Quality Principles for the Ergonomics of Human-Computer Interfaces of M&S Software

Dr. Tuncer I. Ören
M&SNet: Ottawa Center of the MISS
University of Ottawa
Ottawa, ON, Canada
http://www.site.uottawa.ca/~oren

Dr. Levent Yilmaz
M&SNet: Auburn M&S Laboratory
Computer Science & Software Engineering
Auburn University, Auburn, AL 36849
http://www.eng.auburn.edu/~yilmaz
Aim

- Our **aim** is to present a framework within which systematic studies on the assessment of the ergonomics of the Human-Computer Interfaces for M&S software and environments can be performed.
Our Perspective

- A set of **goals** that correspond to project level quality objectives need to be well understood.

- Achieving those quality goals require adherence to and use of **principled** design strategies.

- Conformance to a methodology governed by these principles should result in a simulation model that possesses **characteristics** considered to be desirable and beneficial in satisfying the identified quality goal.
Plan

1. Background – *Quality*

2. Quality objectives and principles
   - Usability
   - Communicativeness
   - Reliability
   - Evolvability

3. Toward a Systematic Approach Toward Evaluation and Design

4. Conclusions
Quality – The Basics:

IEEE

• “Totality of features of a software product that bears on its ability to satisfy given needs.” [Source: IEEE-STD-729]

• “Composite characteristics of software that determine the degree to which the software in use will meet the expectations of the customer.” [Source: IEEE-STD-729]
Design quality is also about fitness to purpose

  does it do what is needed?
  does it do it in the way that its users need it to?
  does it do it reliably enough? fast enough? safely enough? securely enough?
  will it be affordable? will it be ready when its users need it?
  can it be changed as the needs change?

This means quality is not a measure in isolation

  it is rather a measure of the relationship between simulation software and its application domain

  might not be able to measure this until you place the software into its environment…and the quality will be different in different environments!
Quality M&S Software

Reliable/accurate
- reliability
- fault tolerance
- testability

Secure/private
- privacy
- confidentiality
- security
- availability
- performance
- intrusion tolerance
- fault tolerance

Timeliness

Functional attributes
Non-functional attributes ("ilities")
Common Strategy: **Measuring Quality**

We have to turn our vague ideas about quality into measurables.

**The Quality Concepts**
- (abstract notions of quality properties)

**Measurable Quantities**
- (define some metrics)

**Counts taken from Design Representations**
- (realization of the metrics)

- **reliability**
  - mean time to failure?
  - run it and count crashes per hour???

- **complexity**
  - information flow between modules?
  - count procedure calls???

- **usability**
  - time taken to learn how to use?
  - minutes taken for some user task???
Achieving those quality goals requires engineering quality via the use of *principled* model design strategies.
Plan

1. Background – Quality

2. Quality objectives and principles
   - Usability
   - Communicativeness
   - Reliability
   - Evolvability

3. Toward a Systematic Approach Toward Evaluation and Design

4. Conclusions
Advantages of satisfying convenience or 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.
## Usability Principles

<table>
<thead>
<tr>
<th>Name of the principle</th>
<th>Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convenience of the language</td>
<td>The natural language used in an interface should not hinder the proper use of the system.</td>
</tr>
<tr>
<td>Convenience of the terminology</td>
<td>The terminology used in the interface should be based on the application domain’s terminology.</td>
</tr>
<tr>
<td>Convenience of the metaphor</td>
<td>The metaphor used in the interface should be most appropriate (i.e., natural) for the application domain.</td>
</tr>
<tr>
<td>Convenience of the inputs</td>
<td>A <em>front-end interface</em> should be capable to accept the types of inputs most appropriate (i.e., natural) for the application.</td>
</tr>
</tbody>
</table>
## Usability Principles (cont’d)

<table>
<thead>
<tr>
<th>Principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>An interface should offer complete set of abilities to specify problems and to process, analyze, and present results.</td>
</tr>
<tr>
<td>Separation of concerns</td>
<td>An interface should allow focusing on different aspects of M&amp;S.</td>
</tr>
<tr>
<td>Simplicity</td>
<td>An interface should not have distractive information.</td>
</tr>
<tr>
<td>Consistency and uniformity</td>
<td>There should be no ambiguity to initiate an action in different parts of the interface.</td>
</tr>
<tr>
<td>Minimum memory load</td>
<td>Users should not be obliged to remember information from one part of the interface to be used in another part.</td>
</tr>
<tr>
<td>Navigability</td>
<td>Activities should be initiated as directly as possible.</td>
</tr>
<tr>
<td>Least training</td>
<td>An interface should require least amount of training.</td>
</tr>
</tbody>
</table>
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 of support for different types of users.
# Communicativeness Principles

<table>
<thead>
<tr>
<th>Communicativeness</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informativeness/guidance</td>
<td>An advanced interface should be able to prompt several types of knowledge which may (or should) exist in the system.</td>
</tr>
<tr>
<td>Perceptiveness</td>
<td>An interface should be able to observe the user to perceive the intentions and/or to decide when to initiate an advice.</td>
</tr>
<tr>
<td>Explanation ability</td>
<td>An interface should be able to explain the results and justify its decisions.</td>
</tr>
<tr>
<td>Expressiveness</td>
<td>An interface should be able to provide necessary output modes warranted by an application.</td>
</tr>
<tr>
<td>Esthetic/cultural acceptance</td>
<td>Information displayed by an interface should be consistent with universal, as well as local cultural and aesthetic norms.</td>
</tr>
<tr>
<td>Relationship with user</td>
<td>Patronizing, informal, and insulting tone should not be used.</td>
</tr>
</tbody>
</table>
Advantages of satisfying evolvability criteria include the interchangeability, adaptability, and maintenance of families of interfaces.
# Evolvability Principles

<table>
<thead>
<tr>
<th>Evolvability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptability</td>
<td>An interface should be adaptable to the needs of different types of users.</td>
</tr>
<tr>
<td>Customizability</td>
<td>One should be able to easily tailor an interface.</td>
</tr>
<tr>
<td>Learning ability</td>
<td>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.</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Maintenance of the interface should be easy.</td>
</tr>
<tr>
<td>Portability</td>
<td>A good HCI should be portable to different platforms.</td>
</tr>
</tbody>
</table>
Advantages of satisfying reliability criteria include the possibility of preventing and avoidance of errors.
<table>
<thead>
<tr>
<th>Reliability</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Access security</td>
<td>An interface should protect a system from attacks.</td>
</tr>
<tr>
<td>Access reliability</td>
<td>An interface should allow only authorized users to access to the system, if there is such a need.</td>
</tr>
<tr>
<td>Prevention of input errors</td>
<td>The <em>front-end interface</em> should screen the inputs to prevent errors.</td>
</tr>
<tr>
<td>Prevention of output errors</td>
<td>A <em>back-end interface</em> of the system should filter the outputs and intercept unacceptable outputs.</td>
</tr>
</tbody>
</table>
Reliability Principles (cont’d)

<table>
<thead>
<tr>
<th>Error tolerance</th>
<th>A front-end interface should tolerate errors (with confirmation).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caution</td>
<td>An interface should warn the user and tolerate about irreversible actions.</td>
</tr>
<tr>
<td>Predictability</td>
<td>An interface should do what its users would expect it to do.</td>
</tr>
<tr>
<td>Consistency</td>
<td>An interface should be consistent in reacting to the intentions of the user in different contexts.</td>
</tr>
</tbody>
</table>
Plan

1. Background – Quality
2. Quality objectives and principles
   • Usability
   • Communicativeness
   • Reliability
   • Evolvability
3. Toward a Systematic Approach Toward Evaluation and Design
4. Conclusions
A Strategy for Engineering Quality

The quality principles can be used as a systematic basis for the development of quality criteria, evaluation of existing interfaces, and/or for the design and implementation of new interfaces.
Criteria – Principles - Metrics

M&S Interface
Ergonomics

Criteria

Usability
Communicativeness
Evolvability
Reliability

Usability principles
Communicativeness principles
Evolvability principles
Reliability principles

A1 A2 A3

An
Plan

1. Background – *Quality*
2. Quality objectives and principles
   - Usability
   - Communicativeness
   - Reliability
   - Evolvability
3. Toward a Systematic Approach Toward Evaluation and Design
4. Conclusions
Conclusions

• As important components of any simulation system, interfaces require particular care. Therefore, quality of user/system interfaces is of paramount importance.
• As a sequel to this preliminary work, 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