An Architecture for Multimodal Environment Agents

Antonio Camurri, Alessandro Coglio, Paolo Coletta, and Claudio Massucco

DIST – Dipartimento di Informatica, Sistemistica, e Telematica
Università di Genova
Viale Causa 13, I-16145 Genova, Italy
http://musart.dist.unige.it

Abstract. *An architecture to build software agents for interactive, real-time Multimodal Environments (MEs) is presented. Typical applications include virtual musical instruments, theatrical automation, dance-generated or –controlled music, intelligent human-machine interfaces. Agent outputs are produced from agent inputs through three kinds of elaboration: reactive, emotional, and rational. Sensor fusion (input processing) and media integration (output processing) are two main features of the proposed architecture.*

1. Introduction

Interactive Multimodal Environments (MEs) are active spaces capable to observe users (e.g. movement and gesture of dancers/performers) and to establish high-level communication with them. An ME should be able to observe users in their general, full-body, context-dependent movement and gesture (as well as in their audio, voice, music output), in a sort of Gestalt approach. To accomplish these goals, it is often necessary to dynamically adapt the focus of attention to local, fine details of movement (and sound) as well as to very general ones.

The paradigm we adopt in our ME architecture is *a human observer of the dance (or music performance)*, where the focus of attention changes dynamically according to the evolution of the dance and of the music produced. MEs should therefore be able to change their social interaction and rules over time.

In short, the ME scenario can vary from adaptive virtual musical instruments to dance/music interactive systems, and include machines interacting with humans like robots navigating on stage and effectors in general (Camurri and Ferrentino 1997).

From an engineering viewpoint, an ME is a system consisting of hardware devices and software applications. The software component of a ME can be partially realized as a population of (communicating) agents. This paper describes a general architecture for these agents. Such an architecture stems from a requirement analysis for ME agents, as well as from the experience of software systems and applications previously developed in our laboratory in the HARP project (Camurri 1995). The architecture proposed here is a new model, employed in the ME for the permanent exhibition "Città dei bambini" (Camurri, Dondi and Gambardella 1997) for interactively teaching music and science and for this Workshop concert.

Each ME agent interacts with the external world[1] by obtaining inputs (e.g., data from movement sensors, messages from other agents, etc.) and producing outputs (e.g., sounds, animations, messages to other agents, etc.). In order to allow richer and more stimulating interactions between agents and human users, outputs are produced from inputs through three different kinds of elaboration:

1) *reactive* elaborations, which map inputs to outputs quite directly and instantaneously (e.g. a movement directly produces a sound);
2) *emotional* elaborations, which use inputs to modify an emotional state which influences the outputs (e.g. slow and wide movements give rise to a happy and relaxed mood which determines a "modulation" of the agent's music output toward a particular timbre);
3) *rational* elaborations, which use inputs to update a high-level knowledge about the external world and/or the agent itself, and produce outputs based on inferences performed on such a knowledge (e.g. the agent composes in real time based on its musical knowledge and external situations).

As we will see, these three elaborations do not operate in isolation, but can influence each other in various ways (e.g. reactive elaborations may vary according to the emotional state).

2. Overall Structure

The overall structure of an ME agent is depicted in Figure 1.

A rectangle represents an active component of the agent. There are five such components:

1) *input component* (*IN* for short);
2) *output component* (*OUT* for short);
3) *reactive component* (*RC* for short);
4) *emotional component* (*EM* for short);
5) *rational component* (*RT* for short).

A white (and thick) arrow from a component *A* to a component *B* represents a FIFO buffer of data upon

[1] Here by "external world" we mean all the software of the ME which is external to the agent (e.g. the device drivers, as well as the other agents).

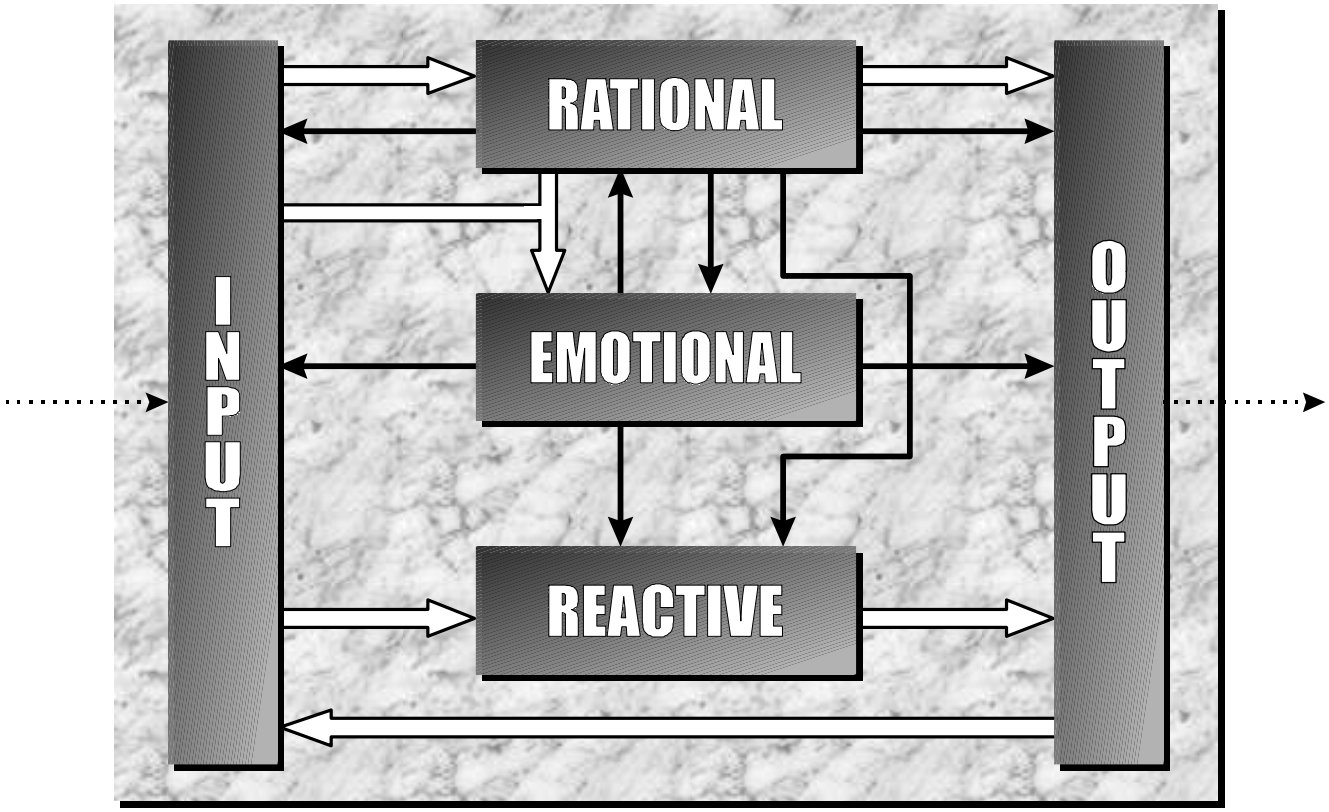

Figure 1: Overall structure of the agent.

which *A* acts as a producer and *B* as a consumer. There are five such buffers:

1) from IN to RC, containing *reactive inputs*, i.e. objects of a set In_{RC};
2) from RC to OUT, containing *reactive outputs*, i.e. objects of a set Out_{RC};
3) from IN to RT, containing *rational inputs*, i.e. objects of a set In_{RT};
4) from RT to OUT, containing *rational outputs*, i.e. objects of a set Out_{RT};
5) from OUT to IN, containing *internal feedbacks*, i.e. objects of a set *IF*.

A double-tail white (and thick) arrow from two components *A* and *A′* to a component *B* represents a FIFO buffer of data upon which *A* and *A′* act as producers and *B* as a consumer. There is just one such buffer, the one from IN and RT to EM, containing *emotional stimuli*, i.e. objects of a set *ES*.

A black (and thin) arrow from a component *A* to a component *B* represents a data container upon which *A* has write access and *B* read access. There are eight such containers:

1) from EM to IN, containing an *emotional-input parameter*, i.e. an object of a set $P_{EM,IN}$;
2) from EM to OUT, containing an *emotional-output parameter*, i.e. an object of a set $P_{EM,OUT}$;
3) from EM to RC, containing an *emotional-reactive parameter*, i.e. an object of a set $P_{EM,RC}$;
4) from EM to RT, containing an *emotional-rational parameter*, i.e. an object of a set $P_{EM,RT}$;
5) from RT to IN, containing a *rational-input parameter*, i.e. an object of a set $P_{RT,IN}$;
6) from RT to OUT, containing a *rational-output parameter*, i.e. an object of a set $P_{RT,OUT}$;
7) from RT to RC, containing a *rational-reactive parameter*, i.e. an object of a set $P_{RT,RC}$;
8) from RT to EM, containing a *rational-emotional parameter*, i.e. an object of a set $P_{RT,EM}$.

A dashed arrows just represents a generic data flow whose nature we do not describe in detail. There are two such data flows:

1) from the external world to IN;
2) from OUT to the external world.

Of course, in general the sets In_{RC}, Out_{RC}, In_{RT}, Out_{RT}, *IF*, *ES*, $P_{EM,IN}$, $P_{EM,OUT}$, $P_{EM,RC}$, $P_{EM,RT}$, $P_{RT,IN}$, $P_{RT,OUT}$, $P_{RT,RC}$, and $P_{RT,RC}$ vary across different ME agents.

3. How the Components Interact

Let us now see how the five components of an ME agent interact with each other and with the external world.

3.1 Input Processing by IN from the External World to RC, EM, and RT

IN elaborates data from the external world to produce reactive inputs for RC, emotional stimuli for EM, and rational inputs for RT. Examples of reactive and

rational inputs are positions, speeds, recognized movement features, symbolic gestures, etc.
The working of IN is parameterized over emotional-input parameters. For example, consider a robot navigating on stage which encounters a human along its path: this situation can give rise to a certain emotional stimulus if the robot is in a sad and depressed mood, to an opposite stimulus if the robot is in an elated and conceited mood. This is achieved by having EM reflect the current emotional state into the emotional-input parameter.
The working of IN is also parameterized over rational-input parameters. For instance, the time slices over which IN integrates sensor data to extract movement features, can vary dynamically, e.g. on the basis of the "quality" of movement recognition: a decreased quality (e.g. different recognition modules producing conflicting data) can cause RT to instruct IN (through the rational-input parameter) to vary its time slice to improve the recognition quality. In other cases, RT could cause a substitution of recognition modules of IN, if necessary.

3.2 Output Processing by OUT from RC, EM, and RT to the External World

OUT elaborates reactive outputs from RC and rational outputs from RT to produce data for the external world. Examples of reactive and rational outputs are (data representing) musical notes or excerpts, commands to a navigating robot (e.g. "stop", "turn left", "slow down to speed *x*"), etc.
The working of OUT is parameterized over emotional-output parameters. For example, a same note or excerpt can be performed with different timbre, interpretation. As another example, a robot could avoid an obstacle in either a timid or a bold way. Note that EM, unlike RC and RT, does not produce any "emotional output" for OUT, but it modulates the behavior of OUT through the emotional-output parameter.
The working of OUT is also parameterized over rational-output parameters. For instance, RT could control the volume (and possibly brightness) of music according to facts inferred by reasoning about the behavior of observed humans. Note that RT can directly control the behavior of OUT through both rational outputs and the rational-output parameter: the former should be used to issue "commands" to OUT, the latter to modulate the effects of such commands on the external world.

3.3 Reactive Processing by RC Modulated by EM and RT

RC elaborates reactive inputs from IN to produce reactive outputs for OUT.
The working of RC is parameterized over emotional-reactive parameters. For instance, a robot encountering an obstacle might utter slightly different exclamations in different emotional states.
The working of RC is also parameterized over rational-reactive parameters. Consider for example a human making nervous movements in a particular zone of the space. After some time, these may result in the "creation" of a virtual percussion instrument in that zone of the space, so that subsequent movements there produce percussion sounds. While the creation of the virtual instrument is carried out by RT, the direct production of sounds from movements is carried out by RC. So, RT instructs RC to react in this way to movements through the rational-reactive parameter.

3.4 Interaction between Rational Processing by RT and Emotional Processing by EM

RT can send emotional stimuli to EM as a consequence of the agent reasoning about itself (and possibly about the external world). For instance, becoming aware that some of the agent's goals have been fulfilled could cause RT to rise a positive emotional stimulus for EM (or a negative one in case the goals have not been fulfilled).
The working of RT is parameterized over emotional-rational parameters. This modulation of RT by EM could take place in two different ways. First, (a part of) the emotional-rational parameter could contain explicit facts about the emotional state, so that RT can explicitly reason about such facts. Second, (another part of) the emotional-rational parameter could affect the working of the inference engine of RT (e.g. as a kind of "perturbations") so that RT might infer different consequences from a same set of facts in different emotional states.
The working of EM is parameterized over rational-emotional parameters. For instance, a robot might control the dynamic of variation of its emotional state (i.e. a same emotional stimulus could produce more or less great variations of the emotional state) depending on inferred facts about the external situation, e.g. the dynamic could be faster in situations of closer interactions with humans.

3.5 Feedback from OUT to IN

Besides producing data for the external world, OUT can send internal feedbacks to IN, which can elaborate them to produce reactive inputs, emotional stimuli, and rational inputs for RC, EM, and RT. Internal feedbacks can contain information about the state of OUT. For instance, a rational output may instruct OUT to produce a musical excerpt, and a subsequent reactive output might cause such excerpt to be aborted. In such a case, an internal feedback might be used to notify RT that the excerpt was aborted (and upon notification, RT could for example re-issue the rational output instructing to produce the excerpt).

4. How the Components Work

We now describe the working of the five components IN, OUT, RC, EM, and RT of ME agents.

4.1 The Working of IN

Data from the external world may be obtained by IN in various ways, e.g. by periodically requesting the status of some hardware device, or in an event-driven fashion. Anyway, such data is elaborated by IN to produce reactive inputs, emotional stimuli, and rational inputs. These elaborations can be more or less complex. Examples of information produced by analyzing data from full-body human movement sensors are positions, speeds, accelerations, how frenetic the movement is, how much in tempo the dancer (or part of her body) is, how she occupies the stage space, the smoothness of the movement, the coordination between arms, a qualitative evaluation of her equilibrium and stability, her potentiality to move in the immediate future, etc. This information is typically obtained by integrating over time a number of different sensor data. In typical applications, integrations take place in two different observation time slices, approximately 0.5 – 1 s and 3 – 5 s.

Since an ME usually contains many different hardware devices with which the agents communicate (through device drivers and other non-agent software), IN usually contains various modules, each in charge of obtaining data from one (or sometimes more) devices. Each module produces its own reactive inputs, emotional stimuli, and rational inputs, which are merged into the buffers to RC, EM, and RT. Furthermore, each module has access to the current emotional-input and rational-input parameters, which modulate the elaborations performed by the modules. For example, IN might contain a module receiving data from ultra-sound sensors, another from on-floor analogical pressure sensors, another from other agents, and so on.

4.2 The Working of OUT

OUT elaborates reactive and rational outputs to produce data for the external world. These elaborations can be more or less complex. Examples of information elaborated by OUT are musical notes or excerpts, speech, graphical animations, etc. So, often the effects of reactive or rational outputs last for more or less long time intervals after they are consumed by OUT.

Analogously to IN, also OUT usually contains various modules, each in charge of producing data for one (or sometimes more) hardware devices. Each module consumes its own reactive and rational outputs in the buffers from RC and RT. Furthermore, each module has access to the emotional-output and rational-output parameters, which modulate the elaborations of the modules (e.g. the musical timbre may vary according to the emotional state). For instance, OUT might contain a module producing data for sound synthesizers (e.g. MIDI cards), another for wave oscillators (e.g. to play speech samples), another for video screens, another for lights in a stage, another for other agents, and so on.

4.3 The Working of RC

The working of RC can be described by a function[2]

$$rc \in [In_{\mathrm{RC}} \times P_{\mathrm{RT,RC}} \times P_{\mathrm{EM,RC}} \rightarrow Out_{\mathrm{RC}}^{*}].$$

Each reactive input $in_{\mathrm{RC}} \in In_{\mathrm{RC}}$ produces a finite sequence of reactive outputs $outs_{\mathrm{RC}} \in Out_{\mathrm{RC}}^{*}$ which also depends on the current emotional-reactive and rational-reactive parameters $p_{\mathrm{RT,RC}}$ and $p_{\mathrm{EM,RC}}$:

$$outs_{\mathrm{RC}} = rc\ (in_{\mathrm{RC}}, p_{\mathrm{RT,RC}}, p_{\mathrm{EM,RC}}).$$

The elaborations performed by RC are relatively low-level and hence relatively fast. In fact, they realize the real-time behavior of the agent. So, it is adequate to describe RC by just a function, without any internal state, since there is a direct mapping from reactive inputs to reactive outputs.

The function *rc* can be implemented in a variety of ways, including neural networks, associative mappings, and so on.

4.4 The Working of EM

The working of EM can be described by:

1) a set X_{EM} of *emotional states*;
2) a set $TI \subseteq \mathbf{R}^{+}$ of *time intervals*[3];
3) a function $em \in [X_{\mathrm{EM}} \times ES^{*} \times TI \times P_{\mathrm{RT,EM}} \rightarrow X_{\mathrm{EM}}]$;
4) a function $prm_{\mathrm{EM,IN}} \in [X_{\mathrm{EM}} \rightarrow P_{\mathrm{EM,IN}}]$;
5) a function $prm_{\mathrm{EM,OUT}} \in [X_{\mathrm{EM}} \rightarrow P_{\mathrm{EM,OUT}}]$;
6) a function $prm_{\mathrm{EM,RC}} \in [X_{\mathrm{EM}} \rightarrow P_{\mathrm{EM,RC}}]$;
7) a function $prm_{\mathrm{EM,RT}} \in [X_{\mathrm{EM}} \rightarrow P_{\mathrm{EM,RT}}]$.

The emotional state has a temporal evolution whose discretization is governed by the equation

$$x_{\mathrm{EM}}' = em\ (x_{\mathrm{EM}}, ess, ti, p_{\mathrm{RT,EM}}),$$

where x_{EM} is the emotional state at instant $t \in \mathbf{R}$, x_{EM}' is the emotional state at instant $t + ti$, *ess* is the sequence of emotional stimuli produced by IN and RT between instants t and $t + ti$, and $p_{\mathrm{RT,EM}}$ is the current rational-emotional parameter.

Note that the above equation is quite adequate to describe situations where the emotional state substantially consists in (possibly fuzzy) coordinates (in some emotional space), which change according to some (possibly fuzzy) physical model (e.g. the movement of electric charges), where emotional stimuli give rise to (possibly fuzzy) forces. However, the equation is also adequate to describe situations where there is no such physical metaphor, and the emotional state just evolves step by step (each step being caused by an emotional stimulus) in a time-

[2] If S is a set, we write S^{*} to denote the set of all finite (possibly empty) sequences of elements of S.

[3] $\mathbf{R}^{+}$ is the set of all non-negative real numbers. The elements of *TI* may vary across applications, depending on how time intervals are measured (e.g. as natural numbers or as non-negative floating point numbers).

independent way (in this case the function *em* should "ignore" its argument *ti*).
Two examples of concrete instances of the above formal description can be found in (Camurri and Ferrentino 1997) and in (Camurri, Chiarvetto, Coglio et al 1997).
The emotional-input, emotional-output, emotional-reactive, and emotional-rational parameters are all functionally determined from the current emotional state each time it is updated, respectively by means of the four functions $prm_{EM,IN}$, $prm_{EM,OUT}$, $prm_{EM,RC}$, and $prm_{EM,RT}$, which extract the relevant information from the emotional state.

4.5 The Working of RT

The working of RT can be described by:
1) a set *Asr* of assertions;
2) the set $X_{RT} = Asr^*$ of rational states;
3) a function $asr_{IN} \in [In_{RT} \rightarrow Asr^*]$;
4) a function $asr_{OUT} \in [Asr \rightarrow Out_{RT}^*]$;
5) a function $asr_{EM} \in [Asr \rightarrow ES^*]$;
6) a function $ie \in [X_{RT} \times P_{EM,RT} \rightarrow X_{RT}]$;
7) a function $prm_{RT,IN} \in [X_{RT} \rightarrow P_{RT,IN}]$;
8) a function $prm_{RT,OUT} \in [X_{RT} \rightarrow P_{RT,OUT}]$;
9) a function $prm_{RT,RC} \in [X_{RT} \rightarrow P_{RT,RC}]$;
10) a function $prm_{RT,EM} \in [X_{RT} \rightarrow P_{RT,EM}]$.

The state of RT is a sequence[4] of assertions, which constitute the current knowledge. Rational inputs add assertions to the current knowledge x_{RT}: when a rational input in_{RT} is processed by RT, the current knowledge becomes[5] $x_{RT} \lozenge asr_{IN}(in_{RT})$. In other words, a rational input triggers the addition of some assertion to the current knowledge. The inference engine of RT is modeled by the function *ie*, with each inference step mapping a knowledge x_{RT} to a new knowledge x_{RT}' also depending on the emotional-rational parameter $p_{EM,RT}$:

$$x_{RT}' = ie\ (x_{RT}, p_{EM,RT}).$$

Each such inference step also produces a sequence of zero or more rational outputs $outs_{RT}$ and a sequence of zero or more emotion stimuli *ess*, depending on which new assertions have been added, as follows:

$$outs_{RT} = asr_{OUT}(a_1) \lozenge \ldots \lozenge asr_{OUT}(a_n),$$
$$ess = asr_{EM}(a_1) \lozenge \ldots \lozenge asr_{EM}(a_n),$$

where $[a_1, \ldots, a_n]$ is the sequence of all assertions present in x_{RT}' but not in x_{RT} (in the same relative order they have in x_{RT}'), that is the assertions newly produced by the inference step. In other words, inferences trigger the production of rational outputs and emotional stimuli.

RT can be implemented in a wide variety of ways. The knowledge can be expressed as first-order formulae, KL-ONE networks, Horn clauses, and so on. Often the knowledge might be split into a fixed part (containing assertions which are always true) and a changing part (containing assertions which are true only at certain times, thus codifying different situations at different times). The inference engine can consist in a theorem prover, a Prolog interpreter, a planner, and so on. Anyway, the general formal description of RT we have given above, captures all these particular cases.
Analogously to EM, the rational-input, rational-output, rational-reactive, and rational-emotional parameters are functionally determined from the rational state after each inference step, respectively by means of the four functions $prm_{RT,IN}$, $prm_{RT,OUT}$, $prm_{RT,RC}$, and $prm_{RT,EM}$.

5. Flow of Control

We complete the description of the architecture by describing how the five components of an ME agent execute, i.e. the flow of control inside the agent.
The computations of the agent are carried out through three threads of control:
1) IN-RC-OUT thread;
2) EM thread;
3) RT thread.

5.1 The IN-RC-OUT Thread of Control

The IN-RC-OUT thread cyclically activates IN, RC, and OUT. In each cycle, IN is first activated. The various modules of IN process data from the external world and put reactive inputs, emotional stimuli, and rational inputs into the buffers to RC, EM, and RT. Then, RC is activated, which consumes all the reactive inputs present in the buffer (just produced by IN), putting reactive outputs into the buffer to OUT. Finally, OUT is activated. Its modules consume all the reactive outputs present in the buffer (just produced by RC), sending data to the external world. Furthermore, if rational outputs are present in the FIFO buffer they are consumed by the modules of OUT as well.
The main reason why there is one thread for IN, RC, and OUT is that RC must realize the real-time behavior of the agent, so it must produce outputs from inputs in time.

5.2 The EM Thread of Control

The EM thread manages the temporal evolution of the emotional state. At each cycle, the emotional stimuli in the buffer are consumed and used to update the emotional state.
In case the emotional state evolves according to some physical metaphor, of course it is necessary to have some lower bound on the cycle rate of EM, otherwise the discretization which approximates a continuous evolution might become too inaccurate.

[4] Of course, the relative ordering of the assertions within the sequence is immaterial from the point of view of the knowledge they express. However, ordering is formally necessary to determine the ordering of the rational outputs and emotional stimuli produced by RT, as we will see shortly.
[5] The operator $\lozenge$ denotes the concatenation operator upon finite sequences.

5.3 The RT Thread of Control

The RT thread manages the inference steps performed by the inference engine. At each cycle, rational inputs are consumed (which introduce new assertions), and new assertions are inferred, which trigger the production of rational outputs and emotional stimuli.
Clearly, there are no strict timing constraints on the rate at which inference steps are performed, except of course the requirement that inferences must be performed fast enough to be useful to the agent.

6. Conclusions and Future Work

The architecture proposed here seems very flexible and powerful to structure ME agents. As mentioned in Section 1, we have employed it in the ME for the permanent exhibition "Città dei bambini". For example, we have instantiated our agent model to control a robot (RWI's Pioneer 1) equipped with on-board audio and remote control facilities, which acts as "cicerone" in the interactive Music Atelier of the exhibition (see Figure 2). The agent can navigate, speak (to help visitors to improve their fruition of the Atelier's interactive exhibits), compose and play music, control environmental lights (which reflect its mood), and produce computer animation. This system can be also used in multimedia concerts.
An interesting feature of our architecture is that it exhibits some analogy to the human nervous system (in fact it realizes the paradigm of a *human* observer, as mentioned in Section 1). The elaborations of IN and OUT are analogous to those performed by nerves and brain respectively upon raw sensorial information (e.g. from eyes and ears) to obtain higher-level information (e.g. recognized shapes and melodies), and upon high-level commands (e.g. to utter a word) to obtain peripheral impulses. The elaborations of RC are analogous to reactions to external stimuli which do not "pass through" conscious rational reasoning (e.g. an exclamation for a surprise). The emotional state of EM, as obvious, is analogous to that of a human, which can influence the ways the human perceives, acts, reacts, and also reasons. The elaborations of RT are of course analogous to human rational reasoning, and the current knowledge can influence the way the human perceives, acts, reacts, and changes mood.
The formal descriptions of the five components of an ME agent, which we have given in Section 4, are very general. An interesting direction in future work is finding more detailed formal descriptions, without of course loosing flexibility and generality in a considerable way. If this were taken to a sufficient extent, we might produce software tools allowing high-level specifications of ME agents, and automatic synthesis of executable agents from specifications.

Figure 2 : The cicerone robot at work.

References

Camurri, A. (1995) Interactive Dance/Music Systems. Proc. *Intl. Computer Music Conference ICMC'95*, Banff, ICMA Press.

Camurri, A. (1997). Network Models for Motor Control and Music. In P.Morasso, V.Sanguineti (Eds.) *Self-Organization, Computational Maps and Motor Control*, Elsevier Science B.V.

Camurri, A., Chiarvetto, R., Coglio, A., Dapelo, R., Di Stefano, M., Liconte, C., Massucco, C., Murta, D., Palmieri, G., Stroscio, A., Trocca, R. (1997). Toward Kansei Information Processing in music/dance interactive multimodal environments. Proc. *AIMI Intl. Workshop on Kansei – The Technology of Emotion*, DIST-University of Genova.

Camurri, A., Dondi, G., Gambardella, G. (1997). Interactive science exhibition: A playground for true and simulated emotions. Proc. *AIMI Intl. Workshop on Kansei – The Technology of Emotion*, DIST-University of Genova.