A STUDY ABOUT HOW TO COMBINE INTERFACE ANIMATION WITH ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE IN *HELPBOT* CHATTERBOT

Andres Jessé Porfírio, Ewerton José Wantroba, Lucélia de Souza

Universidade Estadual do Centro-Oeste (UNICENTRO) Cx. Postal 730– 85015-430– Guarapuava – PR – Brazil

andresjesse@yahoo.com.br, ewerton.irati@gmail.com, lucelia@unicentro.br

Abstract*. This paper concerns a study about adding animation to interface in chatterbots, more precisely in the chatterbot HelpBot, which operates in the help context of interaction in collaborative ambients. This chatterbot uses the markup language of Artificial Intelligence (AIML) in its knowledge base. Some interface animation technologies were studied and a comparison of these tools is presented. The tool Jmonkey was utilized to separate the bot´s conversations from the animation of graphical interface. The aggregation of the graphical interface in the HelpBot was intended to simplify the interaction between users and the chatterbot, thus making the interface more attractive for conversations.*

Keywords. *Chatterbots; AIML; VHML; Interaction User-Bot; On-line Help.*

1. Introduction

There are several ways of interaction between users and computers, from input command to touch screens. Computational agents discern among these interactions, and are called chatterbots or just bots [3]. Bots are able to interact with users using natural language dialog, as proposed by Allan Turing in 1950 [7].

Currently, chatterbots are developed using Artificial Intelligence Markup Language – AIML [21]. AIML is a specialized markup language for artificial intelligence applications [19,20] that arises from Extensible Markup Language (XML).

Furthermore, AIML is used to manipulate documents which contain structure information.

Animated graphical interface could be used to make the interaction between users and bots more attractive. This kind of interface is designed for several purposes, such as use in games, creation of three-dimensional (3D) ambient and movies. Among many other applications, these sort of tools allow the building of 3D characters, including their movements, gesticulations and animation [2].

This paper describes a study about technologies that *can be applied to develop animated graphical interfaces to bots and also presents a com*parison of them. Besides this introductory section, Section 2 shows a state of art about bots, Section 3 describes technologies that can be utilized in interface development and Section 4 provides a comparison of them. Section 5 presents an aggregation example of animation interface of chatterbot called *HelpBot* (developed by the authors of this paper), which acts as an on-line help for users of collaborate systems [19], and finally, section 6 contains conclusions and future perspectives.

2. Chatterbots

In the literature three chatterbots are considered milestones in the development of the current generations of bots: Eliza, Julia and Alice. The first generation was characterized by the bot Eliza [22] which was utilized by psychologists to diagnose their patients in the 1960s. Eliza does not use artificial intelligence techniques, only grammatical rules. The chatterbot Julia [5] characterizes the second generation of bots. Julia presents a neural networks use and thereby makes possible a more approximated dialog of natural language.

Alice is a chatterbot that belongs to the third generation [21] and like current bots uses AIML language. Furthermore, since the development of Eliza, chatterbots have presented a more persuasive kind of dialog which is able to remember talk, such as names of users, favorite kinds of foods, etc.

Nowadays, animated graphical interfaces that stand out are Fake Captain [4] developed by the Alice Foundation and Guile 3D [6]. The next section describes technologies currently utilized to animate bots interfaces.

3. Technologies of interface animation

To develop a dynamic and attractive interaction between users and chatterbots, this paper examines a panorama of technologies which are utilized to provide three-dimensional rendering in web ambient. These technologies were researched on the Internet as possibilities to develop a graphical interface to *HelpBot* and a comparison of them is presented in Chart 1. Despite the fact that this paper prioritizes freeware technologies, some of them have a commercial license such as: Xith3D [23], Shout3D [17], Sandy 3D Engine [16], MS Agents [13], Java 3D API [9] and Jmonkey Engine [10].

3.1 Xith3D

Xith3D is a Java-based tool that has open source code and is intended to help in the creation of electronic games, visualization of 3D objects and ambient prototypes. This tool uses several graphical resources such as rendering three-dimensional objects, support of many file types (ASE, OBJ, MD2, MD3 and others), textures and animations.

Nevertheless, there are disadvantages associated with using Xith3D, such as the fact that resources must be utilized with this tool, such as canvas, applet, etc. Furthermore, Java Runtime Environment (JRE) is necessary on the client machine.

3.2. Shout3D

Shout3D is a Java-based tool that allows the creation of animations in which 3D content can be developed using a software such as 3D Studio MAX [1], Lightwave3D [14], Maya [12] or Nendo [8].

This tool builds a real-time 3D render inside a Java applet, therefore being able to run in any web browser. Besides the render, using Shout3D it is possible to create

movies and animations, since they do not need interaction with users. After creating movies and animations, these files can be exported using the format of Virtual Reality Modeling Language (VRML) and run in a web browser through Shout3D software.

Shout3D can be executed through a Java applet and consequently makes possible a switch of information, such as movements, gesticulations and speech between the web browser and animations. Therefore, the interaction with other applications such as chatterbots is simplified.

There are two versions of Shout3D: one of versions is commercial and the other is not. However, one disadvantage of the free version is a horizontal line that the tool exhibits during all the execution time (Figure 1).

Figure 1: Render using Shout3D (free version), with the horizontal line of identification in detail.

3.3. Sandy 3D Engine

Sandy 3D is a Flash-based tool which is compatible with the script languages ActionScript 2, ActionScript 3 and Haxe. Some of its important characteristics are its capability to open file types such as MD2, ASE, 3DS and Collada and to render objects that contain animations and textures.

This tool presents many interesting resources, for example, the display of 3D objects with textures, surface with transparency and one primitive system of animation.

Moreover, Sandy 3D contains a system of detection and treatment of collision that is very important when working with electronic games.

One of the main advantages of using this tool is its portability because nowadays most systems are compatible with the Flash platform. Figure 2 shows a 3D textured object exhibited by a web browser using the Sandy 3D tool.

All this tool's graphical elements are processed and rendered without use of the Graphics Processing Unit (GPU), in other words, the calculations are performed by the Central Process Unit (CPU). Thus, there are disadvantages associated with this use as it has a lower performance than Java-based tools, mainly in the case of machines without high processing power. Additionally, the installation of Flash Player is required and video hardware is not necessary in the render process.

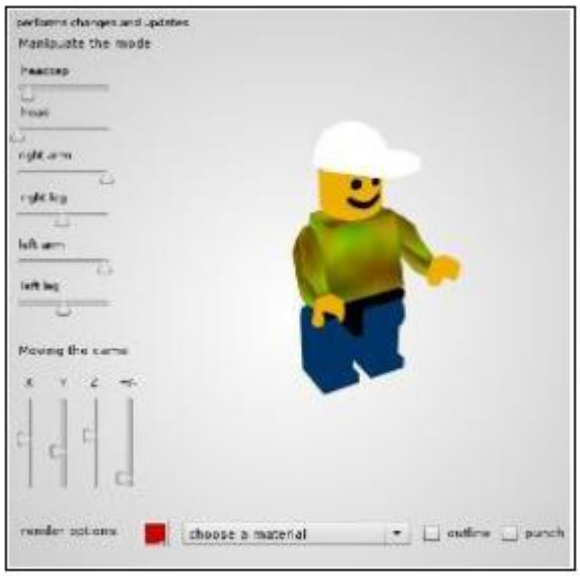

Figure 2: Rendered 3D object using Sandy 3D Engine.

3.4. MS Agents

The animation agents of Microsoft (MS Agents) consist of tools that allow the creation of animated interfaces besides having resources such as voice recognition, synthesizing of texts and animated movements [11].

To use MS Agents it is necessary to install at least one Agent in the user computer. Agents are easy-use tools and all the necessary material to utilize them can be

downloaded from the official Microsoft website[1]. Figure 3 shows the Agents available from the website: Genie, Merlin, Peedy and Robby.

Figure 3: MS Agents.

One advantage of this tool is the possibility of it being utilized in desktop applications as well as the web ambient; however installation is needed in both of these cases.

On the other hand, some disadvantages exist when using MS Agents, for example, due to pre-programmed animations the application could be limited to them and there is a limitation of the operating system as well. Besides these disadvantages, in the cases of the web server, clients must install the Agent.

3.5. Java3D API

Java3D is an Application Programming Interface (API) that is becoming popular because it is used to develop many applications, among them 3D games. This API is a high level layer which dispenses developers of complex tasks such as memory management, optimization and rendering. Thus, due to these resources Java3D simplifies the development process.

Differently from OpenGL [15], that works using the concept of state machine, Java3D offers developers a group of classes and methods for 3D programming. Consequently, this tool provides more intuitive programming for developers.

Nevertheless, Java3D presents a lower performance than other APIs. Generally, applications that are developed with C/C++ which use OpenGL or DirectX are faster.

[1] (http://www.microsoft.com/products/MsAgent/main.aspx)

3.6. Jmonkey Engine

The Jumping into jME tool, called Jmonkey Engine is Java-based and presents resources for the development of graphical applications, including simulations and electronic games. Among these resources are the capability to open several file types (including MD2, OgreMesh and ASE), rendering 3D objects with textures, predefined keyframe animations and dynamic animations using bones.

Applications built using Jmonkey Engine show great performance due to utilizing video hardware (GPU) to render scenes. These applications can be easily added in web pages using applets.

One disadvantage of Jmonkey Engine, as well as Xith 3D and Java 3D API tools, is that Java Runtime Environment (JRE) is required on the client machine.

3.7 Unity 3D

Unity 3D is a tool that is focused on video games development. Although this tool is a professional form of technology, a free version is available for non-commercial use. There are versions for many platforms, including PC, MAC, Web, Iphone e Nintendo Wii. In the case of Web platform, a little plugin (UnitPlugin) must be installed to enable the web browser to recognize three-dimensional scenes inside web pages.

This tool offers state-of-the-art rendering resources, providing the creation of applications with effects without affecting performance, due to the fact that processing of scenes is done with CPU and GPU resources together. Unity 3D has an editor which allows the easy creation and management of 3D objects, sounds, textures, etc. In addition, through this editor it is possible to edit and visualize scenes in real time (Figure 4).

The resources of Unity 3D compose a complex system that manages animations and also allows working with keyframes and bones. This tool is able to import files of several types and so ensure compatibly with software such as Blender 3D, Maya and 3D Studio Max. This tool also contains one system of script which recognizes Java, C# and Boo languages.

Figure 4: Scene editor of Unity 3D.

Among the disadvantages of Unity 3D we found that the free version has limitations on special effects such as lighting and post-processing (filters applied after the scene has been rendered). Another problem is that the free version also displays the logo of Unity 3D in projects. Finally, a plugin is supposed to be installed on client machines.

4. Comparison of interface animation technologies

Chart 1 presents a comparison of the animation interface technologies that were studied and it is based on the following criteria: software license, ease of use, supported formats for importing animations, ease of integration with web ambient and limitations as well.

Chart 1: Comparison of interface animation technologies.

Tools/Criteria	License	Ease of use	Supported Formats	Integration with web	Limitations
Xith3D	Free and Open Source.	Easy to use.	· ASE · AC3D · OBJ · 3DS · MD2 · MD3 · MD5 · Cal3D · COLLADA 1.4 ·Quake3/Half-Life ·BSP	Possible with Java applets.	Java Runtime Environment (JRE) must be installed on the client computer.
Shout3D	*Shareware*	Good documentation.	·WRL ·S3D	Possible with Java applets.	All projects built with the free version display the logo of Shout3D.
Sandy 3D Engine	Free and Open Source.	Good documentation and large amounts of tutorials.	· 3DS · COLLADA · ASE · MD2	Possible using Flash player.	Flash Player must be installed. It does not use video hardware to render.
MS Agents	Free.	Pre-defined animations.	None.	Possible installing the Agent on client machine.	This tool depends on the Windows platform. Few animations. Restricted functions.
Jade 3D API	Free and Open Source.	Available in Java native, it does not need to install other tools.	· 3DS · OBJ · VRML ·X3D ·NWN ·FLT	Possible with Java applets.	JRE must be installed on the client computer.
Jmonkey Engine	Free and Open Source.	Integration with several systems including Web.	· Milkshape · MD2 · MD3 · ASE · XML ·DAE ·JME · OBJ	Possible with Java applets.	JRE must be installed on the client computer.

			· 3DS · X3D		
Unity 3D	Free for non-commercial use.	Creation/Edition of scenes in graphical mode, it imports scenes of many 3D editing types of software.	. MAX . JAS · MAYA · C4D · BLEND · Carrara · Lightwave · XSI 5 · SketchUp Pro · Wings 3D · 3DS · OBJ · Drawing Interchange · DXF · FBX	Possible using a plugin in the web browser.	The free version has limitations on special effects such as lighting. Built projects with the free version show the logo of Unity 3D. A plugin must be installed on the client computer.

5. Joining animation in the interface of *HelpBot*

Like most existing chatterbots in the literature, *HelpBot* was developed to web ambient. Furthermore, the knowledge base of this tool contains AIML files, thus having answers for inputted stimuli of users.

To join animated graphical interface in a developed chatterbot to the web, the knowledge base is supposed to contain actions and animations which will be performed during conversations, such as moving the body, the mouth, or talking. These actions must be preferentially stored inside categories in the AIML files that contain all knowledge of the tool. The chatterbot performs actions such as moving arms or pointing to somewhere on the screen in a synchronized and orderly manner.

Concerning the application in *HelpBot*, the authors preliminarily intended to develop corporal animations. Chart 2 displays one example of an AIML category that contains, besides an answer to stimulus, the animation of a chatterbot, considering a talk when the user says "*Raise your arms*" and the bot performs the bodily action and then asks the user "*What else do you want me to do*?".

Chart 2: AIML Categories which have answers and animations in Chatterbot.

```
<category>
    <pattern> RAISE YOUR ARMS </pattern>
    <template>First I will raise my right arm ( raise the right arm for 3 seconds).
(put the right arm down) Now I will raise my left arm (raise the left arm for 3
seconds). (put the left arm down). What else do you want me to do?</template>
</category>
```

Observing Chart 2, besides the tags of AIML (<category>, <pattern> and <template>), there are special tags that are necessary to store chatterbot actions, such as some parameters, for example, time (in seconds). There are also other parameters which allow animation to be closer to reality, such as smile intensity, number of times the bot will move its head to agree or disagree with something.

5.1 AIML and VHML

One existing proposal to store chatterbot actions using a hierarchical structure is Virtual Human Markup Language (VHML). As well as AIML, VHML is a markup language that uses XML's pattern [18]. This language is divided into sub-languages: Emotion Markup Language – EML; Speech Markup Language - SML; Facial Animation Markup Language - FAML; HyperText Markup Language - HTML; Body Animation Markup Language - BAML; Dialogue Manager Markup Language – DMML [19].

VHML contains tags in order to store chatterbot animations; it is an excellent option to integrate with interfaces of bots, because this also uses the same XML pattern as AIML, which is responsible for storing knowledge of chatterbots.

This language allows a chatterbot to better express what it is talking about and thus acting bodily according to the question. For example, if one user asks: "How are you?" or "Do you like to fly on a plane or not?", the bot can answer expressing its emotions using the tags <happy> and <fear> (Chart 3).

Chart 3: VHML tags (happy and fear), inside an AIML document.

```
<category>
        <pattern>ARE YOU HAPPY TODAY?</pattern>
        <template>
                <happy> Yes, I am happy so much</happy>
        </template>
</category>
<category>
        <pattern> DO YOU LIKE TO FLY? </pattern>
        <template>
                <fear> I am afraid of flying </fear>
        </template>
</category>
```

For a chatterbot to be able to run VHML expressions, it is necessary to develop a parser that separates VHML tags from AIML tags, before the chatterbot can answer the user's questions.

In the case of *HelpBot*, the parser is currently being developed; however, at present it is considering only a few bodily animations. The written part of the answer (AIML tags) will be presented on the screen, while the animated part that belongs to VHML tags will be passed as a parameter to an animated interface that was developed using Jmonkey. This tool was chosen due to its characteristics which were mentioned earlier. Therefore, besides *HelpBot* replies to users' questions, this bot expresses the answer with some bodily movement, thus it is able to be more similar to a human being and it makes the interface more attractive.

6. Conclusions and Future Perspectives

Currently, applications are becoming more attractive and more concerned about making interactions with users more satisfying. In this context, chatterbots can be a differential in current applications because they utilize natural language. Nevertheless, chatterbots must be developed with consideration for the best possible method of user interaction.

As mentioned earlier, there are a great number of tools which allow the creation of animated characters that can make them similar to humans, but they lack the autonomy to dialog. On the other hand, chatterbots which are developed using AIML

language have excellent dialog, however the main concern isn't the interface. The connection between AIML and animated interface makes them closer to humans and thus it induces users to think that they are talking to another person. Among research tools, some have limitations or difficulties relative to integration with chatterbots or web ambient. However, these tools are an excellent option to project and develop animated interface of bots.

Special tags for movements, expressions and other actions are a good way to integrate AIML language with chatterbot's animation. These tags are not present in AIML and they need to be created. In this context, VHML is an interesting option because it uses the same structural pattern as AIML (XML pattern), besides which it is a specialized language for the animated actions of chatterbots.

Some disadvantages of using VHML are its limited specifications and the necessity to have a parser to interpret tags of this language. Consequently, development of a parser is required to separate knowledge from animation presenting in AIML files that contain VHML tags.

In the case of *HelpBot*, a parser is being developed to separate talks from graphical interface animations using Jmonkey software. This tool was chosen for presenting characteristics that combine with the bot in question.

References

[1] Autodesk. 3D Studio Max. Available: http://usa.autodesk.com/adsk/servlet/pc/index?id= 13567410&siteID=123112. Accessed January/10.

[2] R. Boulic and O. Renault (1991) “3D Hierarchies for Animation”, In: New Trends in Animation and Visualization, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons ltd., England.

[3] Chatbots. Available: http://www.chatbots.org/. Accessed November/09.

[4] Fake Captain Bot. Available: http://showcase.pandorabots.com/pandora/talk?botid=d10d53a63e345abf&skin=iframe Accessed January/10.

[5] L. N. Foner. (1993) What’s An Agent, Anyway? A Sociological Case Study. Agents Memo 93- 01, Agents Group, MIT Media Laboratory.

[6] G. Lindroth Filho. "Guile 3d". Available: http://www.guile3d.com.br/chat/bot.asp Accessed March/10.

[7] A. Hodges. (2000) Alan Turing: the enigma. First Edition. Walker & Company. New York.

[8] Izware. Nendo. Available: http://www.izware.com/nendo/index.htm. Accessed February/10.

[9] Java 3d API. Available: http://java.sun.com/javase/technologies/desktop/java3d/. Accessed February/10.

[10] Jmonkey. Jmonkey Engine. Available: http://www.jmonkeyengine.com/. Accessed February/10.

[11] F. C. Machado. (2005) Integrando chatterbot e agente animado de interface em um ambiente virtual de aprendizagem. Monografia apresentada à Diretoria de Pós-graduação da Universidade do Extremo Sul Catarinense-UNESC, para a obtenção do título de especialista em MBA em Banco de Dados.

[12] Maya. Maya 3d. Available: http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=13577897. Accessed February/10.

[13] Ms Agents. Agentes da Microsoft. Available: http://www.microsoft.com/products/MsAgent/main.aspx. Accessed February/10.

[14] NewTek. LightWave 3d. Available: http://www.newtek.com/lightwave/. Accessed January /10.

[15] OpenGL. Available: http://www.khronos.org/opengl/ Accessed February/10.

[16] Sandy3d. Available: http://www.flashsandy.org/ Accessed February/10.

[17] Shout3d. Available: http://shout3d.net/. Accessed February/10.

[18] VHML. Virtual Human Markup Language. Available: www.vhml.org. Accessed January/10.

[19] E. J. Wantroba, J. Ratusznei, L. Souza, S. M. G. S. Venske. (2008) Um exemplo de uso do padrão XML na definição de uma linguagem especializada para a Inteligência Artificial. Revista Publicatio UEPG - Ciências Exatas e da Terra, Agrárias e Engenharias, Vol. 14, Nº 03.

[20] E. J. Wantroba, L. Souza. (2009) Investigação de uso da Virtual Human Markup Language em conjunto com a Artificial Inteligence Markup Language em Chatterbots. Semana de Integração Ensino Pesquisa e Extensão. Available: http://eventos.unicentro.br/siepe2009/cd/pdf/resumo_999.pdf. Accessed January/10.

[21] R. Wallace. From Eliza to A.L.I.C.E. AI Foundation. Available: http://www.alicebot.org/articles/wallace/eliza.html. Accessed November/09.

[22] J. Weizenbaum. (1966) "ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine," Communications of the Association for Computing Machinery 9. Pages 36-45.

[23] Xith3d. Xith3d project. Available: http://xith.org/. Accessed January/10.