DOI: https://doi.org/10.46354/i3m.2019.iwish.019

3D SOFTWARE SIMULATOR FOR PRIMARY CARE TRAINING

Alessandro De Gloria[(a)], Marco Chirico[(b)], Claudio Launo[(c)], Riccardo Berta[(d)], Francesco Bellotti[(e)], Giuseppe Gioco[(f)], Ivan Carmosino[(g)]

[(a)(d)(e)]D.I.T.E.N - University of Genoa, [(b)]SimAv - University of Genoa
[(c):] DIMI – University of Genoa [(f)(g)] Wondertech srl

[(a)]adg@elios.unige.it[(b)]marco.chirico@unige.it[(c)]claudio.launo@unige.it[(d)]riccardo.berta@unige.it [(e)]franz@elios.unige.it,
[(f)]giuseppe.gioco@wondertechweb.com[(g)]ivan.carmosino@wondertechweb.com

ABSTRACT
Medical software simulators are used to teach specific procedures that allow the user to follow only a strict sequence of steps without the possibility of alternative, avoiding considering the consequence of an error and then potentially admitting its tolerance. Usually these applications are a state machine implementation where learners must make a specific action to obtain a specific result.
In our work we propose a brand-new approach with a “open world” serious game medical simulator, based on Agent Based Model Paradigm.
Starting by these concepts, a user can learn and test his skills in a dynamic environment that changes in real time based on his actions.
We provide a configurable starting set of conditions (patient heath state, available medical instruments and drugs) to create, potentially, infinite scenarios; alongside these boundary values the game permits to configure real time events that influence patient in an unpredictable way by the user side.

Keywords
Open word, serious game, Agent Based Model Paradigm, dynamic environment

1. INTRODUCTION

Training software for primary care behavior is one of the best tool to teach the best way to resolve real-life simulation emergencies (e.g. safe life after a hearth stroke).
Applications like MicroSim Inhospital[1] or DrSim[2] offer some clinical cases that must be resolved in a specific way and in a specific time lapse.
Obviously, this approach allows a deep clinical case analysis, but the simulation solution is only one, and in every interaction the user has to make always the same decisions (and obtain the same results) to end the simulation in the right way.

Starting by this observation we propose a new approach to this training scenario: a dynamic environment with his own rules that can change by user interactions or by real time events.
The user can interact with this environment in multiple ways and obtain multiple results.

The software implementation is based on the Agent Based Model Simulation (ABMS).
An Agent is shortly defined as an Entity with a set of actions that can change its own state or other agents state; an Agent can react to external stimuli only, be *cognitive* making its own decision, or a mix of these characteristics [3].
ABMS is used in various disciplines like biology, finance, engineering to study complex scenario too expansive or difficult to replicate [4].

Based on ABMS we defined three Agents (called Game Agents):

- Patient Agent
- Tool Agent
- Director Agent

The simulation was implemented as a 3D serious game where the user plays the doctor role.
We decide to develop this kind of simulation because it has been proved that Gamification [5] produces a better learning curve with respect the classic learning methods [6].

The ABMS paradigm offers the opportunity to approach the game in a “open world” way: to proceed in the simulation it’s no need to do specific actions, but user can make his own decisions and see what happen in the game world: success or failure is not written in a decision tree.

Finally, with the simulator we implemented a powerful and simple editor to create the starting configuration (e.g. patient heath state, events, available drugs etc.); this set is called Plot.

2. PLOT AND GAME AGENTS

To create our training simulator, we focused on two key goals:

- Abstract the simulation concept to create a, potentially, infinite simulations cases
- Create a dynamic simulation environment where simulation can evolve runtime and not only after a user action.

For the first goal we create the Plot concept: a set of configurable variables that represent the starting simulation point.
For the second one we create the Game Agent concept: an Entity based on ABMS paradigm that can change runtime both his own and the environment state.

Our training simulation starts from a certain point and evolves by its own.
Every simulation is different from an another and the same simulation can evolve differently in any interaction.

2.1. Games Agents Interaction

We define three Game Agents; every agent can change his state and, in some case, the state of one or more other agents.
User can interact with simulation using the Tool Agent. Figure 1 explains how this works.

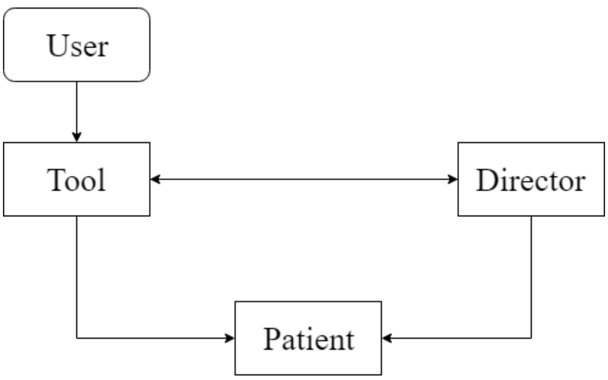

Figure 1: User and Game Agents interactions

2.2. Patient Agent

The Patient is a *hybrid* Agent: his state can be modified by other Agents (external stimuli) and his actions (Rules).

In our simulation environment, Patient is a human being abstraction, characterized by life parameters and physiological rules divided in three macro areas:

- Primary Parameters: a set of vital parameters like blood pressure and breath status.
- Secondary Parameters: another set of vital parameters, (e.g. ECG sinusoidal, CVP) that depend on Primary ones.
- Rules: a set of actions based on physiological functions that modify Primary and Secondary Parameters values.

Primary Parameters can be modified by other Agents and Patient Rules.
From the other hand Secondary Parameters can be modified only by Rules and only when Primary Parameters change occurs.
Patient checks Parameters and apply Rules in real-time. Figure 2 explains this interaction.

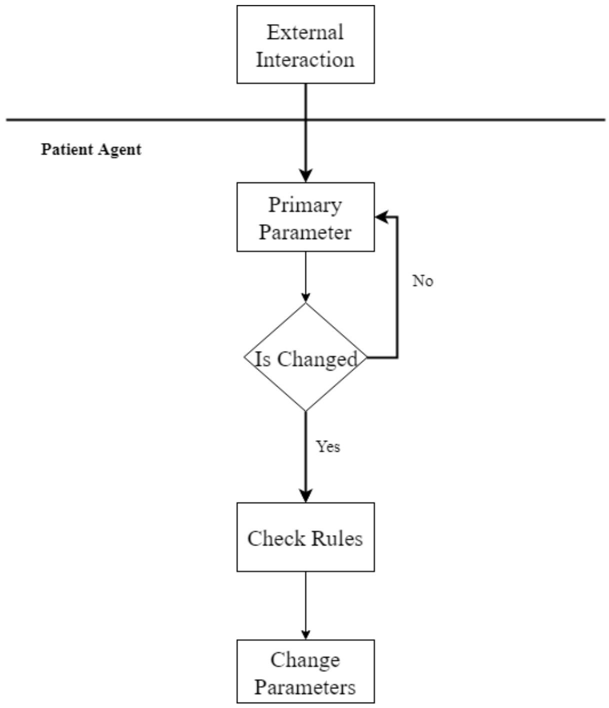

Figure 2: Patient Agent Logics

Here an example how Patient works.
We define a set of Primary Parameters: Systolic Pressure, Diastolic Pressure and Heart Frequency.
These values are related to a set of Secondary Parameters that includes ECG, Radial Pulse and Carotid Pulse.
The relation between these two sets is managed by following Rules:

- ECG rule sets ECG based on Heartbeat and both Pressures.
- Pulse rule set radial and carotid pulse based on systolic pressure threshold.

When, Heart Frequency change, ECG Rule checks the new value and consequently modifies (if needed) the ECG value.

In our implementation Patient Parameters and Rules are not fixed and can be added or modified in a simple way. We can focus, for example, on neurological or cardiopulmonary simulation or maybe a mix of them.

Obviously, an approach like this need a good knowledge of human physiology but, from the other hand, can offer a detailed and realistic simulation.

2.3. Tool Agent

Tool is a *cognitive* Agent that, with its actions, changes his own state and the Patient Agent one.

In our environment a Tool is a 3D object modeled on a real-life medical instrument, procedure or drug and represents the only way a user can interact with Patient**.**

2.3.1. Tool interaction and synchronization with simulation environment

Although is cognitive (it does no need stimuli from external source to work), Tool starts simulation in a Sleep State. It turns awake only when user actively uses it (as shown in Figure 1, Tools are the only way for user to interact with Patient) and only after it is synchronized with the simulation environment.

In real-life, a doctor usually can make only one action at the same time, for example it is impossible to administrate a drug and at the same time make a CPR (Cardiopulmonary Resuscitation)

To simulate this real-life situation, before starting his actions, a Tool must be synchronized with simulation environment and this process is managed by Director that detains simulation status and simulation time.

Finally, when user stops using a Tool, it returns in Sleep State.

2.3.2. Active and Passive Tools

We divide Tools in two macro types: Passive Tools that offer a real time Patient conditions snapshot, visualizing Parameters in a particular way (for example ECG monitor shows a sinusoidal curve based on Heartbeat) or simply showing them as they are (Breath per minutes or radial/carotid pulse are an examples). Figure 3 shows how they work.

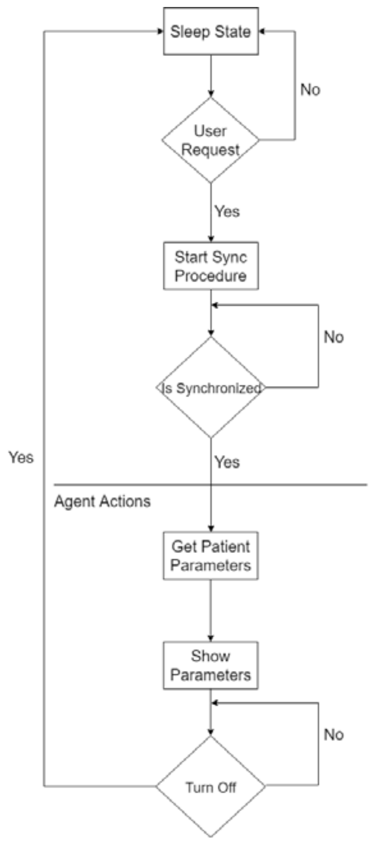

Figure 3: Passive Tool

The other type is Active Tools: the set of instruments, drugs and procedure used during simulation (CRP procedure, drugs and defibrillator are an example). These Tools modify Patient Primary Parameters in many ways: permanently, only when not in Sleep State or for a time lapse.

In real-life, using some of these Tools need a decision making, for example how much drug to administrate or where to apply defibrillator plates: a wrong decision could lead to a failure.

To simulate this aspect, Active Tools actions are parameterized, so, before using them, the user must set these variables. Figure 4 shows how Active Tools work.

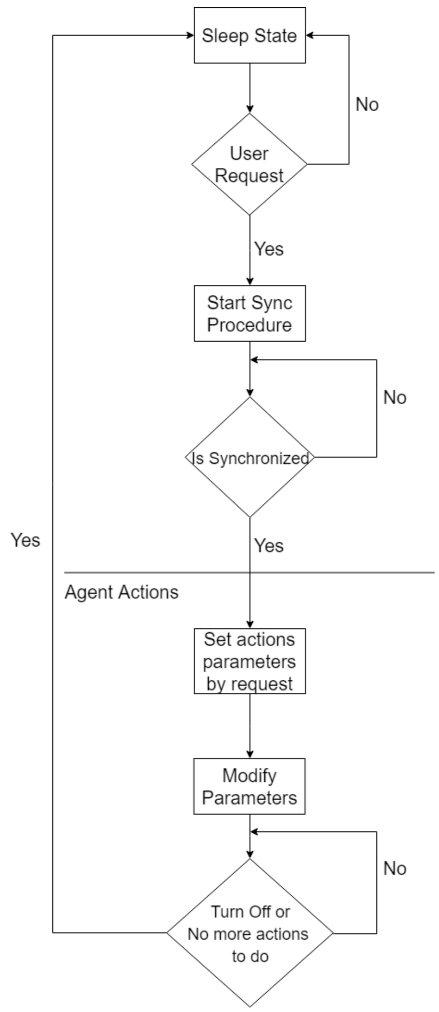

Figure 4: Active Tool

2.3.3. RNG Factor

In real-life a good decision couldn't always lead to success. Something unexpected can occur; let's make an example: during a heart stroke with a ventricular fibrillation doctor can use adrenalin to stabilize the patient. Let's suppose this doctor makes all the right decisions (right drug quantity, right administration way, etc.) but something goes wrong and patient conditions gets worse. At this time doctor must take some new decisions in a really stress situation.

To simulate these unexpected results, we introduce a configurable RNG factor to every Tool that affect actions success percentage.

2.4. Plot

Plot is related to Patient and Tool Game Agents and must be explained to understand Director Game Agent.

We defined Plot as a set of configurable variables that represent the starting simulation point.
These variables include:

- Patient Primary Parameters starting values
- Which Tools can be used
- Realtime Events list

Every training simulation has its own Plot and every Plot differs by little or big aspects.

Another concept we introduced is Realtime Event: A Realtime Event (RTE) is a configurable action that modifies Patient state in a default time/values condition.

In every RTE we can find:

- What Parameter must be modified
- When this change must occur
- In which conditions this change must be made

For example, if we want set Systolic Pressure to 150 after 10 seconds; we create a RTE like this:

- What: Systolic pressure
- When: after 10 secs
- Conditions: no conditions

In another case of we want set the heart frequency to 200 but only if systolic pressure is more than 150:

- What: heart frequency
- When: always
- Conditions: Systolic pressure more then 150

We create also an RTE that occurs after 10 second but only if a specific condition is verified.

2.5. Director Agent

Director is a *cognitive* Agent that manage simulation evolution and can interact with Tools and Patient Agents.

Director duties are:

- Read Plot
- Set starting Patient Parameters.
- Enable Tools availability
- Manage simulation time
- Synchronize Tools
- Register user actions
- Manage the Real Time Events

Director has a set of asynchronous actions, triggered by Tools requests, and a synchronous set made runtime. Figure 5 shows the synchronous behavior.

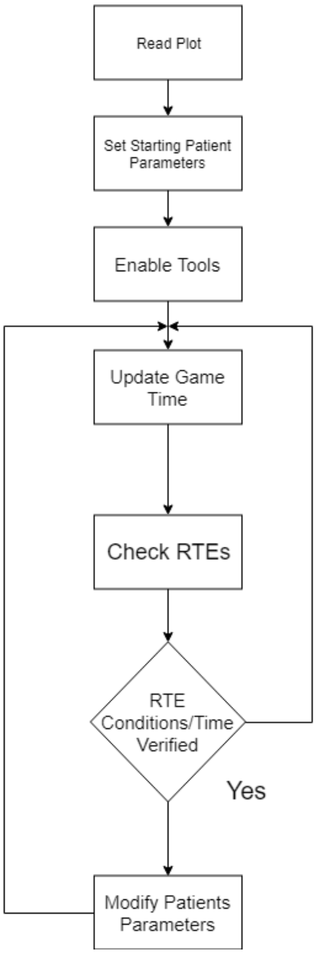

Figure 5: Director synchronous behavior

2.5.1. Asynchronous Actions and Simulation History

As we told, every Tool must be synchronized with simulation environment by making a request to Director that registers every Tool used during simulation, their values (for Active Tools) and the using time.

All these information can be used after the simulation is concluded to evaluate user decisions.

3. GAME WORLD IMPLEMENTATION

Usually simulation software offers detailed clinical cases, but they are limited in number and iterations: a user learns fast how simulation works and therefore can make always the right decisions.

An important aspect, especially in primary care training, is to put some pressure to the learner: in real life a doctor must take difficult decisions in short time and their consequences make difference between life and death: posology comes first then diagnosis.

To satisfy these requirements we create Plot that offer a flexible customization template to create a great number of clinical cases, and Game Agents that, with their own logic and RNG factor, creates a dynamic simulation with unpredictable evolutions.

Next step is to create a game environment where put our Game Agents.

3.1. A Serious Game as Immersive Simulation

In a medical training simulation learner improve his skills and understand how resolve emergency, but he knows that a wrong decision doesn't cost a life.

Many simulators offer all these features but lacks in realism. Emotional aspect, pressure and hurry are difficult to replicate sitting in front of a computer.

To overcome this problem, we propose a video game approach (obviously as serious game).

We told about Gamification and its learn curve, but serious game with its realistic graphic, VR helmet and wearable controller can offer a really immersive experience.

3.2. 3D Environment

Our simulation evolves in a 3d emergency room scenario. User avatar (the doctor) can use all 3d objects in the room (every object is a Tool Agent) and with them interact with the patient (Patient Agent).

Other than doctor and patient, we model a nurse avatar that helps doctor in particular procedure like CRP.

Thanks to the support of SimAv doctors we created a scenario and 3d models very similar to real ones; this is the first step to overcome the wall between a real experience and a simulated one.

3.3. Open World and Game Constrains

In an open world videogame player has the illusion to make what he wants when he wants.

We use this concept in our game and thank to Game Agents, learner has not to follow a script or do his actions in a specific order to proceed in the simulation.

Our Game Agents offers a dynamic world and the learner, as in real life, seed events happen independently of his actions.

3.4. Game Mechanics and Real-life Knowledge

A serious game user could not be a video game player, we focus our game design on this truth.

For this reason, we model Tool Agents as real-life instruments, so the learner used his medical knowledge to understand how an object works and how use it.

4. CONCLUSIONS

Medical training simulators represent an important way to teach how safe a life, especially in emergency.

The challenges are: give a realistic experience, offer multiple clinical case and be easy to use.

With our software we propose a brand-new approach that offer a high customizable scenario, an immersive simulation that evolve in time with real-time events and the opportunity to be different in every iteration.
User has a high freedom degree in his actions.
Finally, our game design permit to any user to play without know particular mechanics or understand difficult GUI. His medical knowledge is the best way to lead to success.

REFERENCES

https://www.laerdal.com/docid/12984905/MicroSim-Inhospital

http://drsim.accuratesolutions.it/

Bandini, Stefania, Manzoni, Sara and Vizzari, Giuseppe (2009). 'Agent Based Modeling and Simulation: An Informatics Perspective'. Journal of Artificial Societies and Social Simulation 12(4)4 <http://jasss.soc.surrey.ac.uk/12/4/4.html>

Abar, Sameera & K. Theodoropoulos, Georgios & Lemarinier, Pierre & O'Hare, Gregory. (2017). Agent Based Modelling and Simulation tools: A review of the state-of-art software. Computer Science Review. 10.1016/j.cosrev.2017.03.001

Deterding, Sebastian & Dixon, Dan & Khaled, Rilla & Nacke, Lennart. (2011). From Game Design Elements to Gamefulness: Defining Gamification. Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek 2011. 11. 9-15. 10.1145/2181037.2181040

Darina Dicheva, Christo Dichev, Gennady Agre and Galia Angelova Journal of Educational Technology & Society Vol. 18, No. 3 (July 2015), pp. 75-88