A PROPOSED CLASSIFICATION OF SIMULATORS

LEONARD M. POTT*, ARNE O. BUdde**, AND W. Bosseau Murray***

Summary
This paper provides a coherent and comprehensive classification of simulators, using a five letter coding system, and is based on the characteristics of the user interface and the logic controller.

Introduction
Simulators are a well-accepted component of training and assessment1,2… A variety of different classifications of simulators have been proposed2,5 but no accepted classification scheme has yet been devised6.

The classification schemes in the literature tend to be broad, and with few categories. A number of recent reviews have examined the various simulators currently available3,5. An ideal classification should provide adequate information about the technology used in the design of the simulator, an indication of how the simulator may be utilized, and the classification system should be easy to remember. Understanding the structure of simulators may aid the choice of an appropriate device for teaching and for assessing a particular skill. Additionally, an understanding of the various design choices available may aid the design and construction of simulators.

Gaba has defined simulators as a “device that presents a simulated patient (or part of a patient) and interacts appropriately with the actions taken by the simulation participant”7. Simulators are devices which represent reality and allow for user interaction. Representations of reality are called models, and the objective of any model builder is to extract those, and only those, aspects of reality required by the model user, and incorporate the identified features into the model. Healthcare simulators therefore do not include every feature of the patient or the patient’s environment.

In this paper we propose a coherent and comprehensive classification of simulators, based on a five letter coding system.
Proposed classification

Simulators can be classified based on three criteria.

Firstly, simulators can be classified in terms of the input and output user interface. Simplistically, input is what you perform, and output is the response to your actions.

Secondly, our classification is based on the method by which a specific output response is linked to a particular input. We call the method which links the input and output the interaction controller. The interaction controller may be seen as a ‘black-box’. The input is how the user interacts with the black-box, and the output is how the black-box responds. How the ‘black-box’ works is nevertheless important for the instructor, and is therefore part of our proposed classification.

The third component of the classification is whether the simulator can be used for more than one learner simultaneously, that is, the degree of interactivity.

Each of these criteria will be discussed in more detail below, and these three criteria will be subdivided further. Each subdivision will be identified by a letter, which will be introduced in the text.

1. The user interface. The user interface consists of the input and output of the simulator, and is assessed from the perspective of the user. Input and output can be:

   R Real-life

   A Real-life input is a user input using equipment and physical movements which are essentially the same as those used in clinical practice. Real-life output is when the user’s senses (visual, auditory, tactile, etc) are stimulated in essentially the same way as in clinical practice. A Real-life interface frequently involves the movement of physical objects, for example squeezing a bag to ventilate a simulated patient as the input, and seeing the chest rise from a mannequin as the output. How the simulator processes the input is part of the interaction controller (which will be discussed in more detail below), and not part of the user interface. Real-life input also includes any monitor or video display found in normal practice. For example, entering infusion rates on an infusion pump is a Real-life input and a rise in blood pressure on the anesthesia monitor is an example of the corresponding Real-life output.

   Real-life input or output will be represented by the letter ‘R’.

   S Simulated

   Simulated input and output involves an interaction with the simulator which is different from a Real-life interaction discussed above. Simulated input typically involves the use of a mouse, or keyboard, or touchscreen, and Simulated output typically involves some form of video display and may be accompanied by sound. Using the above example again, Simulated input would be to choose “intubate patient” from a menu in a computer program as the Simulated input and then seeing a video-clip of an intubated patient and hearing breath sounds as the Simulated output. Infusion rates would be entered by choosing a number for milliliters per minute from a drop down menu per mouse click and computer-based output would be a cartoon of a dripping intra-venous line. Simulated input also includes situations where the user tells another person, such as the simulator operator, what the user intends to do, and the second person transmits that information to the interaction controller (see Table for examples).

   Simulated input or output will be represented by the letter ‘S’.

   D Dual

   Some simulators, particularly virtual reality surgical simulators, have both Simulated (a video screen) and Real-life output (haptic feedback on the surgical instrument). When neither form of output is dominant, then the output can be classified as Dual. Whether a form of output is dominant can be decided by excluding one form of output and checking whether the simulator still functions adequately. For example, the Anesthesia Simulator (Anesoft) has an auditory output which can be switched off without affecting the simulator significantly, and therefore does not have Dual output. We are not aware of any current Dual input simulators.

   Dual input or output will be represented by the letter ‘D’.

   2. The interaction controller. The controller will determine the appropriate output for any particular input. Although the interaction controller may be
viewed as a “black-box” by the user, it is the controller which enables the interaction. The linking of a particular input to a specific output can be structured in a variety of different ways.

One possibility is that the output is merely a mechanical consequence of the input. An example of a simulator with a mechanical controller is a head and airway used for teaching, practicing and assessing intubation skills. Other examples would include hydraulic simulators. Mechanical controllers are represented by the letter ‘N’ (for None) because no specific interaction controller exists. We chose to avoid the logical choice ‘M’ because that letter will be used in another context, discussed below.

A second possibility is that a human operator may activate the response depending on the input observed by the operator. The user may not even be aware that the operator is controlling the simulator, for example, when the operator initiates a response by remote control. The SimMan® is an example of an Operator controlled simulation. What is significant is that the operator makes some (subjective) decision about what the output will be. Operator driven interaction controllers are represented by the letter ‘O’.

The third possibility is that some form of logic device, such as the CPU (Central Processing Unit) of a computer, determines the output. For our proposed classification we consider two alternatives, either mathematically modeled or algorithmically driven (rule-based).

Mathematically modeled.

In the case of mathematical modeling, the output is calculated using a set of mathematical equations\(^5\). These equations are expressed in terms of variables. The specific values of the variables are determined by the user input, and then the output is calculated by the interaction controller. Mathematical models require equations which attempt to describe physiological events. Because the human body is extremely complex, sophisticated modeling techniques are used\(^5\). If the model only represents one aspect of the body’s physiology then the modeling is simpler, for example, the GasMan© program which models anesthetic gas uptake and elimination\(^8\).

Mathematically modeled interaction controllers are represented by the letter ‘M’.

An advantage of mathematical modeling is that graphical representations of the physiological process may easily be produced, which are useful for teaching. Models such as GasMan© closely predict empirically determined data\(^9\).

A relative disadvantage of mathematical modeling of physiological processes is the complexity of the software and the enormously high development costs. Many of the mathematically modeled physiological programs use very similar sets of equations\(^5\). However the mathematical modeling of pharmacokinetic simulators is relatively easy.

Algorithmically driven.

Algorithmically derived output is not calculated but is pre-determined by the simulator developer. How the simulator appears to the user at any particular time is known as the state of the simulator, and these states can change. The simulator program consists of a series of branching conditional statements. Basically what happens is that there is a transition from one state to another state depending on the presence or absence of various conditions. If the prespecified conditions are met, then the state changes. The user provides the input, and if the input is the condition necessary for a state change, then there is a transition to the new state. The resultant pathway essentially follows a clinical decision tree\(^10\) where the choice at decision nodes require input from the user.

A simple example of an algorithmic simulator is a set of text documents (pages). The first page describes a specific medical situation and poses a clinical management question. Various alternative solutions are offered as hyperlinks to the next text document. The hyperlink selected reflects the user’s answer choice and will determine which text document is presented next. After making a series of choices the user will reach the final document (page), and the user’s decision tree can be reconstructed. Schwid calls the “pages” states, and the “hyperlinks” transitions\(^11\). Because there are only a limited number of states, he calls such a program a finite state model\(^11\).

Tightly scripted simulation scenario’s using Standardized Patients or involving mannequins controlled by an operator, are also considered...
algorithmically driven.

Algorithmically driven interaction controllers are represented by the letter ‘A’.

**Combination.**

The basic physiological function of a simulator such as the METI® is primarily under the control of a mathematical model. However, the simulation technician can initiate an event or complication at any time. This is an example of a primary controller being overridden by a secondary controller. Primary and secondary controllers refer to the software design, and for convenience we refer to the primary controller as Controller1 and the secondary controller as Controller2.

In the Anesthesia Consultant Simulator® transitions are primarily determined by user input but may be overridden by the mathematical model. For example, if the user of the simulation delays the ventilation of a paralyzed patient for too long, then the mathematical model will determine that the patient’s oxygen saturation will fall and ultimately the patient will die. Therefore the Anesthesia Consultant Simulator® may be classified as primarily an algorithm driven controller (Controller1) with a secondary mathematical model capable of overriding the algorithm (Controller2).

Logically, there can be no secondary controller without a primary controller, so only one letter can be used if no secondary controller is used; that is, instead of writing an N in the second position this letter can be omitted. For example, instead of writing AN for a pure algorithmically driven simulator only A is written.

Alternative types of interaction controllers are:

1. N - Mechanical
2. O - Operator (manual control by technician)
3. Central processing unit
   a) M - Mathematically modeled
   b) A - Algorithm driven
4. Combination
   a) Primary controller (abbreviated to CONTROLLER 1)
   O - Operator
   M - Mathematical model
   A - Algorithm
   b) Secondary controller (abbreviated to CONTROLLER 2)
   O - Operator
   M - Mathematical model
   A - Algorithm

3. The environment determines if a simulator can be used interactively by a group of two or more persons.
   G - Group interaction possible
   I - Individual person only
   B - Both options are possible

Using the above described definitions, simulators can be characterized by a five letter coding system. We propose the sequence:

**INPUT: OUTPUT-CONTROLLER 1: CONTROLLER 2-INTERACTION**

Our simulator classification system is delineated in Table 1 below:

**Input.** The first letter defines the user input interface and indicates what kind of input is used; for example, R represents a real-life input, or an S for a form of simulated input.

**Output.** The second letter tells which kind of output can be expected, again either an R or a C.

**Primary Controller.** The third letter describes the primary

**Secondary Controller.** The fourth letter describes the secondary controller.

**Interaction.** The fifth letter describes the interaction between the primary and secondary controllers.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Control</th>
<th>Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Controller1</td>
<td>Controller2</td>
</tr>
<tr>
<td>Output</td>
<td>N - None</td>
<td>N - None</td>
</tr>
<tr>
<td>R - Real-life</td>
<td>O - Operator</td>
<td>O - Operator</td>
</tr>
<tr>
<td>S - Simulated</td>
<td>M - Mathematical</td>
<td>M - Mathematical</td>
</tr>
<tr>
<td>D - Dual</td>
<td>A - Algorithm</td>
<td>A - Algorithm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>G - Group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I - Individual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B - Both</td>
</tr>
</tbody>
</table>
controller, which could be N, O, M or A (see text).

**Secondary Controller.** The fourth letter describes the secondary controller which, if present, may override the primary controller.

**Interaction.** The fifth letter indicates that the simulator can be used by a Group or by a Individual person only, or Both alternatives (G or I or B).

**Discussion**

Classification and coding systems have been used in many fields to aid the understanding and use of products. An example is the classification of electrical cables by the National Electrical Code. The use of coding systems is well established in gynecology and obstetrics using the GPTPAL code as well as in cardiology, where a specific code for cardiac pacemakers was first introduced in 1974 and then further developed and revised. Our proposal is that simulators be classified using a similar method.

A simulator is the device used, and simulation is the process. Some of the more complex simulators, such as the METI can be used in more than one simulator mode. So, for example, the METI mannequin can be switched off and still be used to practice intubation. Therefore, in terms of function, one particular simulator can be classified in different ways. It is our suggestion that simulators are primarily classified in terms of their structure. If however, a simulator is used differently (such as the METI for intubation practice), then a statement such as “….the METI, used as a RRNS simulator…” is applicable. Examples can be seen in Table 2.

**Table 2**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>Controller 1</th>
<th>Controller 2</th>
<th>Interaction</th>
<th>Task Example</th>
<th>Simulator Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>N</td>
<td></td>
<td>I</td>
<td>Checking peripheral pulses</td>
<td>METI, SimMan, SP</td>
<td>METI expensive, SP possibly better choice</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>N</td>
<td></td>
<td>I</td>
<td>Needle decompression for a tension pneumothorax</td>
<td>SP</td>
<td>inappropriate to train this task</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>O</td>
<td>N</td>
<td>I</td>
<td>Palpation of tender abdomen</td>
<td>METI, SimMan</td>
<td>hands on training, no harm for the simulator</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>M</td>
<td>A</td>
<td>I</td>
<td></td>
<td>Anesoft</td>
<td>how to diagnose, cognitive skills</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>O</td>
<td>N</td>
<td>I</td>
<td>Intubation</td>
<td>Laerdal Head</td>
<td>sufficient to train the task</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>O</td>
<td></td>
<td>I</td>
<td></td>
<td>METI</td>
<td>more realistic environment due to full body mannequin</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>A</td>
<td>M</td>
<td>I</td>
<td>Advanced Cardiac Life Support (ACLS)</td>
<td>Anesoft</td>
<td>Teaching of theoretical principles</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>M</td>
<td>A</td>
<td>B</td>
<td></td>
<td>Anesoft, Body</td>
<td>Teaching of theoretical principles</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>M</td>
<td>O</td>
<td>B</td>
<td></td>
<td>METI</td>
<td>Hands on training</td>
</tr>
<tr>
<td>R</td>
<td>S</td>
<td>M</td>
<td></td>
<td>I</td>
<td>Laparoscopic suturing</td>
<td>LapSim VR simulator</td>
<td>Virtual reality</td>
</tr>
<tr>
<td>R</td>
<td>D</td>
<td>N</td>
<td>M</td>
<td>I</td>
<td></td>
<td>ProMIS AR simulator</td>
<td>Augmented reality, more realistic due to haptic feedback</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>Various</td>
<td>Various</td>
<td>G</td>
<td>CRM</td>
<td>Meti, SimMan, Resuci Anne</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>M</td>
<td>A</td>
<td>G</td>
<td>Emergency room triage</td>
<td>Second Life</td>
<td>Still experimental</td>
</tr>
</tbody>
</table>

M.E.J. ANESTH 20 (2), 2009
The proposed classification code serves to summarize some of the features of a particular simulator, but cannot include all information about a simulator. Additional information that may be necessary will depend on the circumstances and may include, for example, the product name and supplier, or the physical dimensions. Some classifications make a distinction between high-fidelity and low-fidelity simulators\(^\text{16}\). Our classification makes no such distinction because the degree of realism depends to a large extent on the surroundings in which the simulator is used\(^\text{3,4,17}\).

Rall and Gaba have stated that “(s)ome individuals consider the screen-only microsimulator (also known as a screen-based simulator) to be a training device and not a simulator”\(^\text{6}\). We have made no distinction between simulators used for training and those used for assessment, and agree with the inclusive approach of Cumin and Merry\(^3\). Any device which is a representation of reality and allows for interaction is considered a simulator.

We have chosen letters for the code which are unique in each category, so that if a category of the classification is not relevant in a particular situation, it may simply be omitted. For example, -MA- (mathematical model overridden by an algorithm) only refers to the interaction controller, and -RR- (real-life input, real-life output) only refers to the user interface. We chose the term ‘Real-life’ rather than the alternative ‘Physical’ because the term is intended to convey the sense that the object looks and feels like the equivalent real-life object.

A classification of simulators such as the one presented in this paper has some major advantages.

1. By specifying the classification of the controller the software architecture is clear; something which is not always immediately apparent using other classifications. Knowing that the primary controller is algorithmic indicates that the program can be relatively easily modified.

2. If a simulator does not have an ‘R’ (Real-life) for input then it is extremely unlikely that such a simulator could be used for training, or evaluating, psychomotor skills\(^3\). In order to become proficient at a motor task practice is necessary, and only a real-life input simulator will provide the conditions for practice necessary to develop the coordination of motor skills. While other classes of simulator may be usefully employed to provide the theoretical and cognitive background, there is probably no substitute for physical practice.

3. If objective assessment is desired then any simulator with an ‘O’ in either of the controller positions, that is, any simulator which is controlled or overridden by the instructor, is unlikely to be suitable. As soon as a person, who is aware of the situation pertaining to the evaluation, has the discretion to alter the simulation sequence then the evaluation becomes subjective. On the other hand, if the operator works according to a very rigorous and precisely defined script then the operator is essentially an algorithmically driven controller, and with little effort can be replaced by a software program. To the extent that the script remains undefined, the assessment is subjective.

4. Certain theoretically possible classes of simulator have not yet been constructed, and this classification identifies simulators of the form S-R, that is computerized input and a real-life output. Depending on the educational or assessment objectives, it may prove useful to design and build a simulator representing a specific category.

5. SS-A simulators are eminently suitable for Web-based applications, whereas the modification of simulators for Web applications with R in either position will be problematic.

We present this classification of simulators in the hope that, by adopting a simple coding of simulators, communication in the simulation community will be enhanced. Although our proposal may require modification and adaptation, we believe that the classification is coherent, comprehensive and easy to use.
References


