

A Multimodal Presentation Mark-up Language for Enhanced Affective Presentation

Sylvain Descamps
Dept. of Information and Communication Eng.,
University of Tokyo
Mail address:
Ishizuka laboratory
Dept. of Information and Communication Eng.,
School of Eng., University of Tokyo
7-3-1, Hongo
Bunkyo-ku, 113-8656 Tokyo
Japan
Tel: +81 3 5841 6755
Fax: +81 3 5841 6698
descamps@miv.t.u-tokyo.ac.jp

Helmut Prendinger
University of Tokyo
helmut@miv.t.u-tokyo.ac.jp

Mitsuru Ishizuka
University of Tokyo
ishizuka@miv.t.u-tokyo.ac.jp

Abstract

Nowadays, agent systems are flourishing. We can see them presenting news on the Internet, or guiding virtual tourists in 3D worlds. All these agents are considered as a new interface between user and computer, more friendly and natural. This involves natural voice communication and natural behaviour management. Our interest is to create a language called Multimodal Presentation Mark-up Language (MPML), to command interface agents to do presentations so that users can easily add such attractive presenters on their WebPages. We will present how we define a presentation and what the features of MPML are. Moreover, we have been interested in creating affective behaviours for the agents, which includes emotion expression and personality, and we want to make it easy to use for the author.

Keyword: MPML, presentation agent, emotion expression, Internet presentation.

1. Introduction

The main goal of our research is to create presentation with animated agents, which will be used on the Internet. Indeed, human presentations are still much better than those given by agents and so, when a traditional presentation is possible, it should be used, as for conferences or courses. However, as soon as computers are integrated between two humans, the effectiveness of the presentation will depend on the tools they are using, like of the Internet. Classic Web pages, emails or forums and slides with voice [20] are already ways to communicate, and they all have their advantages and drawbacks. However, only the Web page or the slides are a practical way of producing presentations, whereas email and forums more focus on information exchange. Besides, as soon as complex features, such as applet or database, are added, authors not accustomed with such techniques cannot add such useful tools and are limited to text, image and video.

Video can also be difficult to use, as it requires camera [23].

The purpose of MPML is to make it possible for authors to add agents on their Web pages, which will greatly improve simple images or texts, without the high requirement of videos. Our motivation is mainly based on the assumption that humans treat computer socially [19]. The interest of using agent systems is then to increase this effect and to permit more human-like interactions (facial expression, gaze, voice tone, etc.). We are not the only ones to consider this kind of systems, practical for future presentation. Indeed, Microsoft Narrator [14] allows authors to add simple agent script to a PowerPoint presentation. It can then be used through the Internet. This looks mostly like a PowerPoint presentation, and lacks traditional features of Web pages. Other tools can be found on the Internet to ease the use of Microsoft Agent, like MASH [5]. The salient features of MPML are the presentation scripting facilities, the interaction with the HTML background and emotional expression for the agents.

The rest of the paper is organized as follow. Section 2 will present the interface agent system we are using and its advantages. Section 3 will then describe how MPML controls the system. It will first explain how to structure a presentation using MPML, interacting with the Web page background and then it will present how we provide emotional control for the agents, thereby increasing the believability of the presentation.

2. Interface Agent system

2.1. What is an interface agent system?

There are several types of *interface agents*. These agents can be either human-like [8] or more cartoon-like [12], and they can have a full body or only a face. When on the screen, they can play some animations representing gesture, body language or facial expression, sometimes concerning emotion like *sadness* or *happiness*. Cartoon-like agent animation can be quite funny and not so natural, more related to cartoon techniques with exaggerated gestures and unambiguous meaning. The Visual Software Agent is on the other extreme, a human face, showing natural expression [10]. Some other research groups chose a virtual face, to present news for example [1, 22]. We are using mainly Microsoft Agent (see Screenshot 1) and thus full-body cartoon-like characters. However, we also have a human-like single-face and full-body interface agent system in progress (see Screenshot 2).

The single-face agents are more likely to be used only for speaking to the user, it may look like current news broadcasts on television. Full body agents are more like puppets, there can be more on the screen, if the organisation of the presentation is well controlled and act more like a theatre, speaking to the user but also to each other. MPML will be able to use both possibilities.



Screenshot 1: Microsoft Agent used with MPML, two agents used.



Screenshot 2: Visual Software Agent (VSA) developed in Ishizuka Lab, University of Tokyo

The second part of an interface agent system concerns the voice management, with two components: the voice generation module and the voice recognition module. Voice generation can be done either using recorded speech (audio files) or Text-To-Speech engines (TTS engines), which can read text aloud. In our case, we are using TTS engines compatible with the SAPI norm, which can be used with Microsoft Agent. One problem here is

that we only have controls for three parameters: pitch, volume and speed. When we plan to generate affective voice to express an emotion, we are restricted [7], but we can still have naturally sounding results. The other direction for the voice is from the user to the computer. However, even without dealing with natural language understanding, sentence matching can still be a problem. The user may sometimes have to repeat his/her order several times, but it can still be used. More complex systems exist, using 3D immersion, wearable, camera [8], but we are dealing with systems that can be used without special equipment.

2.2. Why and how to use an interface agent system?

We can find several categories of use of agents, especially in the case of a presentation. First, the agent can represent a virtual character. We mean that it will consider itself as an individual, and will interact with its environment from this perspective. That is the case of two systems which present news on the Internet [1, 22]. In the case of face-only agents, it really looks like video-communication and the agent can be supposed to be at its desk in front of cameras. In the case of full-body agents, that's a bit different. Indeed the agent is seen in totality, and it looks more like a puppet, living inside the computer, which discusses with the user or other agents for the presentation. This kind of agent can easily be used to create a virtual theatre [18].

The second possibility is that the agent represents someone existing, a real human person. In the case of presentation, it can often be the author, with maybe some of his/her assistants or friends [6]. In this case the agent has much more responsibilities, as any mistake or misconduct it may do can be ascribed to the represented human being. Features like emotion, personality or even behaviour can then take more importance than when the agents were only virtual. MPML will then make it possible for any author who chooses to use interface agent to do it with little effort and great new opportunities.

3. Our Multimodal Presentation Mark-up Language (MPML)

3.1. MPML description

The main goal of the MPML project is to enable easy creation of Internet presentations using agents. We will now try to characterise the agent system we

are using in more detail. For now, MPML is able to command two different agent systems. However, our main recent interest concerns the Microsoft Agent system. As it is free to download from the Internet, everyone should be able to use and watch such a presentation. The second system is a 3D agent currently developed in another laboratory. There are also at least three other agent systems currently under construction in our laboratory, a human face in a powerful environment [10], a VRML full-body agent, and another 3D human face. All these are or will be controlled via MPML.

As indicated in its name, MPML is above all a language. However, it is really easy to use. It is similar to HTML in the format (mark-up language), but with much less tags and a globally high level use. We will detail its characteristics in the following parts. There are two ways of using an MPML presentation once written right now. The first one is to use a plug-in of Internet Explorer 5 or higher based on XML technology [24]. It allows the MPML script to be used directly by the browser. This version focused on the use of button, textbox and other useful web content, directly inside the MPML script. The background is a simple Web page. The second way is a converter software. It reads the MPML script and creates complex JavaScript code corresponding to the simple MPML instructions. Once converted, the presentation can be read directly into Internet Explorer. In this case, it is possible to use any complex Web page as a background, with its own JavaScript content if needed. Then the presentation follows the MPML script initially written by the author, and can interact with the background pages. The main addition concerns the affections system. Indeed in this version the agent shows affective behaviours easily described in the script. We will detail all these characteristics in the following sections.

3.2. MPML Agent action in MPML

The available actions for the agent defined in MPML are as follow. We call *action tag* a tag used to make the agents act. Agents can do several actions: speak, think, play, and move.

Speak: The agent speaks one or several sentences and the text is displayed in a balloon. The parameters of the voice are set using the current emotional state of the agent (emotion and mood). Some actions can occur at the same time as the speech when a strong emotion arises.

```
<SPEAK agent="merlin">
```

My name is Merlin, I am glad to meet you today.

```
</SPEAK>
```

Think: Like *speak*, but the agent does not speak aloud. Instead there is only the text in a balloon. This is practical when several agents would like to "say" something in the same time. Indeed using the <THINK> tag, several balloons can appear and only one agent will be speaking, keeping the voice understandable, instead of having everyone speaking in the same time.

```
<THINK agent="rocky">
```

I hope Merlin will soon leave so that I can be alone.

```
</THINK>
```

Play: The agent can play an animation, like *greet* or *announce*. This adds meaning to the presentation. Using this the author should know what the agent he/she is using is capable of doing, else the act may not occur.

```
<PLAY agent="merlin" act="announce"/>
```

Move: The agent moves on the screen.

```
<MOVE agent="rocky" spot="spot3"/>
```

3.3. Structure of an MPML presentation

The organization of presentation in MPML is mainly accomplished by a graph-like structure. A first tag, the <SEQ> tag, defines part of the presentation where the actions of the agents happen sequentially. This will certainly be the case for most of the presentation, since if too many things happen in the same time the user would be lost. However, there is the possibility to define some parts in which actions will happen simultaneously, using the <PAR> tag. With a nice combination of both tags, it is possible to generate almost any profile for the flow of the presentation (see Figure 1).

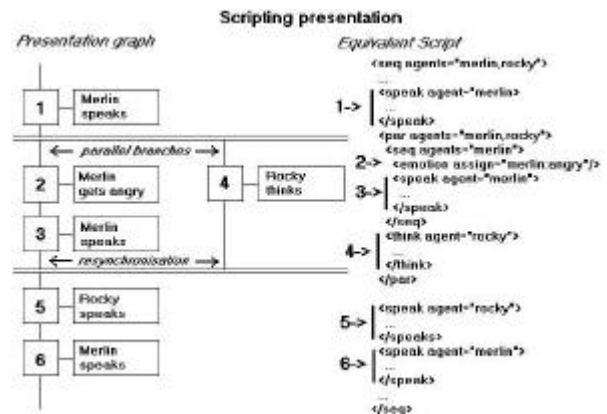


Figure 1: Use of <SEQ> and <PAR> in MPML script

The last presentation tag concerns more the intelligibility of the script. It is very convenient to separate long parts of a presentation into several *scenes*. Like in theatre performance, the agent will play the scene one after another, going in and out, as required by the scenario of the presentation. The next part gives detailed explanation and example of the use these tags.

3.4. Presentation tags in MPML

Scene: The presentation is organised like a theatre with different scenes. Each scene indicates which actors (here agents) will play in the scene, and so, appear on the screen. Scenes are also used as a reference to be used for some piece of presentation in another part of the script, like some basic procedure.

```
<SCENE id="introduction" agents="merlin"> ...  
</SCENE>
```

Seq: The tag <SEQ> is used to define a sequential presentation. Within the start tag <SEQ> and the end tag </SEQ> the order will be taken one after one another. Combined with the <PAR> tag it allows a complex scripting of the presentation. We also indicate which agents will be used in the tag.

```
<SEQ agents="merlin,rocky"> ...  
</SEQ>
```

Par: This tag is to be used with the <SEQ> tag. It defines a part of the presentation where orders will be executed all in the same time. However it may be impossible to execute the given order if two orders are given to the same agent for example, as depending on the system used, only one order may be executed at a time.

```
<PAR agents="merlin,rocky"> ...  
</PAR>
```

3.5. Interaction of MPML script with the background

In a MPML presentation, we can consider that there are two different components: the agents and the background. The agents are the actors, they speak, move and thus present. But except for the case of virtual theatre where the presentation can be self-sufficient, the background plays a very important role as a support for the presentation, as in the case of human presentation.

The common way of navigating between pages on a Web site is using *links*. We click on a link contained in an image or a text and a new page appears. Then it is possible to go back and forth at convenience. MPML enables a new navigation method. Indeed, the script can change the background page to match the explanation of the agents, using the <PAGE> tag. However, this page navigation method is efficient in the case of a single presentation, but if the author wants to organize the Web site into a big presentation database, then the best way is certainly to combine both methods, a MPML control for presentations, and classical links in-between.

Another important point concerning the background is that a large number of people already have a Web page, but only a few are using interface agent systems. It would be very unpractical for Web masters to have to rewrite every Web page in order to add agents in it. So that's why it is important to be able to adapt an already existing system to an agent-enhanced presentation with little effort. The MPML presentation can use any existing Web page as a background, and then add the agents to make it livelier. Only a few adaptations have to be made manually.



Screenshot 3: Web Page with Agent and interactive content (button, checkbox, etc.)

Finally, in order for MPML to be not only easy to use but also a powerful language, we provide a simple interface with JavaScript functions. So, if the background contains a JavaScript script, realizing some complex functions, it is possible to integrate and synchronize the presentation with some events happening on the Web page (see Screenshot 3). We give some examples of the possibilities of using an interactive background in the next part.

3.6. Interaction tags in MPML

Page: This defines the background of the presentation. The page used can be any Web page and it appears behind the agents. No action can be made by the agents if there is no background defined.

Wait: The first possible interaction between MPML script and JavaScript is the <WAIT> tag. It simply consults the value of the variable or function included in the "target" attribute until it comes to be different from "0". Then the script goes on.

```
<PAGE id="page"
ref="MPML2_5_Evaluation_Emotion_Submit.htm">
  <WAIT target="Pressed"/>
</PAGE>
```

Consult: This tag is like an equivalent of the well-known C "switch" instruction. It consults the data or function given in the "target" attribute and then executes the <TEST> tags to choose the one to execute. The <CONSULT> tag can either make one test only and then go on even if no match is found or wait until a match is found.

Test: This tag is to be used with a <CONSULT> tag as parent. It compares the value of the <CONSULT> tag target with a value and executes the following script if it matches.

```
<CONSULT target="dangerous" mode="pass">
  <TEST value="true"> ...
</TEST>
</CONSULT>
```

Execute: This tag executes a JavaScript instruction. It should be mainly used to call function in the background page of the presentation. However its used is strictly limited only by the need of the author.

```
<EXECUTE target="Alert()"/>
```

Txt: This tag is used to introduce variable content into the speech of the agents. It can permit some customisation of the dialogue, to include names or some answer of the user.

```
<SPEAK agent="rocky">
  Welcome, <TXT target="name"/>!
</SPEAK>
```

3.7. Affective behaviour: emotion, mood and personality

3.7.1. Existing emotion models

Emotion is one important challenge for agents. Indeed, in human-human communication, emotion plays a very important role, as it adds a second meaning to the dialog, and so, conveys more information. To be able to control emotion would make the conversation between the user and the agent more natural.

Several models have emerged from psychologist studies. One of them is the famous model from Ortony, Clore and Collins (OCC model). [17] The OCC model classifies 22 emotions depending on what makes them occur, and their influence on the environment of the agent. This model seems more useful when there is computation about the emergence of the emotions, like for autonomous agents with automatic emotion. Another theory is the “basic emotion” theory from Ekman [11]. It filters out 6 emotions (*anger, fear, sadness, enjoyment, disgust, and surprise*). It is argued that every emotion can be expressed as a combination of these basic emotions. It also explains that positive emotions have almost all the same expression. In this direction, there are several works concerning the voice associated with an emotion. Murray and Arnott [16] gave some results about how the voice changes according to the emotion involved. They also considered 6 emotions with only slight differences from Ekman (*anger, fear, sadness, happiness, disgust, and grief*). We used their research but we had to extend to other positive emotions like *surprise* or *gratitude*.

3.7.2. Emotion and mood in MPML

Our interest is more focused on the expression of emotions rather than emotion models. Indeed, the author decides which emotion he/she wants to use and so we need meaningful names for emotions useful in case of presentation, and then find an accurate way to express them. As we mentioned before, the main two components of agent communication are the voice and the gestures. In the case of Microsoft Agent we have very little liberty regarding the gestures and we can use only predefined behaviours. However, we have more control on the voice; we can set the speed, pitch and volume and add some emphasis on words. We then used Cahn and Murray and Arnott’s research [7, 16] to set the voice parameters according to the emotion.

We also use some expressive gestures to accentuate the emotion understanding. The emotions we chose until now are *joy, sadness, gratitude, anger, shame, surprise* and *fear*. We can still adjust the proposed emotion easily.

Besides emotion, we are also using mood. While emotion is defined as short and intense, mood is longer and with a lower intensity. It works more like a background emotional state, when no strong emotions are occurring. The main function of moods in MPML presentation is to define how the author wants the agent to behave when no emotions occur. We only defined three moods: happy mood, neutral mood, and unhappy mood. As the intensity of the expression is lower, there is little interest in using more. The mood of an agent can change during the presentation. This should not happen too often, as mood is a long term emotional state, but when an emotion occurs the author can decide that the mood of the agent goes better or worse (if Rocky has an unhappy mood but gets a gift, then the *joyful* emotion occurs and the mood can raise to neutral or even happy).

The way we combine mood and emotion is simple. When there are no emotions occurring, we simply use the parameter values defined for the mood. We also add a slight perturbation to increase the believability. Then, when an emotion occurs, we switch to the emotion parameters quickly, as emotion is intense and fast, and then we gradually come back to the mood parameters (see figure 2). It gives a natural expression for the emotion. We will explain how the personality influences this system in the next Section.

Evolution of the voice parameters with emotion

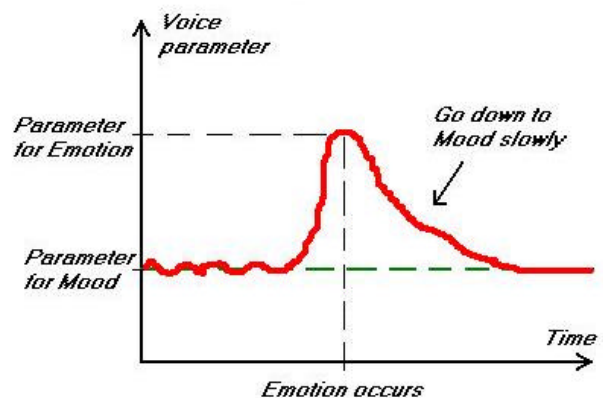


Figure 2: Voice parameter evolving with an emotion

3.7.3. Personality models and use in MPML

Personality is another very important feature of agents. Indeed, as soon as several agents appear on the screen, they may all behave the same way, speak with the same voice, and do similar gestures. Personality is one of the characteristics of humans that makes everyone act in its own way, which gives us uniqueness and makes us interesting [15]. And so an agent needs such a characteristic to act in a way different from the other agents on the computer screen.

One personality model coming from research in psychology is the Five Factors model [13]. It is purely descriptive but it gives a nice way to describe a given personality and it seems commonly used. It deals with five dimensions for the description of the personality: *extraversion*, *agreeableness*, *conscientiousness*, *neuroticism*, and *openness* (please refer to the original article for further details). These five dimensions form a very complete model, however, for practical use, only two or three dimensions are used [2, 3, 4]. In our case, the author has a lot of liberty to define the personality of the agents as he/she can choose what they say, how they speak, and how they react to events; almost everything depends on the choice of the author. Some other models exist [21] but most of them seem to include the two dimensions we are using.

From the Five Factors model, we are only using two dimensions: *agreeableness* and *activity* (from *extraversion*). *Activity* only affects the average frequency of gestures the agents makes. The author should then respect the trait of personality in the dialog content. *Agreeableness* has more impact on the presentation. Indeed, we explained how emotion and mood work together, but the agreeableness has its impact on the system. Indeed, friendly characters and unfriendly characters will not react the same way to an emotion. For example, when an agreeable agent with a happy mood is insulted by another agent then it will get angry. However, as it is an agreeable character, it will soon recover to its happy mood. On the opposite, if it gets joyful, it will stay joyful longer.

3.8. Emotion, mood and personality tags in MPML

We will describe how to use emotion, mood and personality in an MPML script. Combining these tags, it is possible to render believable behaviour for the agents (see *Demonstration 1*). In case of only one agent, it may not be necessary to use personality, but

as soon as there are two or more agents, it greatly increases their believability. The meaning of the `<EMOTION>` tag is the emergence of the emotion, and so it is punctual. However, the effect can last longer, depending on the emotion and the personality.

Result of a simple MPML script	
<code><body></code>	->Start of the presentation.
<code><seq actors="merlin,rocky"></code>	->Everyone has a neutral mood.
<code><mood assign="neutral"></code>	->The background appears.
<code><page ref="intro.html"></code>	->Only Merlin appears.
<code><scene agents="merlin"></code>	->Merlin makes a big smile.
<code><emotion assign="merlin:joyful"/></code>	
<code><speaking agent="merlin"></code>	and begins to speak with a joyful voice.
<code>Welcome to my presentation.</code>	
<code></NB/></code>	
<code>My name is Merlin, I am glad to be here.</code>	The voice gradually becomes neutral.
<code>...</code>	
<code>But let me present my partner.</code>	
<code></speak></code>	
<code></scene></code>	->The scene changes and Rocky appears!
<code><scene agents="merlin,rocky"></code>	->In the same time, two actions:
<code><par agents="merlin,rocky"></code>	- Rocky makes an angry gesture
<code><emotion assign="rocky:angry"/></code>	- Merlin speaks
<code><speaking agent="merlin"></code>	
<code>You are just on time!</code>	
<code></speak></code>	->End of the simultaneous part
<code></par></code>	
<code><speaking agent="rocky"></code>	->Rocky speaks and sounds angry
<code>I don't think so, you already started!</code>	
<code></speak></code>	
<code></scene></code>	->The agents disappear
<code></page></code>	
<code></mood></code>	
<code></seq></code>	
<code></body></code>	->End of the presentation

Demonstration 1: A simple MPML script illustrates the effect of emotional tags.

Below is the exact description of these tags:

Agent: This is the definition of an agent. It gives an agent an id, a voice, a personality and an initial position. This agent will then be used in the presentation.

```
<AGENT id="merlin" character="merlin"
spot="spot2" voice="{C2D4EE00-B025-11D4-
BE23-0000F44780AB}"
personality="agreeable,active"/>
```

Mood: This is a tag to define the mood of the agents. It can contain a large part of the presentation, as moods are often stable over a long period of time. It can however change when required by the action involved in the presentation, especially when some emotions occur. The author has to decide how to use it well.

```
<MOOD assign="merlin:happy,rocky:neutral">
...
</MOOD>
```

Emotion: This is a tag defining a short event. For example if an agent gets angry or is very pleased by a good answer from the user (who can be a student), it will make some gestures to demonstrate the emotion and its voice will also change for a while. Then the emotion intensity decreases and the emotion stops. The personality influences the variation of the

emotion intensity (an agreeable agent will stay angry shorter for example).

```
<SPEAK agent="merlin">  
  <EMOTION assign="merlin:angry"/>  
  You stole my presentation Rocky! ...  
</SPEAK>
```

4. Conclusion

We presented MPML as a mark-up language to create Web-based presentations using interface agents. MPML enables authors to create more attractive WebPages without too much difficulty, as MPML is easy to use. The script will describe the presentation flow, and it is possible to interact with the background pages to have more powerful content. We can use one or several agents, which will speak and makes attractive gestures to communicate with the user. The agents will also show emotion and mood, and will be directed by their own personality to become unique.

The interface agent system we mainly use has been Microsoft Agent but we are now developing interfaces with new systems such like 3D faces and VRML Agents to enhance the use of MPML for presentation with interface agent. We also plan to begin an evaluation period for our system to be sure the emotion system is convincing and to adjust it depending on the results.

5. References

- [1] Ananova Web page, <http://www.ananova.com>
- [2] Andre, Klesen, Gebhard, Allen and Rist, Exploiting Models of Personality and Emotions to Control the Behavior of Animated Interactive Agents, Agents2000 Workshop, 2000.
- [3] Andre, Rist, Presenting through Performing: On the Use of Multiple Animated Characters in Knowledge-Based Presentation Systems, International Conference on Intelligent User Interface 2000.
- [4] Ball, Breese, *Emotion and Personality in a Conversational Agent*, Embodied Conversational Agents, MIT Press, page 189-219, 2000.
- [5] Bellcraft Web page, <http://www.bellcraft.com/mash/>
- [6] Bickmore, Cook, Churchill, Sullivan, *Animated Autonomous Personal Representatives*, Autonomous Agent 1998.
- [7] Cahn, *The Generation of Affect in Synthesized Speech*, MIT Press, 1990.
- [8] Cassel, *Nudge Nudge Wink Wink: Elements of Face-to-Face Conversation for Embodied Conversational Agents*, Embodied Conversational Agents, MIT Press, 2000.
- [9] Digital city project, <http://www.digitalcity.gr.jp/openlab/agentpage.html>
- [10] Dohi, Ishizuka, *Visual Software Agent*, <http://www.miv.t.u-tokyo.ac.jp/dohi/ijcai97-ims/paper.html>
- [11] Ekman, *An Argument for Basic Emotions*, in *Cognition and Emotion*, page 169-200, 1992.
- [12] Lester, Towns, Callaway, Voerman, and FitzGerald, *Deictic and Emotive Communication in Animated Pedagogical Agents*, Embodied Conversational Agents, MIT Press, 2000.
- [13] McCrae, John, *An introduction to the five-factor model and its applications*, *Journal of Personality* 60, page 175-215, 1992
- [14] Microsoft Web page, <http://www.microsoft.com>
- [15] Moon, Nass, *Adaptive Agents and Personality Change: Complementarity versus similarity as forms of adaptation*, CHI 1996.
- [16] Murray, Arnott, *Implementation and testing of a system for producing emotion-by-rule in synthetic speech*, *Speech Communication* 16, 1995.
- [17] Ortony, Clore, and Collins, *The Structure of Emotions*. Cambridge University Press., 1988.
- [18] Prendinger and Ishizuka, *Carrying the Role-Playing Metaphor to Interactive Learning Environments*, Proceeding of IUI2001, accepted as Poster/Demonstration, to appear.
- [19] Reeves, Nass., *The Media Equation*, 1996.
- [20] Rosbottom, *Synchronised Slides 'n Sounds On-line*, ED-MEDIA, 1999.
- [21] Rousseau, Hayes-Roth, *A Social-Psychological Model for Synthetic Actors*, Autonomous Agent 1998.
- [22] Winteractive Web page, <http://www.winteractive.fr>
- [23] Wong, *Selecting Internet Technologies to Support Interactive Teaching and Learning at a Distance*, ED-MEDIA, 1999.
- [24] XML homepage, <http://www.w3.org/XML/>