Quality Principles for the Ergonomics of Human-Computer Interfaces of Modeling and Simulation Software

Tuncer Ören
M&SNet: OC-MISS (Ottawa Center of the McLeod Institute of Simulation Sciences)
School of Information Technology and Engineering (SITE), University of Ottawa
Ottawa, ON, Canada
oren@site.uottawa.ca

Levent Yilmaz
M&SNet: AMSL (The Auburn Modeling and Simulation Laboratory)
Computer Science and Engineering
Auburn University
Auburn, AL, USA
yilmaz@eng.auburn.edu

Abstract

This article points out that while quality objectives such as testability and reusability have been widely investigated and disseminated for modeling and simulation (M&S) software, there are few systematic studies on the assessment of the ergonomics of the Human-Computer Interfaces (HCIs) for M&S software and environments. This article presents 21 quality principles for the HCIs for M&S software in four broad categories which are usability, communicativeness, reliability and evolvability. The principles can be used as a basis for developing quality criteria, metrics, and guidelines for human-computer interfaces of modeling and simulation software. They can be used for proper design of new interfaces, or to improve, evaluate, and compare existing ones.

1. INTRODUCTION

The availability of affordable hardware and software packages hosted within commercial, off-the-shelf PC platforms, is stimulating steady growth in several areas including the adoption of Virtual Reality or Synthetic Environment for defense training applications. Attention is now turning to more human-centered issues, focusing on pragmatic means of defining training content, sensory fidelity, interaction style and the need for specialized display and control peripherals (ErgonomicsSociety 2004).

Simulation model quality and credibility assessment has long been a significant area of interest within the simulation modeling community. While quality objectives such as testability and reusability have been widely investigated and disseminated for M&S software, there are few systematic studies on the assessment of the ergonomics of the Human-Computer Interfaces for M&S software and environments. With modern computer simulation technology, ergonomic questions are being raised during development of new products. Simulation is now being widely used to study ergonomic problems (DigitalHumanSim 2004, RAMSIS 2004, Walter 2003).

1.1 Interactive M&S Software

The term M&S software denotes both software to represent executable version of the computerization of a simulation study and simulation environment. The later denotes software for the specification of a simulation study and generation of the executable simulation program.

As depicted in Figure 1, a human/computer interface in an M&S software consist of two parts: front-end and back-end interfaces (Ören and Cetin, 1988). The study of their functionalities and requirements for ergonomics of M&S software needs to be done to explore their quality requirements systematically. In this article, due to space limitation, only highlights are given.

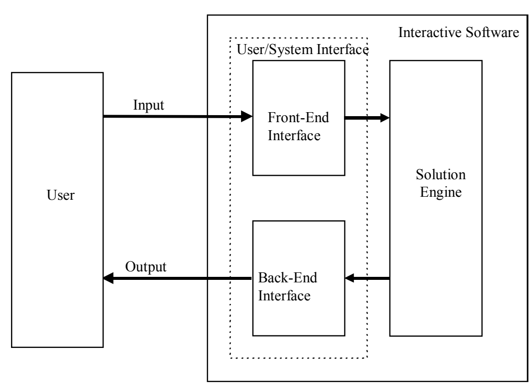

Figure 1. The elements of interactive simulation software

1.2 Front-end Functionalities of M&S Software

The *front-end interface* is used to monitor and control access to the simulation system as well as for the entry of data, requirement, and/or query. Depending on the sophistication of the M&S software, the front-end interface may also include a problem specification /formulation as well as built-in quality assurance modules to check acceptability of the inputs. Some other possibilities for front-end interface are deictic and verbal inputs.

1.3 Back-end Functionalities of M&S Software

The *back-end interface* is used to process and display model behavior. Depending on the M&S system, back-end interface may include functionalities for virtual and augmented reality, interpretation, and explanation of alternatives. Such functionality can also be used for real-time validation of online and symbiotic simulations against system behavior. Intelligent interfaces are well accepted in software engineering in general and M&S has solid methodological possibilities to incorporate them. Artificial intelligence and agent-technology offer several functionalities for human/computer interfaces called AI-supported or agent-supported simulation.

1.4 Quality Principles for HCIs

Several authors and institutions developed quality principles for HCIs for software in general. For example, Shneiderman's three principles (Shneiderman, 1998) consist of the following: (1) Recognize the Diversity (usage profiles, task profiles, and interaction styles). (2) Use the Eight Golden Rules of Interface Design and (3) Prevent Errors (correct matching pairs, complete sequences, and correct commands)

His eight golden rules of interface design are

1 Strive for consistency.
2 Enable frequent users to use shortcuts.
3 Offer informative feedback..
4 Design dialog to yield closure.
5 Offer simple error handling.
6 Permit easy reversal of actions.
7 Support internal locus of control..
8 Reduce short-term memory load.

Mayhew's "General Principles of User Interface Design" are (Mayhew, 1992):

1 User compatibility
2 Product compatibility
3 Task compatibility
4 Work flow compatibility
5 Consistency
6 Familiarity
7 Simplicity
8 Direct manipulation
9 Control
10 WYSIWYG
11 Flexibility
12 Responsiveness
13 Invisible technology
14 Robustness
15 Protection
16 Ease of learning and ease of use

IBM's "Design Principles for Tomorrow" (IBM) are:

1 Simplicity: Don't compromise usability for function
2 Support: Place the user in control and provide proactive assistance
3 Familiarity: Build on users' prior knowledge
4 Obviousness: Make objects and their controls visible and intuitive
5 Encouragement: Make actions predictable and reversible
6 Satisfaction: Create a feeling of progress and achievement
7 Availability: Make all objects available at all times
8 Safety: Keep the user out of trouble
9 Versatility: Support alternate interaction techniques
10 Personalization: Allow users to customize
11 Affinity: Bring objects to life through good visual design

The following visual design principles that promote clarity and visual simplicity in the interface are recommended by IBM:

1. Subtractive design (to reduce clutter)
2. Visual hierarchy (of users' tasks)
3. Affordance (to determine the actions that should be taken with an object) and
4. Visual scheme (that maps to the user model and lets the user customize the interface)

1.5 Need for Quality Principles for HCIs of M&S Software

Adopting the quality principles developed for the HCIs of software in general is not sufficient. In this article, the aim is to specify quality principles for the ergonomics of M&S software by taking into account the front-end and back-end functionalities of M&S software. For this purpose, 21 interface quality principles are presented in a systematic way in four broad categories which are usability, communicativeness, reliability, and evolvability.

The principles can be a basis for the development of HCI quality criteria, metrics, and guidelines for M&S software and environments. They can be used for evaluation and comparison of existing interfaces. They can also be used as a systematic guide for the design and implementation decisions of new interfaces.

2. QUALITY PRINCIPLES FOR HCIs of M&S SOFTWARE

The 21 principles for M&S HCIs are grouped in four areas, which are usability, communicativeness, reliability, and evolvability.

2.1 Usability Principles

Advantages of satisfying usability principles include the use of the computer (or more specifically, the software) to support minimalist approach to documentation, display of necessary information on the screen when needed to reduce memory load of the user, and the use of the terminology of the application area.
As represented in Table 1, usability principles can be related with users or problems. Usability principles related with users include least training, minimum memory load, simplicity, and familiarity principles. Those related with problems include separation of concerns and functionality.

Principle of **least training** asserts that an interface should require least amount of training. Any needed knowledge should be available as just-in-time learning facility. It is also highly desirable to have a self-paced demo on the utilization of the system. For example, the intricacies of the GPSS language are clearly indicated for the benefit of the user by Schriber (1991). However, as indicated by Ören (1991), these recommendations should be part of the specification functionality of the front-end interface of an environment freeing the user from the syntactic details of a language to concentrate on problem oriented aspects.

According to the principle of **minimum memory load**, the users should not be obliged to remember information from one part of the interface to be used in another part. Users should not be obliged to memorize instructions or series of operations. Instructions to use the system should be visible (for example, through icons or pull down or pop up menus). If there is a sequence of activities to perform a task they should be performed for the user or at least the sequence should be made clear to the user. For complex and/or routine tasks, software agents can and should be used to alleviate the workload of the user (Bradshaw 1997).

Table 1: Quality Principles for HCIs of M&S Software – Usability Principles

Usability principles	
- related with users	
Least training	An interface should require lest amount of training.
Minimum memory load	Users should not be obliged to remember information from one part of the interface to be used in another part.
Simplicity	An interface should not have *distractive information*, should be *uniform*, *unambiguous* and should allow *easy navigation*. - *Unambiguity*: There should be no ambiguity to initiate an action in different parts of the interface. - *Navigability*: Activities should be initiated as directly as possible.
Familiarity	An interface should be based on the familiarity of its users to the *language*, *terminology*, *metaphor,* and *inputs*. - The *natural language* used in an interface should not hinder the proper use of the system. - The *terminology* used in the interface should be based on the application domain's terminology. - The *metaphor* used in the interface should be most appropriate (i.e., natural) for the application domain. - A *front-end interface* should be capable to accept the types of inputs most appropriate (i.e., natural) for the application.
- related with problems	
Separation of concerns	An interface should allow focusing on different aspects of M&S.
Functionality	An interface should offer complete set of abilities to specify problems and to process, analyze, and present results. (Functionality needs to be balanced with simplicity.)

Principle of **simplicity** requires that an interface should not to have distractive information and should be uniform. *Uniformity* is needed to avoid ambiguity in initiating an action in different parts of the interface. As such, the displays should be as uniform as possible and should not be context-dependent.

Good *navigability* requires that the activities should be initiated as directly as possible. Navigation should be done with least movements. At every state of the system, the user should know: (1) how to exit from the current state, (2) how to exit the system, (3) the list of activities that are initiatable from that state, and (4) how to initiate them.

Principle of **familiarity** requires that interfaces should be based on the familiarity of the user with the natural language, terminology, metaphor, and inputs.

The *natural language* used in an interface should not hinder the proper use of the system. Ideally, the interface may be at least bilingual: an international language and the native language of the user, if it is not the international language. Instead of being hardwired, the natural language would be switchable. By providing a mechanism, by which families of related or dependent interface objects can be created, the system can be configured with alternative presentation without specifying or hardwiring a concrete representation.

The *terminology* used in the interface should be based on the application domain's terminology. Furthermore the terms should be unequivocal. Computer jargon should be avoided.

The *metaphor* used in the interface should be most appropriate (i.e., natural) for the application domain. Examples of metaphors are: desktop, book, index card, form, calendar, agenda, instrument panel, warning or traffic lights, map, office, supermarket, layout, (factory, theater, airplane). A door is an example of an element of a 3-D metaphor in virtual reality. Visual specification of models using front-end interface facilities implies a high-level specification language for the internal representation of the specifications.

A *front-end interface* should be capable to accept the types of inputs most appropriate (i.e., natural) for the application. Examples (conventional): keyboards, pointing devices (several types of mouse, (such as wired, wireless, optical), trackball, joystick), touch screen, touch pens. Examples (relatively new types): handwriting, data glove, deictic input (gestures), haptic inputs (touch, pressure), eye gaze tracking, speech or voice, and multimodel input. Deictic inputs provide flexibility in virtual and/or augmented reality. Eye gaze tracking is important in civilian as well as defense applications. Interactive voice technology, after maturing, can be the basis for applications and/or for operating systems; thus allowing voice command.

Principle of **separation of concerns** or focusing on different aspects of M&S implies separation of specifications of model, model parameters, experiments, as well as model behavior generation, processing, and display. A further step in separation of concerns (or aspect-orientation in M&S) may be realized by the help of interfaces.

As an example, it was shown that existing simulation software can be pre-processed to add virtual gauges. A program generator can then generate an augmented version of the source program leaving the source program intact. Virtual gauges of the augmented program can display the values of selected variables during run time. In addition to the displays, values of the variables can be monitored with respect to lower and upper thresholds to set warnings and alarms. All these functionalities can be realized without rewriting the source program and the shape and location of the virtual instruments can easily be modified (Abdullah and Ören 1997).

Principle of **functionality** requires an interface to offer complete set of abilities to specify problems and to process, analyze, and present results. Therefore, the *input* functionalities can be as advanced as the computer-aided problem solving environments. The *output* functionalities can be graphically oriented as it is in the case of virtual or augmented realities. They can also include statistical and/or analytical data compression abilities as well as reasoning abilities to present and/or explain results. However, a balanced approach to functionality and simplicity is desirable.

2.2 Communicativeness Principles

Advantages of satisfying communicativeness criteria include the possibility of visualization of the behavior of the simulation, obtaining information about the simulation via monitoring its executions, and the viability the following categories:

As represented in Table 2, communicativeness principles can be related with users, formulation and solving problems, or with displays. Communicativeness principle related with users is the principle of restrained relationship with users. Those related with formulation and solving problems include informativeness, perceptiveness, and explanation ability principles. Those related with display include expressiveness and aesthetic/cultural acceptance principles.

According to the principle of **restrained relationship with users**, patronizing, informal, and insulting tone should not be used. Human-like entities (including avatars) should be used when warranted and not just as technological curiosities.

Table 2: Quality Principles for HCIs of M&S Software – Communicativeness Principles

Communicativeness principles	
***- related* with users**	
Restrained relationship with users	Patronizing, informal, and insulting tone should not be used.
- related with formulation and solving problems	
Informativeness	An interface should be able to prompt several types of knowledge which may (or should) exist in the system.
Perceptiveness	An interface should be able to observe the user to perceive the intentions and/or to decide when to initiate an advice.
Explanation ability	An interface should be able to justify its decisions and explain the results.
- related with display	
Expressiveness	An interface should be able to provide necessary input and output modes warranted by an application. This ability includes different types of inputs and outputs such as haptic data and immersive displays as well as typography including selection of typeface and color.
Aesthetic/cultural acceptance	Information displayed by an interface should be consistent with universal, as well as local cultural and aesthetic norms.

According to the principle of **informativeness**, an advanced interface should be able to prompt several types of knowledge which may (or should) exist in the system. An interface may have access to the following types of knowledge:

- Knowledge that the interface is incrementally receiving from the user and/or other knowledge which exist in the system (including a user profile that the system should maintain).
- Knowledge that the interface should be able to deduce from the knowledge provided by the user.
- Knowledge about the methodology on which the system is based on. For example, several types of modeling methodologies.
- Fundamental scientific and engineering knowledge. This may necessitate agents directly accessing to appropriate sources of knowledge, by on-line payment of a fee, if necessary.
- Knowledge about the application domain (defense, business, etc.)
- Knowledge about the software system and how to use it.

Principle of **perceptiveness** requires an interface to be able to observe the user to perceive the intentions and/or to decide when to initiate an advice.These features are implementable by software agents (Bradshaw 1997).

According to the principle of **explanation ability** an interface should be able to explain the results and justify its decisions. Explanations and justifications can be about the results or the solutions recommended by the system as well as the decisions taken by the system in the generation of its recommendations.

An interface should be able to provide necessary output modes warranted by an application. This is called the principle of **expressiveness**. Examples for expressiveness include direct feeding of actuator devices, voice annotation, on-line video help, multimedia outputs where text, picture (from files or rendered), animations, and video may coexist.

Principle of **aesthetic/cultural acceptance** points out to the need for information display and symbol consistency with universal, as well as local cultural and aesthetic norms. The principle applies to shape, size, location, color, and movement of displayed objects; sound of audio signals and messages; and their relations to other objects. With the increasing number of multinational simulation training exercises, the importance of this principle is likely to increase.

2.3 Reliability Principles

Advantages of satisfying reliability principles include the possibility of filtering and preventing errors. As represented in Table 3, reliability principles can be related with users, usage, or computerization. Reliability principle related with users is the access reliability principle. Those related with usage include predictability, consistency, and safety. The reliability principle related with computerization is the built-in quality assurance principle.

Principle of **access reliability** requires an interface to allow only authorized users to access to the system, if there is such a need. In addition to this feature, when there is a need, an interface should be able to monitor and keep a log of access or attempt to access to the system and report it on-line and/or off-line.

Table 3: Quality Principles for HCIs of M&S Software – Reliability Principles

Reliability principles	
- related with users (access)	
Access reliability	An interface should allow only authorized users to access to the system, if there is such a need.
- related with usage	
Predictability	An interface should do what its users would expect it to do.
Consistency	An interface should be consistent in reacting to the intentions of the user in different contexts.
Safety	The interface should allow a safe usage of the software. This ability includes *error tolerance*, *caution*, and *robustness*. - *Caution* implies that an interface should warn the user and tolerate about irreversible actions.
- related with computerization	
Built-in quality assurance	The interface should allow filtering as well as preventing, as much as possible, specification errors, including *prevention of* input and output *errors*.

The principles of **predictability** and **consistency** require an interface to do what its users would expect it to do and be consistent in reacting to the intentions of the user in different contexts, respectively.

Principle of **safety** asserts that an M&S HCI should allow safe usage of the software. It implies error tolerance, caution, and robustness.

If the errors can not be prevented, a *front-end interface* should, if necessary, tolerate errors (with confirmation). This is called *error tolerance*. In case of erroneous activation of a task, the user should be able to exit without any side effect. An interface should encourage trial-and-error learning without causing frustration. As defined by *caution* an interface should warn the user and tolerate about irreversible actions. One plausible strategy is to include an undo-command (preferably at several levels).

Principle of **built-in quality assurance** requires that an M&S HCI interface should allow filtering as well as preventing, as much as possible, specification errors, including *prevention of* input and output *errors*. Some modeling methodology-based rules for built-in quality assurance were presented in an early article by Ören and Sheng (1988).

Prevention of input errors requires that the *front-end interface* should screen the inputs to prevent errors. This way, one can eliminate several classes of errors. For example in coupling specification of component models, the compatibility of physical units of the input and output variables of the component models to be coupled can be checked. When, unit systems are different such as International System of Unit (SI) and US system of measurement, transducers can be added automatically to convert from one system of unit to another one. Similarly, when both component models use the same measurement system but the input and output variables have different prefixes transducers can be added automatically to convert from one unit to the other one. This type of reliability can be useful for computer-aided modeling environments, for modeling and/or model composition.

On the other hand, *prevention of output errors* requires a *back-end interface* of the system to filter the outputs and intercept unacceptable outputs. A more refined concept is to have control filters for the outputs of component models. If the model description language/environment is advanced enough, one can set flags for low and high values to indicate alarm and danger levels. A software component can then monitor the values and set necessary auxiliary outputs or stop the system.

2.4 Evolvability Principles

Advantages of satisfying evolvability principles include the interchangeability, adaptability, and maintenance of families of interfaces.

As represented in Table 4, evolvability principles can be related with users or software product. Reliability principles related with users include adaptability, customizability, and learning ability principles. Those related with software product include maintainability, and portability principles.

The users that interact with a simulation software can be novices, occasional users, transfer users, and experts. The principle of **adaptability** requires an interface to be adaptable to the needs of different types of users.

Furthermore, as depicted by the principle of **customizability**, one should be able to easily tailor an interface. Customization may be useful for the following types of needs: (1) Preferences (For example tailoring toolbars) and (2) Nationalities. The natural language used in the interface should be easily and correctly understood by users. For this reason, switching from a natural language to another one could be realized in interfaces with multilingual abilities.

Table 4: Quality Principles for HCIs of M&S Software – Evolvability Principles

Evolvability principles	
- related with users	
Adaptability	An interface should be adaptable to the needs of different types of users with different skills and preferences.
Customizability	One should be able to easily tailor an interface.
Learning ability	An interface should be able to remember the usage of the system and should provide relevant knowledge to enhance problem solving abilities of the user.
- related with software product	
Maintainability	Different types of maintenance of the interface should be easy.
Portability	A good HCI should be portable to different platforms.

Learning capability of an interface can be a critical facility to improve the interaction of its users. Personalization is one such capability that takes advantage of learning. Hence, principle of **learning ability** requires an interface to be able to remember the usage of the system and to provide relevant knowledge to enhance problem solving abilities of the user.

Principle of **maintainability** requires that the maintenance of the interface should be easy. Maintenance can be done for corrective, adaptive, preventive, and perfective reasons. *Corrective* maintenance is the modification to correct discovered problems. *Adaptive* maintenance is modification to keep it usable in a changed or changing environment. *Preventive* maintenance is modification to detect and correct latent faults. *Perfective* maintenance is improving performance or maintainability.

Principle of **portability** requires a HCI to be portable to different platforms.

3. A SYSTEMATIC APPROACH FOR EVALUATION AND DESIGN

The quality principles can be used as a systematic basis for the development of quality criteria, metrics, and guidelines that can be used in turn, for the evaluation of existing interfaces, and/or for the design and implementation of new interfaces. One strategy is to define a set of **goals** (based on usability, communicativeness, reliability, and evolvability) that correspond to project level quality objectives. Achieving those quality goals require adherence to certain design **principles** that characterize the rules and guidelines by which interface is developed. Adherence to a process governed by these principles should result in a simulation product that possesses **characteristics** considered to be desirable and beneficial in satisfying the identified quality goal. While the strategy is reminiscent to the goal-question-metric (Basili 1996), the approach is rather based on the definition of the attributes that can serve as a metric or indicator for the principle. By computing a weighted average over the indicators of a given principle, one can compute a proxy measure indicative of the extent to which a principle is adhered to. This measure refers to the measure of satisfiability of a principle. Finally, a similar strategy can be used over the vector of principles associated with the criteria under examination. A weight associated with each principle can be used to indicate the strength of dependency between a criteria and a principle. Along with the measure of satisfiability of each principle the associated weights enable the computation of the satisfiability of a given criteria.

4. CONCLUSION

As important components of any simulation system, interfaces require particular care. Therefore, quality of user/system interfaces is of paramount importance. A set of quality principles is offered for user/system interfaces. The principles can be used as a basis for developing quality criteria, metrics, and guidelines that can be used for proper design of new interfaces, and/ or to evaluate and hence to improve existing ones. As a sequel to this article, studies are under way in the following directions: (1) development of quality principles for intelligent human-computer interfaces (2) a systematic development of quality criteria for M&S software, and (3) development of a software tool for assessing/comparing legacy interfaces and for designing new interfaces.

References

Abdullah, B., Ören, T., (1997). Enhancement of a Simulation Environment with IMAGES (Intelligent Multi-Agent Based Virtual Gauges). In: Proc. of the 1st World Congress on Systems Simulation, Singapore, Sept. 1-4, 1997, pp. 359-363.

Basili, R. V. "Software Modeling and Measurement: The Goal Question Metric Paradigm," *Computer Science Technical Report Series*, CS-TR-2956 (UMIACS-TR-92-96), University of Maryland, College Park, MD, September 1992

Bradshaw, J. (ed.) (1997). Software Agents: AAAI Press, Los Altos, CA.

DigitalHumanSim (2004) Digital Human Simulation and Ergonomics http://www.ccad.uiowa.edu/focus/human/ergo.html

IBM http://www-306.ibm.com/ibm/easy/eou_ext.nsf/publish/6

Mayhew, D.J. (1992). Principles and Guidelines in Software User Interface Design, Prentice Hall, Englewood Cliffs, NJ.

Ören, T.I. (1991). Review of the Book, " An Introduction to Simulation Using GPSS/H " by T.J. Schriber, Wiley, New York, 1991. ACM Transactions on Modelling and Computer Simulation (TOMACS) 1:3, 275-276. Also in ACM Computing Reviews, 33:7 (July), 372-373. Review nb. 9207-0401.

Ören, T.I., Cetin, S. (1999). Quality Criteria for User/System Interfaces. In: Proc. of the Symposium on Modelling and Analysis of C3I Systems, NATO R&T Agency, 12-14 January 1999, Fort d'Issy les Moulineaux, near Paris, France, pp. 18-1 – 18-8.

Ören, T.I., Sheng, G. (1988). Semantic Rules and Facts for an Expert Modelling and Simulation System, In: Proceedings of the 12th IMACS World Congress, Paris, France, July 18-22, 1988, Vol. 2, 596-598.

RAMSIS (2004). Ergonomics and Simulation http://www.humansolutions.com/main_produkte_ergo_e.php

Schriber J. T (1991). An Introduction to Simulation Using GPSS/H. Wiley Publishers.

Shneiderman, B. (1998). Designing the User Interface: Strategies for Effective Human-Computer Interaction, 3rd Edition. Addison Wesley Longman Inc. 638p.

Walters, B., 2003, Evaluating User Interfaces with Simulation Modeling, HCI International, June 22-27, 2003 http://www.hcii2003.gr/program/sigs/s03.asp