Modeling Intelligent Agents in Virtual Worlds

Israel Guzmán-Pérez[1], Darnes Vilariño-Ayala[1], María J. Somodevilla-García[1] and Ivo H. Pineda-Torres[1],

[1] Benemérita Universidad Autónoma de Puebla

ig89852005@hotmail.com, {darnes, mariasg, ipineda}@cs.buap.mx

(Paper received on November 28, 2010, accepted on January 28, 2011)

Abstract. Virtual worlds become more popular day by day. Today in these virtual worlds are carried out various activities (cultural, educational, entertainment, leisure among others). So it is necessary to develop intelligent software entities within these environments, with the aim of providing various services to users. The purpose of this paper is to propose a model that is able to meet the needs virtual environment. Besides, a case study was presented to implement the proposed model on an avatar driven by an agent to teach the abacus, in the virtual world of Second Life.

Keywords: Virtual World, Second Life, Intelligent Agent, Avatar.

1 Introduction

For many people virtual world comes to be a work place or a place for knowledge acquisition or for entertainment. Universities have joined to the virtual worlds by offering common spaces to students and professors for education-learning process [1].

A metaverse is a virtual world. Second Life (SL) [2] is a metaverse developed by Linden Research Inc.. It was launched in 2003 and from 2006 has gained a crescent international attention. SL was inspired by "Snow Crash" science fiction novel written by Neal Stephenson and the literary movement "Cyberpunk" [2].

Agents' theory is booming from the last days and it is inside of education-learning process, financial and TAC CAT y TAC SCM [3] market simulations and information retrieval among others.

One of the most ambitious projects of the last years is the Edd Hifeng [4] and it is a result of the intelligent software development incorporation to the virtual world. A group of researchers from Rensselaer Polytechnic Institute working to change that by engineering characters with the capacity to have beliefs and to reason about the beliefs of others. The characters will be able to predict and manipulate the behavior of even human players (avatars), with whom they will directly interact in the real, physical world, according to the team [5].

There are now intelligent agents in virtual worlds, but each agent has its own model trying to fill a need or trying to provide a specific service, the intention of this paper is to propose a model which considers the most needs or services and from which it is easy to develop intelligent agents in virtual environments.

2 Background

Virtual world environments, such as Second Life have been around for some years now. After the initial hype that virtual worlds would unleash new unlimited commercial success, the focus is now on more pragmatic and serious virtual environments applications. Several authors have identified a classification for how both profit and non-profit organizations can use virtual environments, including areas such as marketing, support for mass customization, virtual markets/shopping, communication and collaboration, consumer research, innovation, virtual education and learning, and recruitment [6]. RunAlong is developing an international web community, primarily for women joggers, that is entirely designed through a user-driven innovation approach based on a series of physical user innovation workshops [7].

The largest existing worlds include South Korea's Cyworld, and its 20 million uniques, World of Warcraft with its 8 million subscribers, and Europe's Habbo Hotel with its own 7 million regular users. Also, 26 million online world users by 2011 in China alone are estimated [8]. 18 million accounts were registered in Second Life in January 2010, although there are no reliable figures for actual long-term consistent usage [9] [10].

Agents in Metaverse, built using Active Worlds, are capable of performing the tasks typically associated with human beings, such as taking tickets for rides and acting as shopkeepers. However, these agents are basically reactive agents which work in a hard-coded manner. Virtual psychotherapist ELIZA [11], designed to take care of the 'patients', is also achieved with rule-based, adeptly modeled small talk. A conversational virtual agent Max has been developed as a guide to the HNF computer museum, where he interacts with visitors and provides them with information daily [12].

Agent in the virtual world capable of providing a kind of agency for collaborating with other agents and interacting with the virtual world, and on the other hand acts as a Design Agent capable of designing and constructing dynamic virtual places for the Avatar agent as needed.

3 Proposed Model

A general model was designed so that agents to manage avatars within a virtual world can be developed following this methodology. The proposed model is a generalization of previous works on intelligent agent on Second Life [13][14].

The architecture of the proposed model is shown in figure 1 and comprises five basic modules as follows:

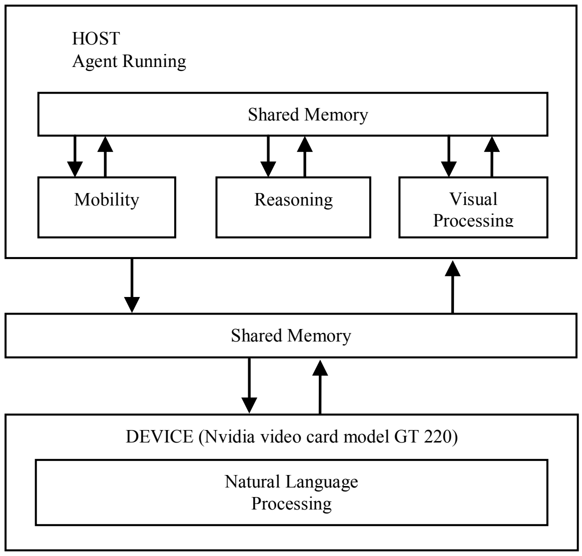

fig. 1. Model General Architecture.

Module 1 Natural Language Processing (NLP): Agent responds messages from other avatars, according the situation.
Module 2 Reasoning: Agent should be able to determine which action to carry out.
Module 3 Visual Processing: Agent processes information coming from virtual world, in order to determine the direction to be taken for the avatar which it is controlling.
Module 4 Mobility: It is used for visual processing; the agent must see what is in their environment.
Module 5 Skill(s): In this module, the agent must possess the necessary behaviors to perform a certain task (Example: teaching, dancing, public address, find information, build things, etc.). The implementation of this module can be composed of more modules, where each is designed to perform a particular task.

3.1 Modules description

Module 1 NLP
Agent should fulfill the following:

a. Text analysis
b. Audio analysis
c. An initial knowledge base and capacity to grow.
d. Analysis of sources such as internet (Google, Wikipedia, Yahoo, etc ...).

Module 2 Reasoning
Must have certain essential features such as:

a. Analysis of the environment in real time.
b. Analysis of the current target.

c. Analysis of the procedure for carrying out the objective (point b).
d. Feedback from the procedure being carried out.
e. Ability to change the procedure if the objective is not being met.

Module 3 Visual Processing
a. Analysis of the environment in real time.
   - Identification of avatars.
   - Identification of objects.
b. Analysis of real time target.
   - Locate avatars in the environment.
   - Locate avatar in the environment.

Module 4 Mobility
a. Define the displacement on the axes (X, Y) or (X, Y, Z).
b. Draw a path to the target.
c. Ability to change the way towards the target.

Module 5 Skill(s)
a. Define the skills set that the agent can develop:
b. $S = \{S_1, S_2, \ldots, S_n\}$.
c. Define the actions set that allow them to develop each skill defined:
d. $Action_j = \{Act_1, \ldots, Act_k\}\ j \in H, k \in [1, \ldots, |Action_j|]$.
e. Interact with the 4 remaining modules to carry out the corresponding actions.

4 Case Study

As a case study Israel Agent is presented. This was designed with the purpose of teaching other avatars, controlled by humans, the basic abacus operations.

4.1 Agent's modules

4.1.1 Natural language processing module

This module interacts with a knowledge database in order to process natural language. A class to permit the connection with the database was created. This class defines methods for receiving and sending messages in the virtual world.

NLP module makes a conversion from text to voice (only in Spanish), thus the agent is endowed with voice, and it makes http connections to Google and DRAE (Dictionary of the Real Academic Spanish).

4.1.2 Reasoning module

Agent Israel applies deductive reasoning to react to diverse avatar's behavior. Logic is used to codify a theory which establishes the best action to be taken for a given situation. A set of rules and a logic database were defined in order to describe the world current state and the set of actions to be performed by agent (see Table 1 for a complete set of predicates).

Table 1. Israel agent predicates.

Predicate	Description
LocateI(Isle)	Agent is located in an (Isle)
Locate(X,Y,Z)	Agent is located in (X,Y,Z)
Facing(d)	Agent is facing to direction (d)
Teach(j)	Agent is teaching abacus' operations to an avatar (j)
Leisure()	Agent is in leisure state ()
Move(X,Y,Z)	Agent is moving to (X,Y,Z)
MoveI(Isle)	Agent goes to the (Isle)
Search(j)	Agent searches an avatar (j)
Identify(u)	Agent identifies objects or avatars (u)
RMessage(m)	Agent receives messages from avatars(m)
SMessage(n)	Agent send a message to other avatars(n)

Using predicates in Table 1 possible actions of the agent are defined as follows:

Act = {move_forward, move_back, turn, write_Message, send_Message, teletransport, interact, adition, subtraction, multiplication, division}

Nine rules were defining using previous actions of the agent (see table 2)

Table 2. Israel agent rules.

Number	Rule
1	LocateI (I) ^ Leisure() ^ ¬Search(A) ^ ¬Identify(A) > Do(teletransport)
2	LocateI (I) ^ ¬Leisure() ^ Search(A) ^ ¬Identify(A) ^ Move(X,Y,Z) > Do(move_forward)
3	LocateI (I) ^ ¬Leisure() ^ Search(A) ^ Identify(O) ^ Move(X,Y,Z) > Do(move_back)
4	LocateI (I) ^ ¬Leisure() ^ Search(A) ^ Identify(O) ^ Move(X,Y,Z) > Do(turn)
5	LocateI (I) ^ ¬Leisure() ^ Identify(A) ^ RMessage(M) ^ Facing(D) ^ Locate(X,Y,Z) > Do (write Message)
6	LocateI (I) ^ ¬Leisure() ^ Identify(A) ^ Locate(X,Y,Z) ^ Teach(Op) ^ SMessage(MA1) ^ RMessage(M) > Do(sum)
7	LocateI (I) ^ ¬Leisure() ^ Identify(A) ^ Locate(X,Y,Z) ^ Teach(Op) ^ SMessage(MA2) ^ RMessage(M) > Do(subtraction)
8	LocateI (I) ^ ¬Leisure() ^ Identify(A) ^ Locate(X,Y,Z) ^ Teach(Op) ^ SMessage(MA3) ^ RMessage(M) > Do(multiplication)
9	LocateI (I) ^ ¬Leisure() ^ Identify(A) ^ Locate(X,Y,Z) ^ Teach(Op) ^ SMessage(MA4) ^ RMessage(M) > Do(division)

where:

^ means "and".

∨ means "or".

> means "then".

also:

I = {I1, …, In } accessible isles
A = {A1, … , Am} visible avatars
(X,Y,Z) = {(0,0,0),…,(a,b,c) } possible coordinates inside an isle
O = {O1, …, Ok} accessible objects inside an isle
M = {M1, …, Mp} messages from public channel
D = {north, south, east, west, northeast, northwest, southeast, southwest}
Op = {sum, subtract, multiplication, division} abacus possible operations
MA1 = {teach to sum}
MA2 = {teach to subtract}
MA3 = {teach to multiply}
MA4 = {teach to division}

4.1.3 Visual processing module

Agent Israel resides in the server and continuously tries to detect an avatar arrival in order to begin the learning process. SL provides some sensors to establish the agent's visibility range. These sensors return an avatars and objects list found in a predetermined range.

4.1.4 Mobility module

Avatar should be continuously moving inside a determined action range. In doing that, it moves from an initial point within a prefixed angle.

4.1.5 Skills module

Skills and actions set are defined as follows:

S_i = {*use of the abacus*} $i = 1$.

To achieve complete fully with any basic operation on the abacus, it performs the following actions:

ActionSet = {*establish representation on the abacus, receive message, retrieve message, detect the type of operation received, detect the basic operation to be performed*}, $j = 1,\ldots,5$.

In order to perform the above actions the algorithm 1 should be applied.

Algorithm 1

```
Step 1: At first the agent sets the quantity represented on the abacus
to 0.0
Step 2: The agent receives and retrieves the message (message, type).
  If (type = abacus)
     Retrieve value (bead)
     Update environment variables
  Else
     Using NLP module,
```

```
    Analyze the message
    Determine operation type
     (addition, subtraction,
      multiplication, division or
      representation)
  If (operationtype=representation)
      Extract from the message the
      number to represent it, and
      Then
        reset back the abacus.
       Else
        represent the operation, by
        moving the bead and follow
        the abacus rules
```

To understand the movement of the beads in the abacus (specified in step 2 of algorithm1), you need to know that it is divided into two sections [15]. It has a section that only has two balls, called heaven-beads, and another section which has more than 5 balls, known as earth-beads. Importantly, the abacus has 49 beads, spread over 7 columns. Each earth-beads in the first column has a value of 0.01, and each heaven-beads a value of 0.05, the second column represents the 0.1 to 0.5 for earth-beads and heaven-beads. To perform the operations of addition and subtraction are used 17 rules, some of which have been modified to use the maximum number on the abacus (166,666.65), adding the use of end heaven-beads and end earth-beads. The same problem arises when performing subtraction, multiplication and division so they make similar changes to the general rules [14].The third column represents the unity; and so on both types of beads increase their value by multiplying by 10. One important thing to note is, that the first time that the agent detects that it will carry out an operation, i.e. addition, subtraction, multiplication or division, is preparing to retrieve the operator (s) and / the operator (s), operator can be given in numerical or written form. So that, the agent can interpret what is written, a table is generated, only once, and then maintained throughout the life cycle of the agent. In this table are stored numbers from 0 to the 9, and a mechanism is created for generating the rest of the numbers.

4.2 Experimental results

To debug the agent, OpenSim [16] server was installed because of its resemblance to Second Life and the final tests were executed inside Second Life agent having the same results within OpenSim. In a computer OpenSim server was installed and it was configured everything necessary for proper operation. Another computer was operated another Second Life viewer controlled by a human, to carry out local tests while the agent was running on another computer. The following describes only the most important tests, all that is not mentioned is because it worked properly and efficiently.

4.2.1 Visibility and mobility

With respect to visibility messages are sent between the agent and sensors. The arrival of these messages is asynchronous, there are two factors that influence this communication, first depend on the speed of your Internet connection and also depend on the speed of the server to run scripts. In testing the agent had an average yield, due to the two previously mentioned factors, with a slight delay in the update of the sensors, which caused a slight displacement between the avatars (see Figure 2). Another important factor is the time it takes the agent to process the information received by the sensors because the sensors send a message with a specific format which is then processed by the agent. The sensor detects the avatars send a message with the format “X Y Z | Avatar1Name X Y Z | Avatar2Name X Y Z | … | AvatarNName X Y Z”, As can be seen is a list of names of avatars and their positions, except the first element does not contain name, as is the current avatar position Israel as the character delimiter "|" between each avatar. The objects responsible for detecting sensor sends a list of objects detected in a radius of one meter from the avatar Israel. Once the agent retrieves the sensor information is passed to the respective module and this triggers a series of actions such as re-routing of the avatar to dodge obstacles. One very important thing to mention is that these objects that are an obstacle may be in motion at a certain speed or variable speed or just static, i.e. no movement. This is because in the virtual world can be built motorcycles, cars, airplanes, beads, dogs, cats, etc.., i.e. they are in motion.

Fig. 2. Avatar Israel (man shaped) controlled by the agent. The agent is trying to place near avatar Israel and avatar Darnes (woman shaped).

4.2.2 Natural Language Process (NLP)

To address each message in the virtual world a thread is created, and so, N threads are created to address the messages received by objects and avatars. The creation of these threads is dynamically and the number of active threads depends heavily on the power of the computer you are using to run the agent. Therefore, the agent can respond to multiple messages at one time, but not necessarily in the order they were received, since each thread consumes a different time depending on the information to be processed (see Figure 3). Although the agent is time consuming to connect to the Internet, the results are satisfactory. Example of this, is that the agent can answer questions like "*what color is Napoleon's white horse*" the answer given is "*is white because are questions of mental agility and trick*" as you can see the correct answer is just "*is white*", but the answer given by the agent's response was found on the Internet using Google, and this is a very good approximation to the real answer.

Despite *eSpeak* [17] [18] was used to give voice to the agent, still it has errors in pronunciation, but these can be fixed by improving *eSpeak*, as it is open source.

Fig. 3. Avatar Israel (man shaped) controlled by the agent. The agent responds an avatar Darnes question where avatar Darnes is controlled by a human.

It is important to note, the Second Life viewer [19] makes revisions on the state of the display of the agent to see if there is one person handling it and avoid it to be expelled. The agent to deceive the Second Life viewer reviews must occasionally send keystrokes to trigger events that produce animations, which are associated with a function key on the keyboard, so it tricks the viewer at the time of make revisions. For example:

1. When the agent sends the message "hello" then triggers an animation that makes the avatar move an arm in greeting.
2. When the agent will represent some number on the abacus, this triggers an animation of the agent pointing to the abacus.
3. When the agent is putting together a message activates an animation of the avatar "writing" and sent the message after the animation stops.
4. When the agent is accessing the Internet makes an animation of thinking.
5. When the agent is doing a sum and will represent the number on the abacus was made an animation of marking.
6. When the agent cannot answer a question makes a movie of frustration.

All these animations are made using different applications, such as Avimator (See Figure 4) and QAvimator (See Figure 5).

The main difference between Avimator and QAvimator is that QAvimator allows greater control over the movements of the avatar, while Avimator have less control on the movement of the avatar.

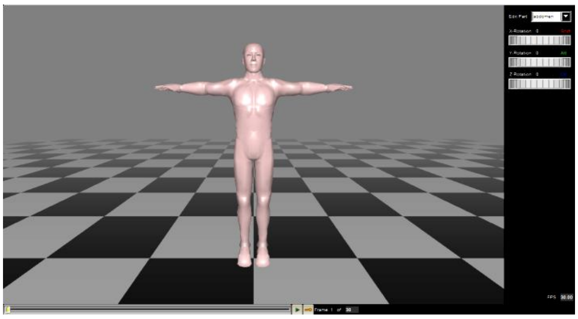

Fig. 4. Dummy avatar that shows how animation is made, moving or rotating any part of the avatar as are the arm, abdomen, legs and head.

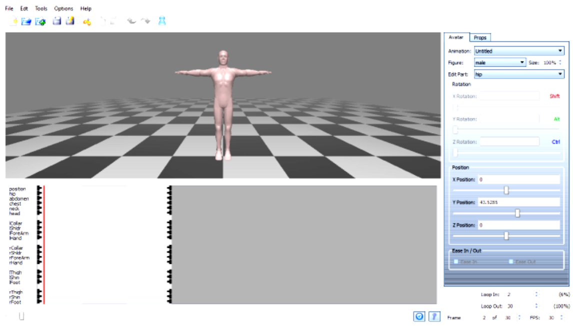

Fig. 5. Greater control over the movement of the avatar using QAvimator.

4.2.3 Using the abacus

EasyTalk [20] protocol is employed to use the abacus, which allows the agent to know the status of the abacus (see Figure 6). The drawback is the time it takes to update the state, since it depends on two factors, one is the speed Internet connection and the other is the time it takes to run the abacus' scripts. The later time depends entirely on the Second Life server and cannot be measured. However, the agent is able to update the status of the abacus, but due to this delay, not controllable by him, on occasion may misinterpret the status. The state of the abacus is updated by the agent considering multiple messages.

Fig. 6. Israel (man shaped) controlled by the agent. Agent is representing the number 5 on the abacus.

5 Conclusions and Future Work

A general model was designed so that agents to manage avatars within a virtual world can be developed. In particular we have shown the results of applying the model on an avatar driven by a management agent to teach the abacus, in the virtual world of Second Life. It is important to note that for the development of such applications is necessary to have a great amount of computational resources.

The module of natural language processing, was extended for including Internet search, in order to provide better results, but the agent's response time was affected significantly. Three different versions of the agent presented in the study case were developed, which differ substantially in how they implemented the natural language processing module.

Version 1, which is the more stable is the one described in this paper. However, the other versions have served as a comparison. Version 2 uses Gecko [21] for connections made by the agent to the Internet, when a response is requested by an avatar handled by humans. This is much more reliable but requires more computational resources. Version 3 takes DLL to connect to the database, which causes them to consume more run time and resources.

In future work is proposed to replace the current vision system which is based on sensors for 3D vision system. This improvement allows agent teleportation be enhanced since agents no longer depend on objects programmed into LSL and thus can teleport between metaverse using the protocol OGP [22].

Besides, it is proposed to improve the natural language processing, adding context analysis, this no doubt that the agent will take longer to respond, but would improve the results in the response.

Also consider that can improve response time [14] using CUDA technology from NVIDIA [23], for parallel computing, because the Second Life viewer does not use it. This module could add support for multiple languages, this can be achieved using the application *solicitudgetCLR* also the agent must connect to the Google translator, and so do the translation of the sentences received a certain language. It also aims to improve the implementation *solicitudgetCLR* (in charge of links to Google and the SAR) to make a better analysis of Web pages or the other option is to completely replace that part of Gecko [21]. Gecko undoubtedly is the best option, but requires a greater amount of computational resources.

References

1. "Institutes in Second Life". [Online] 2010. http://secondlifegrid.net/casestudies.
2. "Definición de Metaverse". [Online] 2010. http://en.wikipedia.org/wiki/Metaverse.
3. TAC Trading Agent Competition Homepage. [Online] 2010. http://www.sics.se/tac
4. "Enseñan a pensar a un avatar de Second Life". [Online] 2010. http://www.tendencias21.net/Ensenan-a-pensar-a-un-avatar-de-Second-Life_a2293.html
5. "Bringing Second Life To Life: Researchers Create Character With Reasoning Abilities Of A Child". [Online] 2010 http://www.sciencedaily.com/releases/

A. Barnetta. "Fortune 500 companies in Second Life, Master Thesis". [Online] 2010 http://www.smi.ethz.ch/education/thesis/Barnetta_SecondLife.pdf

6. RunAlong. [Online] 2010 http://www.runalong.se/
7. Paul R. Messinger, "A Typology of Virtual Worlds: Historical Overview and Future Directions", Journal of Virtual Worlds Research, Austin, USA, pp. 1-18, 2008.
8. "Second Life" [Online] 2010 http://en.wikipedia.org/wiki/Second_Life.
9. Ranking de mundos virtuales. [Online] 2010. http://gigaom.com/2007/06/13/top-ten-most-popular-mmos/
10. Yilin Kang, "Self-Organizing Agents for Reinforcement Learning in Virtual Worlds" in Proceedings of WCCI 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain July, 18-23, pp. 3641-3648, 2010.
11. S.Kopp, L. Gesellensetter, N.C.Krmer, and I.Wachsmuth, "A Conversational agents museum guide-design and evaluation of a real-world application", IntelligentVirtualAgents, pp.329–343, 2005.
12. Israel Guzmán P. , Darnes Vilariño A., Fabiola López L., Maria J. Somodevilla, Mireya Tovar V., Beatriz Beltrán M. ¨ "ISRAEL: An agent inside second life virtual world "Digital Scientific and Technological Journal, Editorial Amate, pp 226-231, 2009.
13. Israel Guzmán, Darnes Vilariño, María J. Somodevilla," An Intelligent Agent who teaches the Basic Operations with an Abacus inside Virtual World Second Life", Research in Computing Science, Advances in Soft Computing, Vol 49, 2010, pp 165-176.
14. How to use an Abacus. [Online] 2010. http://www.educalc.net/144267.page
15. What is OpenSim?. [Online] 2010. www.opensimulator.org, 2010
16. Speech Synthesizer. [Online] 2010. http://espeak.sourceforge.net/
17. Voice over Internet Protocols. [Online] 2010. http://www.asterisk.org/, 2010
18. Herramientas para compilar en Windows el visor de Second Life. [Online] 2010.
19. http://wiki.secondlife.com/wiki/Microsoft_Windows_Builds
20. Protocol EasyTalk, [Online] 2010. http://wiki.secondlife.com/wiki/LSL_Protocol/EasyTalk
21. Motor de Navegación para Firefox. [Online] 2010. http://developer.mozilla.org/es/Gecko.
22. Protocol OGP. [Online] 2010. http://wiki.secondlife.com/wiki/OGP_Base.
23. Cálculo Paralelo de NVIDIA utilizando la tecnología CUDA. [Online] 2010. http://www.nvidia.es/object/cuda_home_new_es.html
24. Framework para desarrollo de Agentes en JAVA, [Online] 2010. http://jade.tilab.com/
25. Bringsjord S., Shilliday A., Clark M., Werner D., Taylor J., Bringsjord A., Charpentier E.. "Toward Cognitively Robust Synthetic Characters in Digital Environments". In Artificial General Intelligence Proceedings of the First AGI Conference, 2008.