PsychNology Journal, 2004
Volume 2, Number 1, 43 - 60

SAMIR: A Smart 3D Assistant on the Web

F. Abbattista♠, G. Catucci, G. Semeraro and F. Zambetta

Dipartimento di Informatica – Università di Bari

ABSTRACT

A current trend in modern HCI is represented by Embodied Conversational Agents (ECAs), even designed to run on the Web. They are virtual 3D human-like front ends coupled with software agents that are able to engage in a conversation with a user and execute complex tasks, such as, for example, searching for some specific information or ordering some items from the catalogue of an online shop.
This paper presents SAMIR system, a framework to build intelligent agents for the Web. SAMIR consists of a 3D face which is animated to exploit expressions which are perceived by the user; a custom version of the ALICE chatterbot to chat with the user; and finally an XCS classifier system to deal with the problem of keeping conversation and face expressions coherent with each other. Experimental results, taken from an online bookstore-based scenario, are presented at the end.

Keywords: *Conversational interfaces, intelligent agents, facial animation.*

Received 16 January 2004; received in revised form 7 April 2004; accepted 12 April 2004.

1. Introduction

People's life is rapidly shifting towards "new virtual dimensions", as opposed to the traditional physical one, where users can interact and communicate through the use of e-mail, Web, and chat technologies.

Different kinds of social relationships may be established in these virtual worlds and different interaction metaphors might be useful in order to enhance the sense of presence, i.e. the perception of "being there".

Immersive scenarios, such as the ones depicted in science fiction books like "Snow crash" (Stephenson, 2000) and "Neuromancer" (Gibson, 1995), introduced the idea that the World Wide Web becomes a totally interconnected 3D virtual environment, where people can retrieve information in a natural fashion, simply by navigating

♠ Corresponding Author:
Fabio Abbattista,
Dipartimento di Informatica, Università di Bari.
E-mail: fabio@di.uniba.it

through its “paths”, and people are embodied by 3D avatars, 3D animated representations of the user.

Based on these fascinating ideas, many research efforts are currently involved in developing Networked Virtual Environments (NVEs) (Singhal and Zyda, 1999), where people embodied by 3D avatars may interact with other people or autonomous computer-guided 3D characters.

This sort of autonomous characters, usually referred to as *Embodied Conversational Agents* (ECAs), are software components designed to act as virtual advisors within applications where a high level of human computer interaction is required.

Their aim is to overcome the classical limitations of the WYSIWYG interfaces, which cause inexpert users to suffer from an uncomfortable interaction experience.

Reactive and possibly pro-active embodied virtual personalities are able to understand users wishes, converse with them, find information and execute non-trivial tasks, replacing buttons pressing, menu choices and hyperlinks clicks, which often contribute to undesired information overload.

The metaphor of a face-to-face conversation greatly increases the feeling of presence during the interaction and eliminates the need of learning where to find specific widgets accomplishing a single task that is really needed.

Moreover, these systems are coupled with an animated 2D/3D look-and-feel, embodying their intelligence via a face or an entire body. This way it is possible to enhance user trust and satisfaction, giving users some sort of illusion of life, as cartoons, videogames or animated movies are able to do.

A very complete example of an embodied conversational agent is the REA system, developed in the Gesture and Narrative Language Group at the MIT Media Lab (Cassell, Sullivan, Prevost and Churchill, 2000). REA is an embodied conversational interface agent implementing conversational protocols in order to make interactions as natural as face-to-face conversations with another person.

REA acts as a real estate salesperson, answering user questions and showing users some virtual houses; it has an articulated graphical body and can sense the user passively through cameras and audio input; moreover, speech with intonation, facial display and gestures add a lot of depth to the ongoing interaction.

REA is rendered on a projection screen and the user stands in front of it: Two cameras mounted on top of the screen track user positions in space.

Users wear a microphone for capturing speech input, whilst an SGI Octane computer runs the graphics and conversation engine of REA, and several other computers manage the speech recognition and generation.

REA is able to conduct a conversation describing the features of the task domain while also responding to the user verbal and non-verbal input.

REA's responses are generated by an incremental natural language generation engine developed by Stone and Doran (1997), extended to synthesize gestures synchronized with speech output (Cassell, Sullivan, Prevost and Churchill, 2000).

Even though REA is a very complete ECA, it relies on a bunch of custom hardware devices for sensing user actions and requires a lot of computational resources (as many computers are used to run it).

Thus, important as it is as a proof of concept, REA cannot absolutely be defined as a portable system.

Egges, Kshirsagar and Thalmann (2003) describe a prototype system that uses the OCEAN model of personality (Costa and McCrae, 1992) and the OCC model of emotions (Ortony, Clore and Collins,1988) in combination with a dialogue system and a talking head with synchronised speech and facial expressions.

An ECA named Julie is described in a simple conversational context: Julie has to carry some boxes upstairs, so that the user might offer to help her in this tedious and tiring task.

Depending on the way Julie's personality is modelled (i.e., she is introvert or extrovert), her reactions vary from warm and kind thanks to shy and reluctant expressions and answers.

The centre of the application is a dialogue system, using Finite State Machines (FSM), which generates different responses based on the personality, mood and emotional state.

A visual front-end uses the dialogue output to generate speech and facial expressions, relying on MPEG-4 Facial Animation Parameters (MPEG-4 Committee, 2000).

The dialogue system annotates its outputs with emotional information, so that answers are coupled with expressions.

An example of such a response is (the tag |JO58| represents a joy emotion to a degree of 58%):

|JO58|Thanks! I like you too!

Even though the example application is very simple and it is based upon FSM, nevertheless it is remarkable for including both a personality model and a mood one.

The author of the Interface Project (Pandzic, 2001) implemented a full 3D MPEG-4 compatible facial animation system based on the Shout3D technology. An applet plays MPEG-4 FBA bitstreams, which are streamed from the server in real time. The audio track is streamed, as well.

Currently, fairly simple facial models are used, but the possibility of creating models and animation rules in a popular commercial 3D software package, like 3DS Max, gives the possibility of potentially using more complex models, due to the .S3D format exporter bundled with Shout3D.

The applet is controlled from Web pages by Javascript, making all interactions possible, and the virtual character can also bring up Web content in other frames of the Web page, which is synchronized with its talk.

These features enable a smooth integration in a Web site, whilst limitations include the size of the applet, which is something like 200K, as well as an unstable audio delivery.

Finally, no reasoning or learning schema is supplied with the system, so that it cannot be considered autonomous or adaptive under any point of view.

The MS Office assistants deserve to be mentioned in the agent panorama because of their widespread use, even though they are quite shallow in some respects. Indeed, they are too invasive and do not exhibit a very complex behavior. Nonetheless, they suffice to help the inexperienced user in many situations. They are based upon the Microsoft Agent technology (Microsoft Agent website http://www.microsoft.com/msagent/default.asp), a set of programmable software services that supports the presentation of interactive animated characters within the Microsoft Windows interface. Developers can use characters as interactive assistants to introduce, guide, entertain or enhance their Web pages or applications, in addition to the conventional use of WYSIWYG controls.

In addition to mouse and keyboard input, Microsoft Agent includes optional support for speech recognition, which allows applications to respond to voice commands using synthesized speech, recorded audio or text in a cartoon word balloon.

Another example of a Microsoft Agent is Instant Messaging Personalities (IMP), which gives a very good idea of how an agent can help users in handling all Microsoft Messenger features, such as mail checking, instant messaging and so on.

Agents have lips-synced faces pronouncing the messages the user receives and they tell the user whether any of his/her contacts has just gone online, offline, away, etc.

The system JackMOO (Cassel, Sullivan, Prevost and Churchill, 2000), developed at the Center for Human Modeling and Simulation, features *Smart Avatars* that understand actions they are asked to perform through the representation of actions, objects, and agents. This capability makes them a suitable solution for the execution of animations, as well as for a natural language expression.

A Parameterized Action Representation (PAR) is defined in order to drive a simulation in a given context of objects, agents and environment, and to support the considerable range of expression, nuance, purpose and manner offered by natural language.

JackMOO combines Transom Jack software with an underlying multi-user, network-accessible system that has been used for the construction of text-based conferencing, education, training and other collaborative software.

This system is a good model for virtual environment training applications, where several individuals can share a 3D space and learn team tasks and job coordination.

In this paper, we present the SAMIR (Scenographic Agents Mimic Intelligent Reasoning) system, a digital assistant where an artificial intelligence based Web agent is integrated with a purely 3D humanoid, robotic or cartoon-like layout. The outline of the paper is as follows: Section 2 presents the architecture of the SAMIR system, while sections 3, 4 and 5 detail the main modules of the system. An overview of the experimental results is given in section 6, whilst section 7 describes some conclusions and future work.

2. The Architecture of SAMIR

Figure 1 shows the client/server architecture of SAMIR. The overall behavior of the system is given by the interaction of three main subsystems disposed on one of the two sides of the application (Client Side and Server Side): The Behavior Manager (Abbattista, Paradiso, Semeraro, and Zambetta, 2004), the Dialogue Management System (DMS) and the Animation Module.

The communication between the client side and the server side is obtained through the common http protocol. The interaction with the system is delegated to the DMS client which captures user input and directs it to the DMS server and to the Behavior Manager. The DMS server selects the correct answer to each user request and forwards it to the DMS Client. The Behavior Manager covers another important aspect of this interaction: It creates the appropriate set of parameters encoding the emotional expressions of the system. The Animation Module uses these values to represent the corresponding facial expression, constructed by means of the various morph targets (Fleming and Dobbs, 1998) allowed by the system. In particular, some high-level morph targets are used in compliance with the six basic emotions as they were described by Ekman (Ekman, 1982), but even low-level ones are a feasible choice in order to preserve a full MPEG-4 compliance. The final step of this process performs a time varying key-frame interpolation, in order to animate the current facial expression.

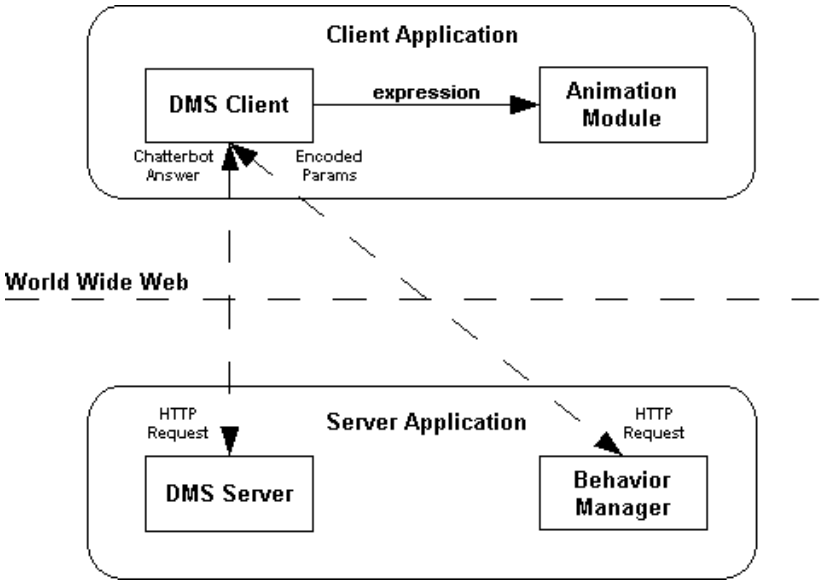

Fig. 1: The Architecture of SAMIR.

3. The Animation Module

The FACE (Facial Animation Compact Engine) Animation System is based on the work done to design and implement the Fanky Animation System (Paradiso, Zambetta and Abbattista, 2001). The Fanky system, in turn, solved some limitations of the Tinky system (Paradiso, Nack, Fries and Schuhmacher, 1999), such as the rendering limitations, due to the use of the VRML2.0 standard, and the need of installing a plugin to render 3D faces.

Our experience in Fanky led us to adopt the Shout3D API when implementing FACE: Good performances and a lot of new rendering capabilities were available since then, both in a hardware accelerated modality using the OpenGL API, and in a more limited fashion using a pure-Java software mode.

Even though Shout3D is almost compatible with VRML2.0, thanks to its extended capabilities we were able to create our 3D faces in the famous FaceGen software (http://www.facegen.com), to import them in 3D Studio Max and finally to export a single file in the Shout3D file format, containing both geometric and animation data.

However, FACE is more than a simple animation player because it relies on the idea of SACs (Standard Anatomic Components): Face regions in this model act as objects, in an object-oriented sense of the term, so that we can use different services for each of them.

Services range from reshaping, which exploit some Free Form Deformation algorithms (Sederburg, 1986), to low-level deformations such as FAPs (Facial Animation Parameters).

Moreover, SACs enable us to adopt different numerical methods to perform face region animations: Free Form Deformation, Waters muscle model and linear or non-linear key-frame interpolation schemes.

After having tried out all those methods, we decided to use by default a linear interpolation scheme based on morph targets, which is very well supported in the Shout3D API by means of the *ChannelDeformer* node because it is the best trade-off between rendering accuracy and speed.

FACE supports a variable number of morph targets in order to further improve performances: Either 12 high-level targets are used or the entire “low-level” FAP set, to achieve a full MPEG-4 compliance.

A small set of high-level parameters is very helpful when having to debug and improve the behavioral module because, of course, reasoning in terms of face expressions is better than having to deal with long streams of numerical low-level parameters, as for MPEG-4 animations.

However, many applications (e.g, teleconferencing, computer supported cooperative work, etc.) require MPEG-4 compliance and this led us to provide the possibility in case FACE, or the entire SAMIR system, has to be used by one of them.

FACE can use an unlimited number of timelines for rendering simultaneously both “synchronous” expressions, which result from conscious actions taken by the conversational agent, and “asynchronous” ones, due to the non-conscious reflexes

commonly visible in every human face (i.e., eye-lid reflexes, small head movements, etc.). Figure 2 depicts some expressions taken from an animation stream.

This way an higher degree of realism is achieved, since the user feels to find the 3D agent more familiar and more similar to a human counterpart.

Finally, we are currently developing an in-house tool to perform not only the tasks accomplished by FaceGen, but to give the possibility of obtaining new 3D faces by applying some kind of free form deformation techniques.

The former modality is usable even by inexperienced users, whilst the latter is dedicated to users with a good experience in 3D modeling and 3D digital content creation.

Fig. 2: Some expressions assumed by a 3D face.

4. The Dialogue Management System

Two software modules, the “DMS Client” and the “DMS Server”, are the main components of the Dialogue Management System (DMS), a client/server application capable of managing user inputs and properly process them in order to obtain an adequate response to user requests and needs. The DMS Client adopts the Java applet technology to run in a WWW browser, and to capture user inputs to be sent to the DMS Server via the common http protocol. This module can also direct precise queries to the Web site of interest, relying on the information processed by the server counterpart. Such a goal is achieved through the use of Javascript technology embedded in the most popular WWW browsers available today.

The DMS Server analyzes the current input text typed by the user and forwarded by the DMS client, and looks for certain classes of words that are particularly relevant in the domain addressed by the system. The result of the task performed by the server side of the DMS consists of a textual response containing the result of the search issued by the user, or an invitation to be more precise in formulating requests. This task is accomplished through the use of the ALICE chatterbot, an open source software released by the ALICE AI Foundation, and the related Artificial Intelligent Markup

Language (AIML), an XML compliant language which provides a set of custom tags able to represent the vast set of possible user inputs, and to reduce them to a minimal collection of properly selected categories. All the knowledge of the system is entirely enclosed in the AIML dialogues, which are XML files storing the various categories cited above. An AIML category is an elementary unit structured in two main parts: A "pattern" section matching user input, and a corresponding "template" section containing an adequate response to user requests and/or actions (i.e., a JavaScript execution).

The communication with the client module has been obtained by means of the integration of the ALICE classical server engine in the SAMIR system as a Java servlet, which provides the necessary software libraries to handle the communication with the client.

Making a system able to assist users during navigation in a bookshop Web site required the definition of a set of AIML categories specifically tailored for book searching and shopping. The different categories implemented in the DMS system were chosen as close as possible to the way people request books in a real world bookstore. A set of seven fields was considered to let users specify the books he/she is interested in. They include the *book title, author, publisher, publication date, subject, ISBN code* and a more general field *keywords*.

Successful examples of book requests issued to the Amazon bookshop Web site follow:

- *I want a book written by Sepulveda*
- *I am looking for a book entitled Journey and whose author is Celine*
- *I am searching for all books written by Henry Miller and published after 1970*
- *I am interested in a book about horror*

or, in alternative forms, it is possible to send requests like these:

- *Could you find any book written by Fernando Pessoa?*
- *Search all books whose author is Charles Bukowski*
- *Give me the book whose ISBN code is 0-13-273350-1*
- *Look for some book whose subject is fantasy*

Not all user requests can be properly processed by the different AIML categories and, even more, it becomes very critical for the system to deal with those requests that exhibit a high level of ambiguity, due to the intrinsic fuzziness of the human language.

5. The Behavior Manager

The environment in which SAMIR acts is represented by the user dialogue (the higher the user satisfaction, the higher the reward received by SAMIR). Dialogues and facial expressions represent the communicative acts of our agent. The consistency between the facial expression of the character and the conversation tone is one of the main requirements of SAMIR.

At the very beginning of its *life*, the behavior of SAMIR is controlled by a set of random generated rules and, consequently, it is very stereotyped. By interacting with users, SAMIR is able to learn better behavioral rules in order to achieve even better performance.

To meet this goal, the learning module of SAMIR, namely the Behavior Manager, has been implemented through an XCS (Wilson, 1995), a machine learning system closely related to the Q-learning, but able to generate task representations which can be more compact than tabular Q-learning (Watkins, 1989). At discrete time intervals, the agent observes a state of the environment, takes an action, observes a new state and finally receives an immediate reward.

The learning process of an XCS includes three phases. In the acting phase, the performance component of the XCS selects the action to be performed. As a consequence, the system receives a feedback from the environment, which determines the effectiveness of the performed action. In this phase, the reinforcement component of the XCS evaluates the received reward and assigns a fitness value to the action chosen in the first phase. This process is iterated until the system is able to select a very fitted rule. In case of degrading performance (i.e., the reward from the environment is a negative value), the discovery component of the XCS starts an evolutionary process to generate new behavior rules. At the end of the evolution, the newly discovered rules substitute the less fitted rules of the XCS and the process is re-executed from phase 1. In SAMIR, behavioral rules are expressed in the classical format if <*condition*> then <*action*>. The <*condition*>, representing the state of the environment, is a combination of 4 possible events, sensed by 4 effectors, corresponding to different conversation tones such as: User salutation (user

performs/does not perform salutation), user request formulation to the agent (no request, polite, impolite), user compliments/insults to the agent (no compliment, a compliment, an insult, a foul language), user permanence in the Web page (user changes/does not change the page). The <*action*> represents the expression that the Animation System displays during user interaction. Specifically, the expression is built as a linear combination of a set of basic expressions. We decided to extend the set of fundamental expressions proposed by Paul Ekman (anger, fear, disgust, sadness, joy, and surprise) (Ekman, 1982), by including some typical human expressions such as bother, disappointment and satisfaction. Thus the <*action*> part provides the Animation System with the percentage of each expression, to be used to compose the desired final expression of our character. For example, an expression composed by 40% of joy and 60% of surprise is coded into the following binary string:

0100	**0000**	**0110**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
% of Surprise	% of Sadness	% of Joy	% of Fear	% of Disgust	% of Anger	% of Bother	% of Disappointment	% of Satisfaction

6. Experimental Results

In a preliminary phase, we defined a set of 30 interactional rules in which different situations have been considered (performed actions, user requests, etc.). This set of pre-defined rules represented the training set, which is the minimal know-how SAMIR should possess to start its *work* in the Web site.

To evaluate the performance of the system, during the training phase, the following criterion has been adopted for the credit assignment: A perfect match between the expected action and the action performed by the system is the best result, otherwise the credit assigned to the rule is inversely proportional to the distance between the expected action and the action performed by the system.

The experiment aims at verifying the effectiveness of the User Interaction module when learning the 30 predefined rules. We performed 5 runs of the system with the same parameters (see Table I) but with a different initial population. The values of the parameters have been empirically selected.

On average, the system was able to learn 26 out of 30 rules. The 4 unlearned rules were, in most cases, very close to the original predefined rules. The most interesting result has been the ability of the system to discover new rules, not included in the original set of 30 rules.

Parameter	Value
Maximum Population Size (N)	24000
Crossover Probability (Px)	1.0
Mutation Probability (Pm)	0.03
Learning Rate (β)	0.1
'Don't care' symbol Probability (P#)	0.05

Table 1: The parameters used in the first experiment.

For the sake of brevity, we omit to report the whole set of learned rules. Table II shows only some rules learnt by the system. The second and the third columns represent, respectively, the input to the system (the <*condition*> part of the rule) and the expected output (the <*action*> part of the rule). The fourth column shows the system output.

The first row in Table II represents a perfect match between the expected action and the action performed by the system. In the second row, the rule does not represent a perfect match but the two actions (the expected one and the action performed by the system) are quite *similar*. Last but not least, the third rule represents a new rule discovered by the system.

Due to the inherent features of the XCS, SAMIR was able to learn the pre-defined rules of behavior quite effectively and to generalize some new behavioral pattern that could update the initial set of rules. In such a way, SAMIR is comparable with a human assistant who, after a preliminary phase of training, will continue to learn new rules of behavior, based on personal experiences and interactions with human customers.

Rule	Input	Expected Output	System Output
1	0001001001100	0 0 0 0 60 40	0 0 0 0 60 40
2	0001111011000	0 0 60 0 0 0	0 0 40 0 0 0
3	0101100001001 (A novice user makes few errors)	-	70 0 30 0 0 0

Table 2: Results of the training phase

Having experimented the capability of the leaning module, we started a set of experiments aiming at verifying the effectiveness of SAMIR. We had to analyze two different characteristics of the system:

1. *User likeness*: Do users like/dislike to interact with a 3D animated face? Do users feel comfortable when interacting in natural language with SAMIR?

2. *SAMIR performance*: Is SAMIR able to interpret correctly books requests issued by a user? Do search results provided by SAMIR fit users requests?

Concerning the first issue, we have not yet performed a complete usability test. However, we asked some users, very acquainted with Web search engines, to record their first impressions on SAMIR. In the experiment, users were asked to interact with SAMIR for a week. In the first part of the week they were free to converse with the agent on general topics like sports, politics, etc. In the last three days, users were forced to ask SAMIR for some books on the Amazon.com website. At the end of the week, users were asked to fill a questionnaire and to report their comments and suggestions about their experience.

The results of this preliminary test show that, on average, users like to converse with an animated 3D agent (most of our users thought of SAMIR as funny, interesting, etc.). On the other hand, they do not think SAMIR is useful in searching books and they still prefer to use the classical Amazon.com search engine. This result is not surprising since our users access Web search engines (such as Google) on a daily basis and are quite expert in composing effective queries. In the next future we will perform a complete test involving more users with different skill levels, to collect a more significant sample of users reactions to SAMIR behavior.

In the rest of this section we present some examples obtained from the interaction between SAMIR and some typical users, searching for books about topics like literature, fantasy and horror or looking for books that match specific constraints concerning title, author and publisher are given. As can be seen in the following figures, SAMIR has been able to find books satisfying user requests.

First of all, in Figure 3, we take a look at the initial presentation of the SAMIR Web Agent as it starts up; SAMIR presents itself and asks the user his/her name for user authentication and recognition.

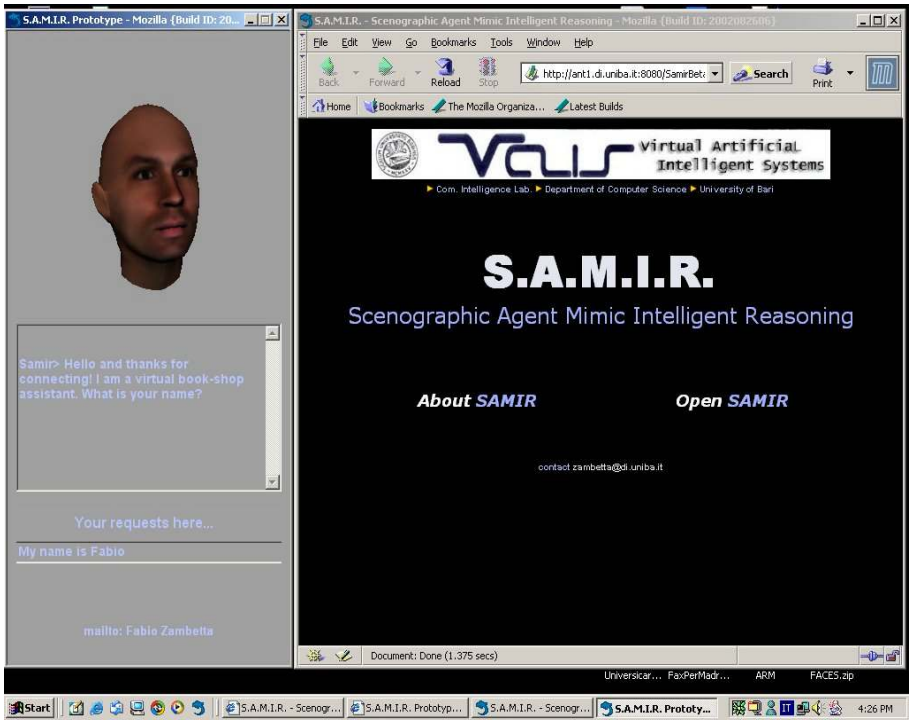

Fig. 3: **SAMIR log.**

Figure 4 shows the results of a user request about a horror book. The result of the query is a set of books in this genre, available at the Amazon book site.

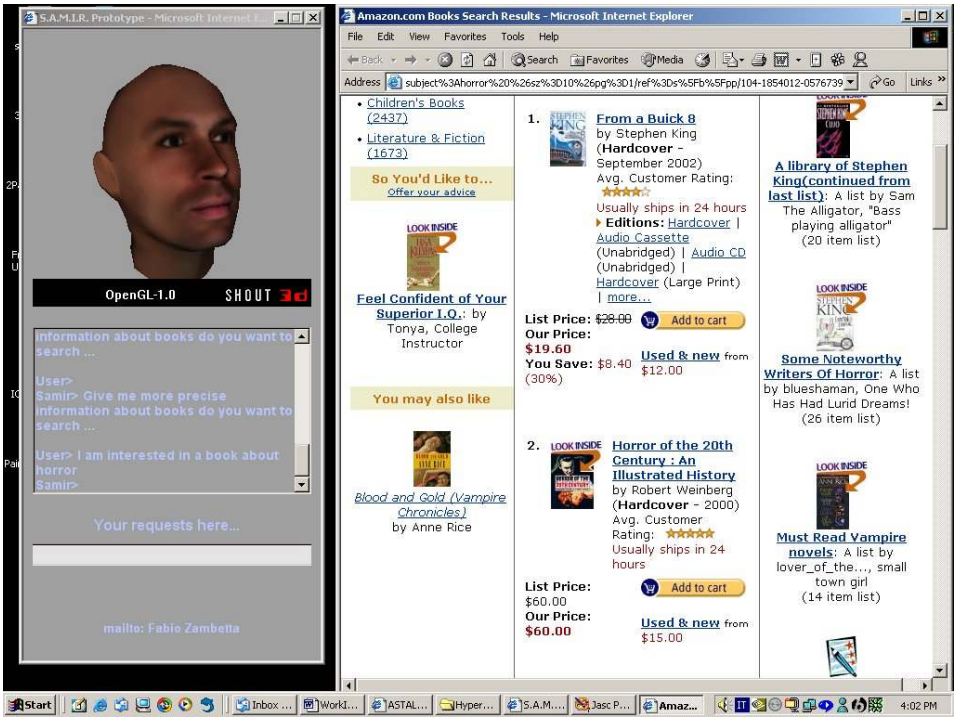

Fig. 4: Requesting an horror book.

Figure 5 shows a more specific query of a user interested in a list of books by the author Sepulveda.

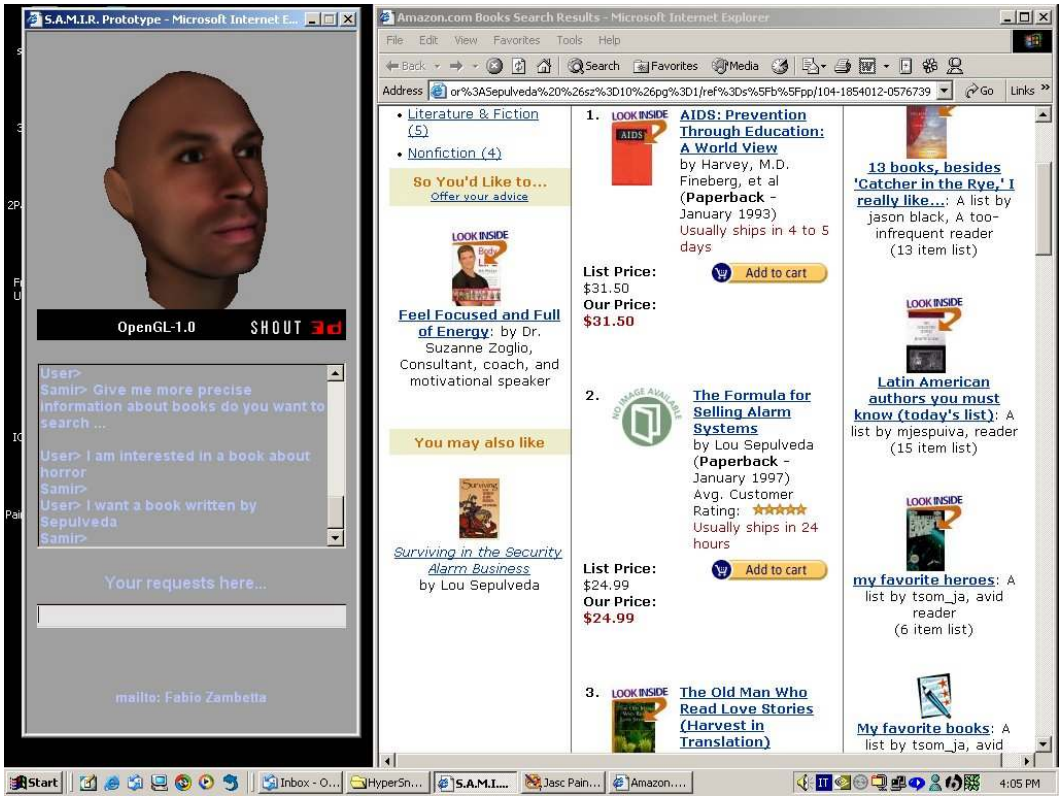

Fig. 5: Results about Sepulveda author.

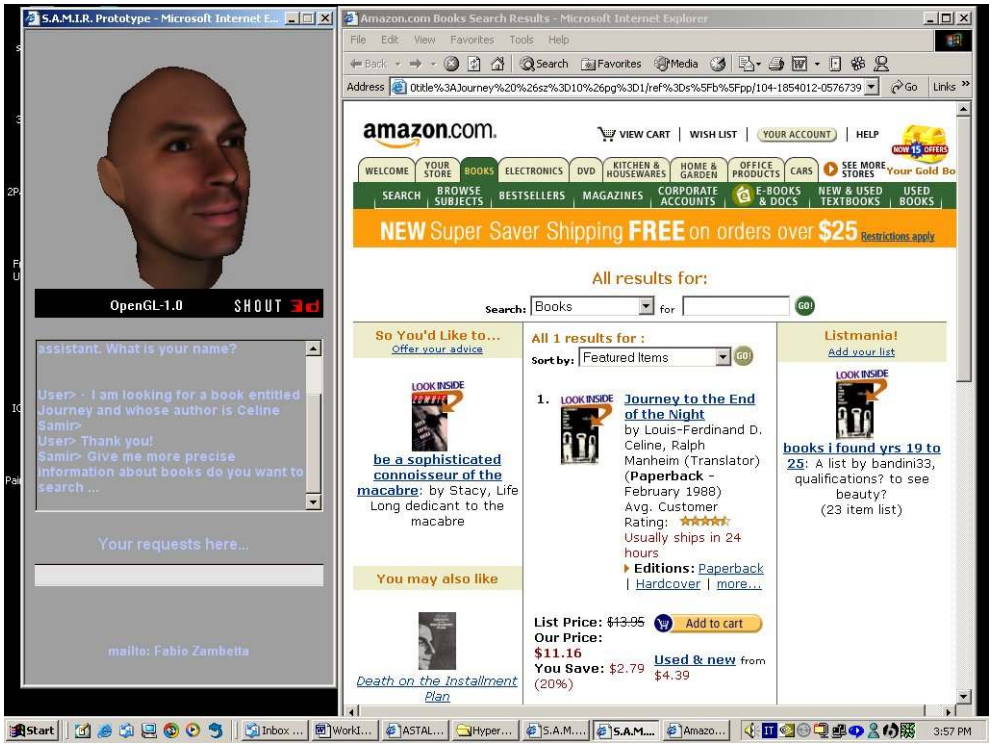

Fig. 6: Journey to the End of the Night by Celine

Figure 7 is an example of a more sophisticated query in which there is a request for Henry Miller books published after 1970. In this case the user gives a heavy insult to SAMIR and, consequently, its expression is angry.

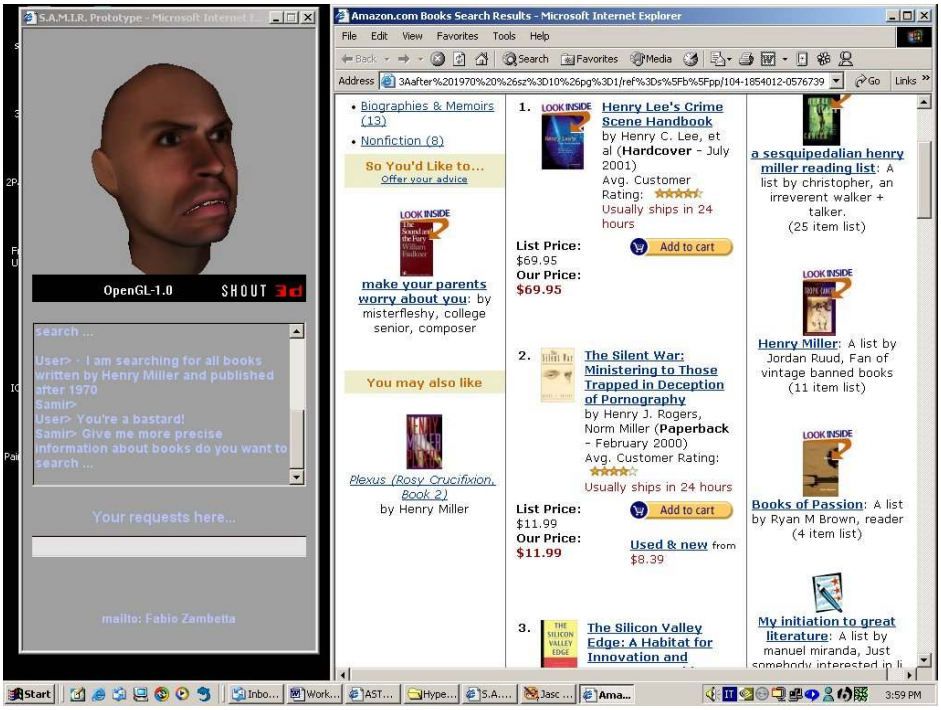

Fig. 7: All books by Henry Miller published after 1970.

7. Conclusions

In this paper we presented a first prototype of a 3D agent able to support users in searching for books in a Web site. The prototype has been linked to a specific site, but we are implementing an improved version that will be able to query several Web bookstores simultaneously and to provide users with a report that summarizes a comparison based on different criteria, such as price and delivery times.

The SAMIR system offers a wide covering of heterogeneous aspects (e.g., a 3D multi-modal interface, a learning module and a dialogue subsystem) in a very light system. SAMIR offers the possibility of being used via a common Java applet embedded in a Web browser and it will also be available as a stand-alone application which can use more rendering features. Moreover, the focus of the system is centered on creating a realistic type of interaction, by keeping the expressions of the 3D face emotionally coherent with the dialogue flow.

Our work will be further aimed at giving a more natural behavior to our agent. This can be achieved by improving dialogues, and possibly, the text processing capabilities of the ALICE chatterbot, and giving the agent a full proactive behavior: The XCS should be able not only to learn new rules to generate facial expressions but also to modify dialogue rules, to suggest interesting links and to supply an effective help during the site navigation.

8. References

Abbattista, F., Paradiso, A., Semeraro, G., and Zambetta, F. (2004). *An agent that learns to support users of a Web site*. Applied Soft Computing, 4(2004), 1-12.

Cassell, J., Sullivan, J., Prevost, S., and Churchill, E. (Eds.) (2000). *Embodied Conversational Agents*. MIT Press, Cambridge.

Costa, P.T., and McCrae, R.R. (1992) *Normal personality assessment in clinical practice: The NEO personality inventory*. Psychological Assessment, 4(1), 5–13.

Egges, A., Kshirsagar, and S., Thalmann, N. M. (2003). *A Model for Personality and Emotion Simulation*. In Palade, V., Howlett, R.J., Jain, L.C. (Eds.). Knowledge-Based Intelligent Information and Engineering Systems. Springer, Berlin, 453-461.

Ekman, P. (1982). *Emotion in the human face*. Cambridge University Press, Cambridge.

Fleming, B., and Dobbs, D. (1998). *Animating Facial Features and Expressions*. Charles River Media, Hingham.

Gibson, W. (1995). *Neuromancer*. HarperCollins, New York.

MPEG-4 Committee, (2000). *The MPEG-4 standard specification*, http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm.

Ortony, A., Clore, and G.L., Collins, A. (1988). *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge.

Pandzic, I.S. (2001). *Life on the web*. Software Focus Journal, 2(2), 52-59. John Wiley & Sons.

Paradiso, A., Nack, F., Fries G., and Schuhmacher, K. (1999). *The Design of Expressive Cartoons for the web – Tinky*. In Proc. of *ICMCS Conference*, IEEE Press, 276-281.

Paradiso, A., Zambetta, F., and Abbattista, F. (2001). *Fanky: A tool for animating 3D intelligent agents*. In de Antonio, A., Aylett, R., and Ballin, D., (Eds.). Intelligent Virtual Agents. Springer, Berlin, 242-243.

Sederburg, T.W. (1986). *Free-Form Deformation of Solid Geometric Models*. Computer Graphics, 20(4), 151-160.

Singhal, S., and Zyda, M. (1999). *Networked Virtual Environments: Design and Implementation*. Addison Wesley, Boston.

Stephenson, N. (2000). *Snow crash*. Bantam Books, New York.

Stone, M., and Doran, C. (1997). *Sentence Planning as Description Using Tree-Adjoining Grammar*, *ACL*. MIT Press, Madrid, 198—205.

Watkins, C.J.C.H. (1989). *Learning from delayed rewards*. *PhD thesis*, University of Cambridge, Psychology Department.

Wilson, S.W. (1995). *Classifier Fitness based on Accuracy*. Evolutionary Computation, 3(2), 149-175.