require specific parameters that are saved in separate configuration files. These files reside in specific areas within the simulation directory structure. One example of this is the terrain configuration files that specify the type of terrain data to be used. This terrain configuration file must be set up in order for the simulation to run and is located in the same folder as the scenario configuration. Another example would be a configuration file that determines the behaviour of a radar system and is located in a folder designated to contain configuration files.

2.2 Running the Simulation

Once a simulation is configured, it may be run within the simulation environment. This is done by invoking the executable from the command line. Various command line parameters can be used to control a simulation run. Presently 53 command line parameters exist and can be used to control various aspects of a simulation run including the scenario being run, how fast it runs, what data gets logged with each run, random number seeds, and various other parameters.

In addition to the command line parameters, run time keyboard hits can also control a simulation run. The run time keyboard hits control the simulation in terms of stopping, pausing and restarting the simulation. The speed of the simulation run (real time or "as fast as possible") can also be controlled at run time.

2.3 Operator Consoles

The simulation is used to simulate an environment that includes human operators. Therefore, the option exists to either simulate the operators (constructive simulation) or have an actual operator interact with the other models within the simulation environment (virtual simulation) via "operator in the loop" terminals. These terminals can run on any PC on the network and must be set up to connect to the PC actually running the simulation model. Although the terminals are simply interfaces that allow an operator to become immersed within the simulation environment, they do require some setup before they can be used effectively. The scenario that is being run by the simulation must also be loaded into the terminals. Before interacting with the simulation, the terminals must be set to manual mode. This can be done at the terminal during run time, or set up as a default by using the appropriate command line parameter when starting the simulation. The terminals are in fact designed to simulate the actual terminals used in the field, and as such the operator's interaction with the simulation follows a specified operating procedure.

3. Software usability metrics

Despite the widespread use of simulation systems and the considerable investment in purchasing or developing them in house, there is no consensus on devising a standard framework for evaluating system usability in this area. We have used the tool described above in 2 to illustrate that it is no longer sufficient to deliver simulation products that have technical excellence but products also need to be easy to use and fit in the work practices and activities of the client and professional users.

The lack of an agreed simulation system quality model is in stark contrast to the extensive work on software quality assurance in general [4]. This paper proposes the ISO 9126 Quality Model [10] as a useful tool for evaluating the usability of simulation modelling tools. Our primary goal is to define the quality metrics with respect to the usability of the software. We take the assumption that all the other five categories: functionality, reliability, efficiency, maintainability, and portability requirements [9][10] have been adequately addressed in the design and implementation of VGD.

Usability is a subset of software quality characteristics. It is an important characteristic in determining the quality of the product, as such, a common approach in measuring usability and other quality measures should be investigated. Usability engineering is concerned with developing interfaces that people can use efficiently and effectively. It deals with issues such as system learnability, efficiency, memorability, errors and user satisfaction [3][5].

3.1 Defining usability measures

The term usability has been used with different meanings, making it a very confusing concept, especially for software developers. Usability refers to both a set of independent quality attributes such as user performance, satisfaction, and learnability, or all at once [2]. As a software quality attribute, usability has not been defined consistently by either the researchers and the standardization organizations or the software development industry. The following definitions illustrate how the term usability has been perceived differently in three distinct standards:

- “The capability of the software product to be understood, learned, used, and attractive to the user, when used under specified conditions.” [10]
- 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.” [8]
- The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component.” [7]

With all these possible definitions, it is very difficult to specify precisely the measurable usability attributes and their interpretations from different perspectives. However we shall adopt convenience as described in [9], being the degree to which accessing and using an interface is comfortable, and possible without excessive effort, mental or physical. The lack of deterrents to use, including organizational and social deterrents, schedule constraints, system availability, learning threshold, system delays, and prerequisites for use.

There is comparatively little information about exactly how to select a set of usability factors or metrics considering things such as management objectives, business goals, competition, economics, or resource limitations on product development. However based on our review of the usability measurement literature, we selected 7 usability factors that we will briefly describe and use to evaluate the simulation tool (VGD).

Efficiency, or the capability of the software product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use.

Effectiveness, or the capability of the software product to enable users to achieve specified tasks with accuracy and completeness.

Productivity, which is the level of effectiveness achieved in relation to the resources (i.e. time to complete tasks, user efforts, materials or financial cost of usage) consumed by the users and the system. In contrast with efficiency, productivity concerns the amount of useful output that is obtained from user interaction with the software product

Satisfaction, In [9], “satisfaction” implies “the capability of the software product to satisfy users in a specified context of use.” Satisfaction in that sense refers to the user’s response to interaction with the product. It includes judgments about product use rather than about properties of the software itself [6].

Learnability, The ease with which the features required for achieving particular goals can be mastered. It is the capability of the software product to enable users to feel that they can productively use the software product right away and then quickly learn other [1] new (for them) functionalities

Usefulness, or whether a software product enables users to solve real problems in an acceptable way. Usefulness implies that a software product has practical utility, which in part reflects how closely the product supports the user’s own task model. Usefulness obviously depends on the features and functionality offered by the software product. It also reflects the knowledge and skill level of the users while performing some task (i.e., not just the software product is considered).

Comprehension, Grasping (understanding) the meaning of informational materials. This may involve getting insight about data relative to some task. This usually requires finding some representation (schema) for the data that is efficient for the task.

[1] suggests breaking down the factors into measurable criteria (sub-factors). A criterion is directly measurable via at least one specific metric. Presented in Table 1 are definitions of the 16 criteria envisaged to be useful for this study. These definitions all assume a particular context of use or stated conditions for a simulation software feature.

Table 1 Usability criteria

Measurement Criteria	**Description**
User Guidance	Whether the user interface provides context-sensitive help and meaningful feedback when errors occur
Legibility	Ease with which visual content can be understood
Task Completion	Whether a user can complete a task within appropriate task time when performing.
Minimal action	Capability of the software product to help users achieve their tasks in a minimum number of steps
Minimal memory load	Whether a user is required to keep minimal amount of information in mind in order to achieve a specified task
Attractiveness	Capability of the software product to be attractive to the user (e.g. through use of colour or graphics design; [9])
Familiarity	Whether the user interface offers recognizable elements and interactions that can be understood by the user
Fault tolerance	Capability of the software product to maintain a specified level of performance in cases of software faults or of infringement of its specified interface [9]
Navigability	Whether users can move around in the application in an efficient way
Simplicity	Whether extraneous elements are eliminated from the user interface
Load time	How fast it responds to the user
Self-descriptiveness	Capability of the software product to convey its purpose and give clear user assistance in its operation
Consistency	Degree of uniformity among elements of user interface and whether they offer meaningful metaphors to users
Effort	Amount of mental or physical effort necessary to operate and control a software product
Responsiveness (delays)	How fast it responds to the user
Controllability	Whether users feel they are in control of the software product

4 Applying usability measures to Simulations Models

Effectiveness of a simulation system can be hindered if there are: defects in navigation through the system, problems in screen design and layout, inappropriate terminology, inappropriate feedback or complete lack of feedback, problems with modality, inconsequential redundancies, and problems in matching with user tasks. Learnability can be impeded if there are: defects in navigation, problems in screen design and layout, inappropriate terminology, inappropriate feedback or complete lack of feedback, and problems in matching with user tasks. Flexibility is impeded if there is no user control over the system and if the system imposes the order of the steps in a task. User attitudes towards the system can be seriously affected by any of the above usability defects.

Computer-based simulation modeling is one of the domains that is particularly demanding in terms of user interfaces. Issues that influence the ‘usability’ of such systems include data input interfaces, model visualisation tools and simulation results analysis tools. In these three key areas there is need to use the measurement criterion described in Table 1 to demonstrate the usability of the simulation.

Data Input

It is apparent that the data input part of the system is considered as less important than, for example, the visual simulation part. Most of the papers on simulation systems only briefly mention the data input capabilities of systems, if at all. However, there is room for a great deal of improvement in the domain of data input and/or model specification that would improve existing simulation systems.

Visual Simulation

Visual programming tools are standard features in all visual interactive simulation (VIS) systems, and drawing tools are very common. Dynamic icons and animation are supported by most visual simulation systems. The interactive change of the simulation parameters and of the speed of animation whilst the simulation is being executed, are also often provided. Panning and zooming is another quite common facility. Most of the current simulation systems have some form of visual animation of a simulation run.

Simulation results

Information visualization exploits the phenomenal abilities of human perception to identify structures by presenting abstract data visually, allowing an intuitive exploration of data to get insight, to draw conclusions and to interact directly with the data. A variety of interactive information visualization techniques have to be provided together with appropriate pre-processing techniques to handle huge amounts of data.

Using the measurement criteria we suggest a qualitative point system in scoring the impact of the criteria on the three main usability concerns of simulation modeling applications. The scores shall be Low, Medium and High, High being the highest score showing the strength of the requirement of the criteria as shown in Table 2.

Table 2 Measurement criteria for simulation modeling

Measurement Criteria	**Data Input**	**Visual Simulation**	**Simulation results**
User Guidance	High	Medium	Medium
Legibility	Medium	High	High
Task Completion	Low	Low	High
Minimal action	High	Low	Low
Minimal memory load	High	Low	Low
Attractiveness	High	High	High
Familiarity	High	Medium	High
Fault tolerance	High	Medium	Low
Navigability	High	High	High
Simplicity	High	High	High
Load time	Low	High	High
Self-descriptiveness	High	High	High
Consistency	High	High	High
Effort (minimal)	High	Medium	High
Responsiveness (delays)	Medium	High	Medium
Controllability	High	High	Low

Table 3 Relationship between factors and criteria [1]

Measurement Criteria	Efficiency	Effective	Productive	Satisfaction	Learnable	Usefulness	Comprehension
User Guidance				+	+		+
Legibility			+	+	+		
Task Completion	+	+	+				
Minimal action	+			+	+		
Minimal memory load	+			+	+	+	+
Attractiveness				+			
Familiarity					+		
Fault tolerance						+	
Navigability							+
Simplicity					+		
Load time	+		+			+	
Self-descriptiveness					+		+
Consistency		+			+		
Effort (minimal)	+		+	+	+		+
Responsiveness (delays)	+		+			+	
Controllability							+

5 Applying Usability measures to VGD

We use the measurement criteria of table 2 to investigate the usability of VGD and the results are demonstrated in table 4

Table 4 VGD usability rating

Measurement Criteria	**Data Input**	**Visual Simulation**	**Simulation results**
User Guidance	Medium	Low	Low
Legibility	Medium	Medium	Low
Task Completion	Low	Low	High
Minimal action	Low	Low	Low
Minimal memory load	Low	Low	Low
Attractiveness	Low	High	Medium
Familiarity	High	Medium	High
Fault tolerance	Low	Medium	Low
Navigability	High	High	High
Simplicity	Medium	High	Medium
Load time	Low	High	High
Self-descriptiveness	High	Low	Low
Consistency	High	High	High
Effort (minimal)	Low	Medium	Medium
Responsiveness (delays)	Medium	High	Medium
Controllability	Low	High	Low

From the results in table 4 it is quite clear that the complexity of the software makes it difficult for the developers to focus on its usability. Using simple graphs Figure 5, Figure 6, Figure 7 to show the difference between the expected score and the one from VGD we expose the areas needing usability improvement and those that conform to the current standards. Scores are on the vertical axis and are a number representation of the previous notation of Low, Medium and High with 1 representing Low, 2 Medium and 3 High. The x-axis represents the measurement criteria from 1 to 16, with 1 being user guidance and 16 being Controllability. The doted line shows the expected score while the solid line represents the VGD score where there is no disparity, the dotted line seems to disappear. VGD seems to score quite low on data input and visual simulation seems to perform better.

The fact that there is minimal user guidance on the display of simulation results can be closely linked to the fact that the results are displayed in an environment familiar to the users. The number of input parameters reduces the efficiency of the VGD but the developers argue that its makes the software very flexible and increases its usefulness by virtue of allowing multiple scenario including invalid ones. However the number of input parameters (53) one requires to know to be able to run the software increases the minimal memory load, minimal action and thus reduces its learnability

The lack of self-descriptiveness on the simulation results follows the same argument as that of user guidance,

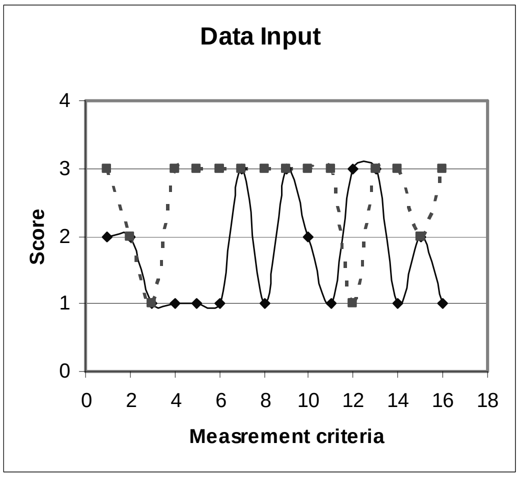


Figure 5 Data Input

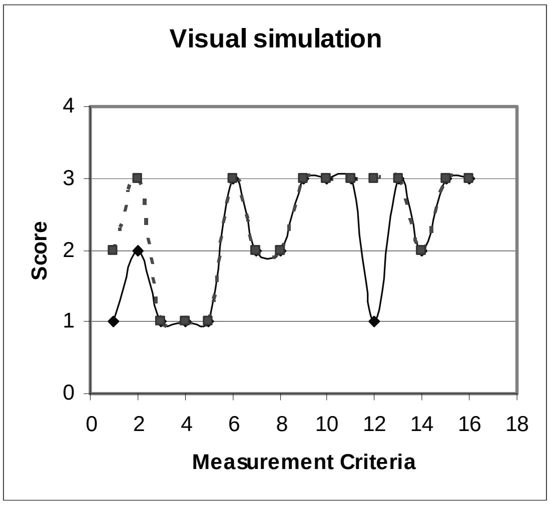


Figure 6 Visual Simulation

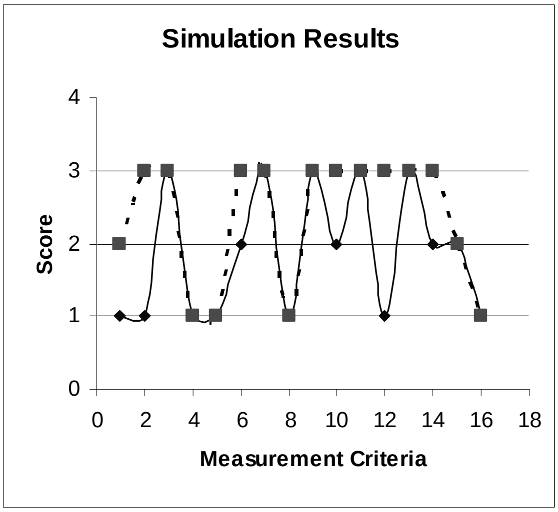


Figure 7 Simulation results

however this has a negative impact in that some relevant and useful information displays are hidden unless one is very familiar with the tool navigating through the results could be quite complex leaving out some details which are useful to the decision making. This is mainly due the limitation of the user interface only being able to display a limited number of legible data points and icons. Thus it would require more effort for one to go through results.

The large disparity between graph of VGD and that of the expected as depicted in Figure 5 goes on to confirm that developers paid the little attention to the human–computer interaction. While the low degree of deviation depicted in Figure 6 and Figure 7 confirm that developers have made every effort to address effectiveness, efficiency, and user satisfaction.

6 Conclusion

With increased application of simulation applications and the critical risks due to the low software usability, the importance of high information access has been getting more important than ever. Although the list of usability as well as the related factors, criteria, and metrics is not consistently defined across different standards or models, we have managed to derive from existing literature a simple a meaningful mechanism of measuring simulation software usability. By applying the measurement criteria missing usability features have been exposed and complex information access scenarios have also been identified.

The lack of some usability features has a direct impact on the ease with which information can be accessed in simulation models. Simulation model results by nature are difficult to interpret and any deficiency in the usability deters the user from the application. However the temptation of having as much information on a single user interface screen seems to be reducing the information relayed to the user as he/she is limited as to the number of points on screen that he/she can simultaneously access and comprehend. The lack of comprehension by user resulting directly from difficult user interface breeds itself into an information access challenge.

7 References

[1] Ahmed Seffah · Mohammad Donyaee · Rex B. Kline · Harkirat K. Padda Usability measurement and metrics:A consolidated model, *Software Qual Journal* 14, 2006,: 159–178

[2] Bevan N. and Azuma, M. Quality in use: Incorporating human factors into the software engineering life cycle. *In Proceedings of the 3rd IEEE International Software Engineering Standards Symposium and Forum,* 1(997).

[3] Cooper, A., and Reimann, R., *About Face 2.0: The Essentials of Interaction Design*, Wiley Publishing Inc., Indianapolis, IN, (2003).

[4] Crosby, P.B. Quality *Is Free: The Art of Making Quality Certain*. McGraw-Hill, New York, (1979).

[5] Hix, D., and Hartson, H. R., *Developing user Interfaces: Ensuring Usability through Product and Process*, John Wiley & Sons, Inc., New York, NY, 1993.

[6] Ho-Won Jung, Seung-Gweon Kim, Chang-Shin Chung, "Measuring Software Product Quality: A Survey of ISO/IEC 9126," *IEEE Software*, *21*(5), 2004,. 88-92.

[7] IEEE Std. 1061. Software Quality Metrics Methodology, 1998.

[8] ISO/DIS 9241-11. Guidance on Usability. Ergonomic Requirements for Office Work with Visual Display Terminals (VDT). 1996.

[9] ISO/IEC TR 9126-2:2003 Software engineering -- Product quality -- Part 2: External metrics.

[10] ISO 9126. Software Product Evaluation: Quality Characteristics and Guidelines for their Use. 1991