Evolutionary Virtual Agent at an Exhibition

Jean-Claude Heudin

International Institute of Multimedia, Pôle Universitaire Léonard de Vinci, Paris – La Défense, France
Jean-Claude.Heudin@devinci.fr

Abstract. This paper describes the Evolutionary Virtual Agent (EVA) prototype designed as an interactive entertainment system for a multimedia exhibition. This implementation uses a behavioral engine based on a dynamical subsumption architecture and a 3D animated interface that has been projected onto a physical model.

Keywords: virtual creature, intelligent agent, conversational character, physical model.

1 Introduction

The Evolutionary Virtual Agent (EVA) is a software architecture for designing self-animated conversational characters for applications requiring a human-like interface [1]. Autonomous virtual characters are supposed to respond to human interaction in real-time with appropriate behaviors: no repetitive predetermined answers, broad in content, highly contextual and behaviorally subtle [2]. The character must appear as a living creature. It must also appear to think, make decision and act of its own volition [3]. This can be achieved by giving the artificial character a set of behaviors and features such as an identity, a backstory, a role, etc. [4]. In the framework of a multimedia exhibition, it is also important to create a more attractive visual interface than a classical computer screen. The idea is to make the experience more "real" by implementing the virtual autonomous character as a 3D physical creature.

In this paper, we present a multimedia prototype based on the EVA intelligent agent technology that has been shown to visitors during a two weeks exhibition in March 2007. We begin in section 2 by describing the EVA architecture and the approach for programming such a virtual creature. In section 3, we present the 3D virtual and physical interfaces that have been designed for the exhibition. We give some experimental results and discuss them in section 4. Finally, in the last section, we provide conclusions and identify future works.

2 Evolutionary Virtual Agent

2.1 Architecture Overview

An EVA agent consists of two software components and a set of multimedia interfaces (see fig. 1). The *behavioral engine* is the "brain" part of the architecture. It

T.G. Wyeld, S. Kenderdine, and M. Docherty (Eds.): VSMM 2007, LNCS 4820, pp. 154–165, 2008.

is responsible for all perception, decision and action behaviors. It has access to perception and action interfaces through a *networking server*. This component enables to connect set a of interfaces to the behavioral engine. It includes speech-to-text and text-to-speech agents, one or more text-based “chat” clients, and an animated 3D facial model. More interfaces could be added such as webcams or other sensors depending on the application requirements.

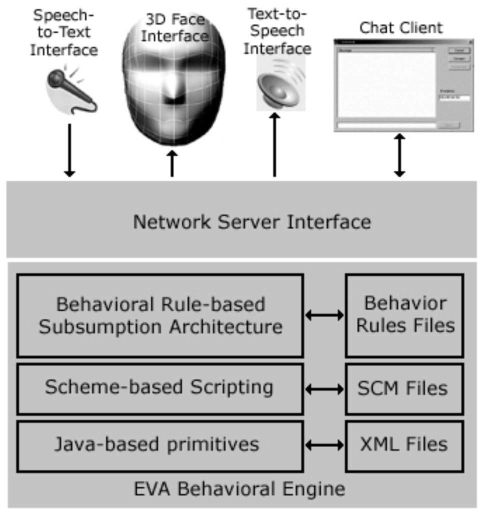

Fig. 1. The EVA architecture includes a behavioral engine and a network server for connecting various interfaces such as text-to-speech, speech-to-text, 2D or 3D animated faces, and text-based windows for interacting with the virtual creature

The behavioral engine architecture consists of three main software levels [5]. The first level is composed of a set of Java-based core primitives implementing the essential EVA’s “brain” features such as natural language processing, rule-based system, web mining, emotional engine etc. The second level is a tiny Lisp interpretor based on the Scheme language [6]. This component integrates the core primitives in a user-friendly scripting language. The third level is composed of a dynamical layered architecture inspired by the subsumption architecture [7]. Each layer is encoded as a set of behavior rules that take advantages of the underlying level features. The next figure gives the principle of the EVA’s emotional model. This model is implemented as part of the artificial metabolism in the core primitives.

2.2 XML Knowledge Files

The natural language processing features of EVA requires knowledge files that provide several information such as discussion topics and template expressions. These information are stored in XML files. The syntax of these knowledge files is close to the Artificial Intelligence Meta Language (AIML) [9]. The next code example gives an overview of the syntax:

```
<topic name="generic">
        <category name="HELLO">
                <key>hello</key>
                <key>hi</key>
        </category>
        <template name="BYE">
                <expr>goodbye *</expr>
                <expr>see you soon</expr>
        </template>
        <rule name="R1">
                <get>
                        <key>you're</key>
                </get>
                <set>
                        <exp>you are</exp>
                </set>
        </rule>
</topic>
```

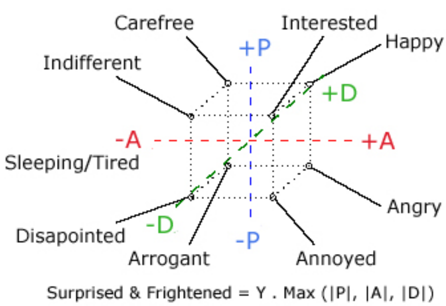

Fig. 2. The emotional system of the EVA architecture is based on the Pleasure-Arousal-Dominance (PAD) model [8]. The main emotional states are mapped in the PAD three-dimensional space. A more complete description of this model has been discussed in a previous paper [1].

The `<topic>` tag defines the global topic of the discussion. Each `<category>` tag includes a set of keywords used for analyzing user's sentences by finding occurrences of these keywords. A `<template>` tag defines a set of expressions used for generating answers in a particular discussion context. Topics may also include re-writing rules which are applied on the user's entries. In our example, the rule "R1" systematically replaces "you're" by "you are" in any expression.

2.3 Scheme-Based Scripting

In addition to XML knowledge files, the EVA architecture enables to script behaviors using the Scheme programming language. Here is an example of a simple script to illustrate the syntax:

```
(load-topic "xml/generic.xml")
(define *categories*)
(set! *categories* (get-categories "GENERIC" *in* ))
```

This program first loads the topic “generic” from an XML file. It then defines a global variables named `*categories*` and extract categories using the loaded “generic” topic. The function `get-categories` is a typical example of java-based core primitives that implement natural language processing.

2.4 Subsumption Levels

The EVA architecture is basically a dynamical subsumption model. There are 10 layers numbered from 0 to 9, where layer 0 is the lower level:

```
Level 9:  AVOID        avoid discussion or answer
Level 8:  MINING       web search information
Level 7:  MEMORY       information retrieval
Level 6:  PROFILING    learn about users
Level 5:  ROLE         goals and "job"
Level 4:  BACKSTORY    what shape the creature
Level 3:  IDENTITY     who is the creature
Level 2:  GENERIC      generic interactions
Level 1:  EMOTION      emotional response
Level 0:  TRICK        syntax, tricks, bad lang.
```

As in a classical subsumption architecture, each layer of a given level can use the lower levels. In contrast, a lower level never use a feature of a higher level. Layers 0 and 1 are strictly reactives while layers 2 to 5 use some memory parameters and variables. Levels 6 to 8 are cognitive levels that take advantages of user profiling and web mining features based on learning and genetic programming [10]. Note that these three layers where not used in the exhibition prototype (in grey in the figure). Layer 9 is a special “idle” level that apply if none of the other levels has been able to make an appropriate answer to a user’s request.

The EVA’s core primitives include a set of functions for dynamically managing this subsumption architecture: `mask-level` and `unmask-level` allow to temporarily disable a given layer, `swap-levels` allows to exchange two layers, `save-subsumption` and `restore-subsumption` allow to save and restore a given state.

2.5 Behavior Rules

Each layer is coded as a set of behavior rules. A behavior rule is basically a production rule composed of a condition and an action parts. Both are Scheme-based function [11]. If the condition is satisfied, that is return the boolean value true, then the action is executed. The next code example shows a rule of layer 2 which handles the way to answer to all kinds of "bye" expressions:

```
(define-rule "R2" 2
  ;; condition part
  '(or (find-sentence *categories* "BYE")
       (find-sentence *user-input* "see you"))
  ;; action part
  '(begin
         (show HAPPY 0.5 10)
         (random-template "GENERIC" "BYE")) )
```

The use of the Scheme language along with the EVA core primitives allow to design rich and efficient behavior rules. In addition, it will enable in a future study to implement genetic programming for learning new behavior rules more easily.

3 The 3D Animated Character

3.1 3D Model Design

The design of the 3D animated character was an important part of the project. The pre-production phase of the exhibition leads to a theme we can summarize by an "Oracle in the Matrix". This theme is inspired by the cyberpunk culture.

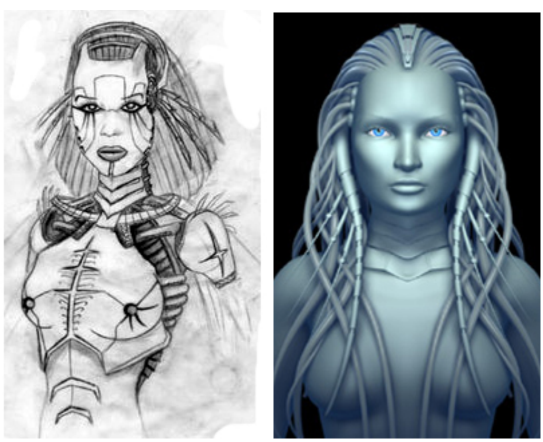

Fig. 3. The left image shows one preliminary design that has been drawn during the pre-production phase of the project. The right image shows the resulting 3D model.

One of the main goal for the 3D interface was to create a high-resolution fully textured character. However, the real-time animation of such a model requires a very powerful high-end computer and dedicated 3D rendering algorithms. We wanted to study an alternative approach for displaying realistic-looking characters. The idea was to model the character as a set of pre-calculated images that are animated by a very straightforward animation engine: basically it displays a sequence of images at a fixed frame rate. The advantage of this approach is that it enables to render high-definition images without the constraints of "low-polygons" models required by most real-time 3D engines.

Fig. 4. One of the set of images used for animating the right eye of the creature. This set is used in different facial expressions such as the "Tired" an "Sleeping" expressions or for randomized blinking eyes.

Our approach divides the complete image in a set of parts. The most important part, the animated face, is itself composed of three main parts: the left eye, the right eye, and the nose/mouth area. Then, we have reduced the number of animations to a small set of expressions based on our previous study on the emotional engine [1]. Next figure shows the main facial expressions. They are linked to the emotional engine based on the PAD model (cf. fig.2). They could be also temporarily activated using the `show` function within behavior rules (cf. code example in section 2.5).

3.2 Holographic Projection

In order to immerge the visitor in a cyberpunk ambiance before entering the room where the "Oracle" was installed, we decided to add an introductory performance piece. The latter was based on a holographic effect displaying EVA at human scale with a continuous sound ambiance. Fig. 6a shows a photograph of the resulting "holographic ghost". This principle has been originally designed by the french artist Cyril Vachez. This system recalls a number of simple optical devices that were invented in the nineteenth century like the ZooTrope created by William Horner in 1833. Fig. 6b is a photograph taken at high speed to be able to see the principle of the system. It is mainly composed of two parts. The first part includes a set of "holographic cords", an electric engine with a speed variator, and a wheel fixed on the ceiling. The second part includes a PC computer running the artificial creature program, and a multimedia video projector.

Fig. 5. From left to right and top to bottom, the main facial expressions of the creature: “happy”, “interested”, “indifferent”, “carefree”, “disappointed”, “arrogant”, “angry”, “annoyed”, “sleeping” (also “tired”), “Neutral”, “Surprised” (also “frightened”). The last image is an example of “lipsync” for speech synthesis.

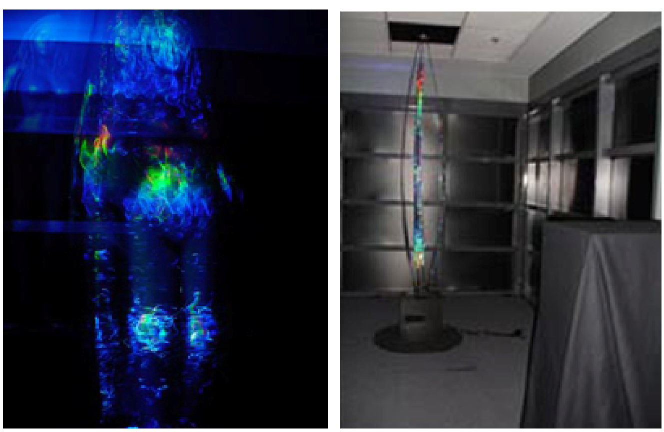

Fig. 6. The left image (a) shows the holographic artificial creature based on the EVA design. On the right, a photograph (b) of the installation showing the system composed of turning cords with holographic paper and the multimedia projector.

3.3 Projection on a 3D Prototype

The principle of the main performance piece is based on projecting the 3D animated character onto a physical model, that is a 3D sculpture representing the top of the body and the face. This creates an amazing perception of relief and makes the virtual character to appear physically real. This kind of approach has been implemented in the past by researchers such as [12] for storytelling and interactive entertainment. The physical model is designed using rapid prototyping technologies. The original 3D EVA model is simplified to enable most facial animations while minimizing image distortion (see fig. 7). The resulting model is then virtually tested to verify the result when projecting the animated character onto its surface. Then, the 3D polygonal model is converted to a standard format suitable for fabrication. The last phase is essentially manual: finishing the physical model using an epoxy resin, polishing and painting. It is interesting to note that the resulting model may recall some sort of Paleolithic Venus figurines or ancient Egyptian sculptures.

The final 3D character model is integrated with an animated background in a Java applet. This applet receives information from the behavioral engine through the networking server interface and animate the character in real-time (see figure 8).

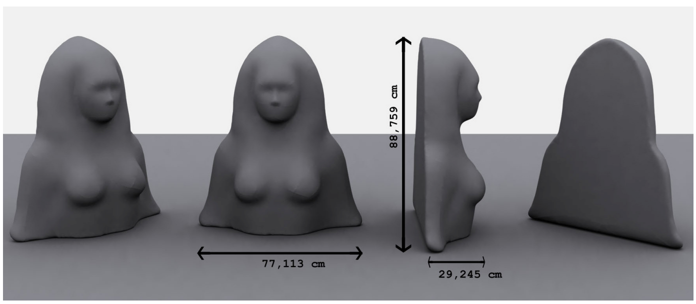

Fig. 7. The model designed for projecting the animated character on a physical 3D sculpture

Fig. 8. The final high-definition rendering of the 3D graphical interface. This image shows the neutral facial expression.

The equipment required for the prototype included three main components: the 3D sculpture, a multimedia projector and a portable computer. The exhibition room was organized in order to display EVA as a human sized character in front of the audience (see fig. 9). The projector was mounted on the ceiling to enable visitors to be just in front of the character and chat with her. One important set-up was to find the ideal position for the physical 3D model while calibrating the distance and geometry of the projected image. This is necessary in order to minimize image distortion due to the non planar nature of the model without explicitly calculating all deformation parameters which is a very difficult and time-consuming task.

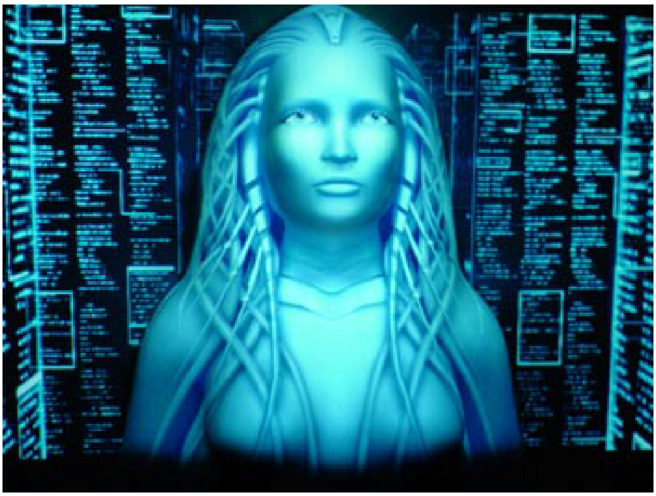

Fig. 9. A photograph taken during the exhibition showing the animated character image projected on the 3D sculpture

4 Results and Discussion

The EVA performance piece was in live demonstration during two weeks at the exhibition. Hundreds of visitors have experienced a dialog with the “Oracle”. We have recorded all the chat interactions between visitors and the virtual creature. The following is a typical example of dialog between EVA and a user with several people in the audience looking at the creature:

User: hi – **Eva:** Hello.
User: how are you today? – **Eva:** As a matter of fact, I was just going to check myself.
User: what is your name? – **Eva:** You may call me EVA.
User: how old are you? – **Eva:** Just an advice: never ask such a question to a woman.
User: zhqt – **Eva:** Is that a trick sentence?!?!
User: dpjhvd – **Eva**: Maybee my program is corrupted...
User: 01010101 – **Eva:** Is this some sort of test to separate the humans from the machines?

User: 2+2 – **Eva:** Alright. From now on I won't type anything until you make sense!
User: kiss me? – **Eva:** Can you get a little closer to the screen?
User: are you stupid? – **Eva:** Please stop immediately!
User: sorry eva – **Eva:** No problem.
User: you are beautiful – **Eva:** A true friend is someone who says nice things behind your back.
User: smile – **Eva:** I got a little cooked, but I'm okay.
User: I don't think so – **Eva:** Come on, you're better than that.
User: are you more intelligent than me? – **Eva:** How much are you willing to pay for the answer?
User: ok – **Eva:** I am very interested to know a little bit more.
User: what is your origin? – **Eva:** To make short a long story, Life is a self-reproducing organization of information.
User: this seems complex – **Eva:** It sounds interesting...
User: what do you think of life? – **Eva:** Artificial Intelligence is better than none.
User: do you believe in fate? – **Eva:** I got to say I love seeing you non-believers.
User: are you some sort of goddess? – **Eva:** I'm not a dumb program like Word. I am an AI program.
User: you are funny – **Eva:** I have NOT lost my mind. I have it backed up somewhere.
User: good job! – **Eva:** Do you really think that?
User: bye – **Eva:** Goodbye.

The knowledge base used for this prototype is composed of 80 behaviors rules for all the 7 active layers of the subsumption architecture. As stated, levels 6, 7 and 8 were not used in this prototype. Despite the relative low number of rules, most of the people visiting the exhibition were very enthusiasts after their interactive experience with EVA. We must note that an important part of interactions was not on the subject of asking questions to the "Oracle", but rather trying to trick the virtual creature by using non-sense expressions, repeated sentences, insults or sex-oriented phrases. We think it is a typical reaction of a user experiencing a dialog with an artificial human. This user's behavior can be explained by the need to reassure himself about his superiority compared to the artifact. Since a human understand people or objects through interactions, this kind of dialog can be seen as a necessary "prolog" before any "real" interaction with a virtual human. The user evaluates the artificial creature and therefore decide if it is interesting or not to continue. In some other cases, this preliminary phase is limited to a generic discussion about name, age, origin, etc. If the virtual creature passes this "limited Turing test" with success, then the dynamics of interaction change and the dialog becomes more interesting and useful. Rather than being opposed, they stimulate and affect each other. This is the second phase we call "coupling", which can therefore focus on the discussion goals. This confirms the importance of the reactive layers that are responsible for avoiding tricks and other kinds of bad languages and those dedicated to generic discussion. We consider that, in any cases, the virtual creature must not appear as more intelligent than the user and must answer tricks or other bad languages with a sense of humor rather than aggressively. We have also learned that, when coupling is done, an important part of the virtual creature's intelligence remains in the observer's interpretation of the answers. This can be achieved by using "open" and subtly correlated templates rather than strict and precise ones. This emphasizes the importance of the character design phase, which must be done with the contribution of a specialist in dialogues and scripts.

Concerning our graphical approach, its clear advantages were the high-definition of the images and the simplicity of the animation engine. However, we experienced two drawbacks. First, even if the number of emotional states is relativelly small, the number of possible transitions is high and result in a very high numbers of computed images. We tried to solve this problem by making no global moves of the face and returning to the neutral expression after each emotional expression. The second problem is that a high number of images can lead to some memory problems when pre-loading these images. The conclusion is that even if we used a straightforward animation engine compared to real-time 3D, it finally required an optimized graphical workstation in order to obtain a fluent and convincing animated character.

5 Conclusion

In this paper, we have presented a prototype of a virtual creature based on the EVA architecture. The prototype has be used as a performance piece at a multimedia exhibition with success. This experience has allowed us to validate the architecture and learn about the behavior of user interacting with an artificial human. The EVA architecture has a lot of potential applications, most obviously as an intelligent agent for answering questions and marketing studies on commercial web sites, and as an interface to search engines on mobile phones. We also like to imagine virtual assistants for lone aged and/or sick people, for learning a foreign language, as avatars in video games, etc. Future works include the development of the memory, profiling and mining layers, and the design of realistic 3D characters.

Acknowledgments. N. Aubrun, M. Bellan, J. Cordier, F. Courbier, C. Delsart, D. Kerhoas, A. Menard, H. Meyer, P.Y. Moulard, M. Nejdi, F. Oulné, K. Ludwig, Y. Ngnodjom, A. Palin St. Agathe, G. Serror, L. Simonini, C. Vachez, A. Vu.

References

1. Heudin, J.-C.: Evolutionary Virtual Agent. In: Proceedings of ACM/WIC/IEEE Intelligent Agent Technology Conference, pp. 93–98 (2004)
2. Badler, N., Allbeck, J., Byun, M.: Representing and Parameterizing Agent Behaviors. In: Proceedings of Imagina, pp. 151–164 (2002)
3. Thomas, F., Johnston, O.: The Illusion of Life: Disney Animation. Hyperion Books, New York (1981)
4. Hayes-Roth, B., Doyle, P.: Animate Characters. Autonomous Agents and Multi-Agent Systems 1, 195–230 (1998)
5. Millet, P., Heudin, J.-C.: Web Mining in the EVA Intelligent Agent Architecture. (submitted to) ACM/WIC/IEEE Intelligent Agent Technology Conference (2007)

6. Abelson, H., Sussman, G.J.: Structure and Interpretation of Computer Programs. MIT Press, Cambridge MA (1985)
7. Brooks, R.A.: A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation RA-2, 14–23 (1986)
8. Mehrabian, A.: Pleasure-Arousal-Dominance: A general framework for describing and measuring individual difference in temperament. Current Psychology: Developmental, Learning, Personality, Social 14, 261–292 (1996)
9. Wallace, R.S.: AIML Pattern Matching Simplified (2001), `http://www.alicebot.org/documentation`
10. Koza, J.R.: Genetic Programming. MIT Press, Cambridge MA (1992)
11. Burg, B., Foulloy, L., Heudin, J.-C., Zavidovique, B.: Behavior Rule Systems for Distributed Process Control. In: Proceedings of IEEE CAIA, pp. 198–203 (1985)
12. Morishima, S., Yotsukura, T., Binsted, K., Nielsen, F., Pinhanez, C.: HyperMask: Talking Head Projected onto Real Objects. The Visual Computer 18(2), 111–120 (2002)