An Architecture for Emotional Agents

Antonio Camurri
University of Genoa

Alessandro Coglio
Kestrel Institute

We propose a general, flexible, and powerful architecture to build software agents that embed artificial emotions. These artificial emotions achieve functionalities for conveying emotions to humans, thus allowing more effective, stimulating, and natural interactions between humans and agents. An emotional agent also possesses rational knowledge and reactive capabilities, and interacts with the external world, including other agents.

Imagine a space—for example, a museum exhibit or theater stage—in which you can stand surrounded by music and control its ebb and flow (the rhythms, orchestration, and so on) by your own full-body movement (such as dancing) without touching any controls.[1] In addition, imagine that this space also includes mobile robots that can interact with, entertain, sing, and speak to you, as though they were actors on a stage.[2] Figure 1 shows an example of this scenario.

This kind of multimedia-multimodal system requires, among other things, intelligent interfaces and adaptive behavior. We believe that including artificial emotions (in the sense we explain below) as one of the main components of these kinds of complex systems will achieve a more effective, stimulating, and natural interaction with humans. For example, consider a robot navigating on stage that after repeatedly finding you in its way, announces (by playing a digitized voice sample) "Stop blocking my way, I'm angry!" while flashing its "eyes" or announces "He hates me, I'm so sad…" while slowly moving to a corner. Clearly, the quality and effectiveness of the interaction between the robot and you is greatly enhanced by this kind of behavior because it relates (at least externally) to human behaviors associated with emotions. In fact, the robot proves more "believable."[3] To implement these behaviors in the software controlling the robot, we can codify an emotional state that evolves over time, is influenced by external events, and influences the robot's actions.

Consider, as another example, the widely accepted belief that music and dance communicate mostly emotional content, embedded in the interpretation and expression of the performers' intentions.[4] Therefore, an active space producing music from full-body movement can try to capture movement features focusing on the discovery of interpretation and emotionally related content in the performance[5] and embed emotional content in the music produced. This can be implemented by codifying some emotional state that movement influences, which in turn influences the music.

In this article we propose a general architecture[6] to build *emotional agents*—software agents that possess an emotional state (in the sense above). An emotional agent interacts with the external world by receiving inputs and sending outputs, possibly in real time. The emotional state evolves over time; it influences outputs and its evolution responds to inputs. The agent also possesses a rational knowledge (for example, about the external world and the agent's goals) that evolves over time, whose evolution depends on inputs and produces outputs. In addition, some outputs are produced reactively from inputs in a relatively direct and fast way (possibly in real time). These emotional,

Figure 1. A sample scenario of an active space observing the dancer's movements and gestures. The dancer can generate and control music as well as communicate to a robot navigating and performing on stage.

1070-986X/98/$10.00 © 1998 IEEE

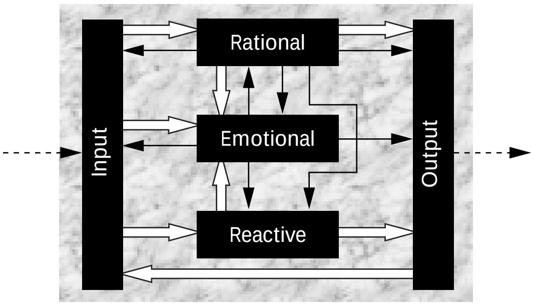


Figure 2. Overall structure of an emotional agent.

Table 1. Buffers of the agent (thick white arrows).

From	To	Contains
Input	Reactive	Reactive inputs
Reactive	Output	Reactive outputs
Input	Rational	Rational inputs
Rational	Output	Rational outputs
Input	Emotional	Emotional stimuli
Rational	Emotional	Emotional stimuli
Reactive	Emotional	Emotional stimuli
Output	Input	Internal feedbacks

rational, and reactive computations don't operate in isolation, but can influence each other in various ways (for example, the emotional state can "modulate" the agent's reactions).

Unlike Sloman's approach,[7] our architecture doesn't attempt to model human or animal agents. Instead, we propose it as a convenient and sound way to structure software agents capable of functionalities such as those sketched previously and those described in the remainder of this article. The motivations for the architecture, therefore, come from the practical behaviors that we intend our emotional agents to implement. They don't come from theories of human or animal behavior, though you can certainly see some naive analogies.[7] The main criterion for evaluating our architecture arises from its convenience in structuring agents whose behaviors convey emotions to the people exposed to the systems.

We're currently employing our architecture mainly in systems involving music, dance, robots, and so on. The software for these applications can be (partially) realized as a population of communicating emotional agents. For instance, there might be an agent for each navigating robot and one or more agents for the sensorized space where humans move. Inputs for these agents include data from human-movement and robot sensors, as well as messages from other agents. Outputs include commands to audio devices and robot actuators, as well as messages to other agents. The emotional states of these agents can be modified, among other factors, by human movements—wide, upward, and fast movements may give rise to a happy and lively mood—and robot interaction with humans—a human blocking a robot's way may generate an angry mood. These states can influence music composition and robot navigation. For example, a happy mood may lead to lively timbres and melodies, while a nervous mood may prompt quick and fast-turning movements.

A number of promising applications have emerged for emotional agents, such as

- Interactive entertainment, such as an interactive discotheque where dancers can influence and change the music, and games like dance-karaoke where the better you dance the better music you get
- Interactive home theater
- Interactive tools for aerobics and gymnastics
- Rehabilitation tools such as agents supporting therapies for mental disabilities where it's important to have nonintrusive environments with which patients can establish creative interaction
- Tools for teaching by playing and experiencing in simulated environments
- Tools for enhancing the communication about new products or ideas in conventions and "information ateliers" such as fashion shows
- Cultural and museum applications that involve autonomous robots that act as guides to attract, entertain, and instruct visitors
- Augmented reality applications

Description of the architecture

Figure 2 shows the structure of an emotional agent. The five rectangles represent active components. A thick white arrow from one component to another component represents a buffer upon which the first component acts as a producer and the second as a consumer. (Table 1 lists the eight buffers.) A thin black arrow from one com-

Table 2. Data containers of the agent (thin black arrows).

From	To	Contains
Emotional	Input	Emotional-input parameter
Emotional	Output	Emotional-output parameter
Emotional	Reactive	Emotional-reactive parameter
Emotional	Rational	Emotional-rational parameter
Rational	Input	Rational-input parameter
Rational	Output	Rational-output parameter
Rational	Reactive	Rational-reactive parameter
Rational	Emotional	Rational-emotional parameter

ponent to another component represents a data container upon which the first component has read-write access and the second read-only access. (Table 2 lists the eight containers.) Finally, the two dashed arrows represent flows of information, from the external world to Input and from Output to the external world.

In the remainder of this section, we'll describe the five components in more detail and the data through which they interact with each other and with the external world. Note that the architecture provides only conceptual requirements for such components and data without constraining their concrete realizations, which can vary widely across different agents.

The Input component

The Input component obtains inputs from the external world—such as data from movement sensors and messages from other agents—by requesting them (such as periodically polling a device driver) and receiving them (for example, being sent new data when available). The Input component processes these inputs to produce reactive inputs for Reactive, rational inputs for Rational, and emotional stimuli for Emotional. The processing can be more or less complex and depend on some internal state of Input. For example, information produced by analyzing data from full-body human-movement sensors may include kinematic and dynamic quantities (positions, speeds, or energies), recognized symbolic gestures, the degree to which dancers (or parts of their bodies) remain in tempo, how dancers occupy the stage space, and the smoothness of the dancers' movement. Typically, we obtain this information by integrating different sensor data. Reactive and rational inputs may then include descriptions of kinematic and dynamic quantities, gestures, and movement features. Different gestures may produce different emotional stimuli. For example, very smooth movements may produce stimuli opposite to those produced by sharp and nervous movements.

Since Input can read the emotional-input parameter, the information it's processing may depend on the agent's current emotional state. In fact, as we'll see below, Emotional holds the emotional state of the agent and keeps information about it in the emotional-input parameter. In this way, Input can do slightly different actions according to the parameter's current value. For example, consider a robot navigating on stage that encounters a human along its path. When Input recognizes this situation (from the robot sensor data), it generates a certain emotional stimulus if the robot is sad and depressed (because the human can be seen as a helping hand) or it produces an opposite stimulus if the robot is elated and complacent (because the human is just seen as a blocking obstacle).

Analogously, the processing of Input may also depend on the robot's current rational state through the rational-input parameter. This proves useful, for example, to dynamically change the focus of attention of the movement analyzers inside Input. In fact, in different situations (the current situation being encoded in the rational state), different aspects and features of human movement may be relevant instead of others (for example, arm movements instead of leg movements or speed instead of energy).

The Input component also receives and processes internal feedbacks coming from Output. As we'll see, they provide information about Output's state changes to Input and hence indirectly to Reactive, Rational, and Emotional (through reactive inputs, rational inputs, and emotional stimuli generated by Input as a result of processing internal feedback).

Since Input can receive various kinds of inputs from the external world and produce various kinds of reactive inputs, emotional stimuli, and rational inputs, Input usually contains various modules, each in charge of performing a certain kind of processing. Of course, the modules must be properly orchestrated inside Input so that they operate as an integrated whole ("conflicting" information must not be produced from different sensor data).

The Output component

Output sends outputs—such as audio data and messages to other agents—to the external world, for example, by issuing commands to a device driver. This component produces these outputs by processing reactive outputs, rational outputs, and the current emotional-output and rational-output parameters. Examples of reactive and rational outputs from which outputs for sound device drivers derive include descriptions of actions to play single notes, musical excerpts, and digitized speech. As we'll see, the emotional-output and rational-output parameters contain information about the current emotional and rational states. So the processing of Output may depend on these two states. For instance, the same musical excerpt can be played with different timbres in different emotional states (a bright timbre for happiness, a gloomy one for sadness) and different volumes in different rational states (a higher volume for a robot loudspeaker in situations where humans are far from the robot).

Output also sends internal feedbacks to Input, as we mentioned above. Internal feedbacks contain information about state changes of Output that must be sent to the agent's other components. Consider a rational output instructing Output to play a musical excerpt and a subsequent reactive output that, as a result of a sudden event (for example, a robot being stopped by an obstacle), instructs Output to abort the excerpt and utter an exclamation. Output can thus generate an internal feedback signaling the abort, which is forwarded to Rational. This might cause Rational to reissue the rational output to play the excerpt, for example. Without internal feedback, Rational would have no way of knowing that the excerpt was aborted and thus no way of replaying it. In general, internal feedbacks assure some communication path between any two components of the agent.

Output can produce various kinds of outputs for the external world and process various kinds of reactive and rational outputs. Therefore, Output usually contains various modules, each in charge of performing a certain kind of processing, analogously to Input. Of course, the modules must be orchestrated to operate as an integrated whole ("conflicting" commands must not be sent to the same device driver).

The Reactive component

Reactive processes reactive inputs from Input and produces reactive outputs for Output and emotional stimuli for Emotional. Reactive processes information relatively fast, and it has little or no state. In fact, if the agent exchanges data with the external world in real time (as is customary in music and dance applications), Reactive realizes (together with some modules of Input and Output) the agent's real-time behavior.

Reactive can perform a rich variety of computations because they depend on the emotional and rational states through the emotional-reactive and rational-reactive parameters (for instance, such parameters can change the hardwiring of reactive inputs and outputs). For example, a robot encountering an obstacle might utter slightly different exclamations in different emotional states. Another interesting example is that of virtual musical instruments, where, for instance, a human's repeated nervous and rhythmic gestures (evoking the gestures of a percussionist) in certain locations of the space continuously transform neutral sounds into drum-like sounds. That is, virtual percussion instruments emerge in those locations, and subsequent movements in each location produce percussion sounds. While Rational creates virtual instruments, Reactive produces the sounds from movements. To do that, Reactive must know which virtual instruments reside in which locations. Rational provides this information to Reactive via the rational-reactive parameter.

These emotional, rational, and reactive computations don't operate in isolation, but can influence each other in various ways.

The Rational component

Rational holds the agent's rational state, which consists of some knowledge about the external world (how humans are moving, which virtual instruments have been created, and so on) and the agent itself (the goals the agent must achieve). This knowledge evolves by some inference engine, and its evolution depends on the processing of rational inputs from Input and produces rational outputs for Output. Rational's knowledge and inference engine can be more or less complex—from state variables and transitions to symbolic assertions and theorem proving, and sometimes a mix of different things such as hybrid models.[1] In contrast to Reactive, Rational generally has no strict timing constraints (except that rational outputs must be produced in time to be useful), and therefore its computations can be quite complex.

Rational's evolving knowledge can also depend on the current emotional state through the emotional-rational parameter. This dependence usually takes place in the following ways:

1. The parameter encodes knowledge about the emotional state (usually in the same form as Rational's other knowledge) so that the inference engine of Rational also operates on this "emotional knowledge" (without modifying it, since Rational has read-only access to it).

2. The parameter affects the inference engine (as a kind of "emotional perturbation") so that the same knowledge evolves differently in different emotional states.

Of course, these two mechanisms are not mutually exclusive.

On the other hand, the evolution of the emotional state can also depend on the current rational state through the rational-emotional parameter, as well as on its temporal evolution through explicit emotional stimuli for Emotional produced by Rational (analogous to rational outputs). For instance, becoming aware that some of the agent's goals have been fulfilled might cause Rational to give a positive emotional stimulus for Emotional (or a negative one in case the agent failed to achieve its goals).

Of course, part of Rational's processing consists in updating the rational-input, rational-output, rational-reactive, and rational-emotional parameters to reflect changes in its knowledge.

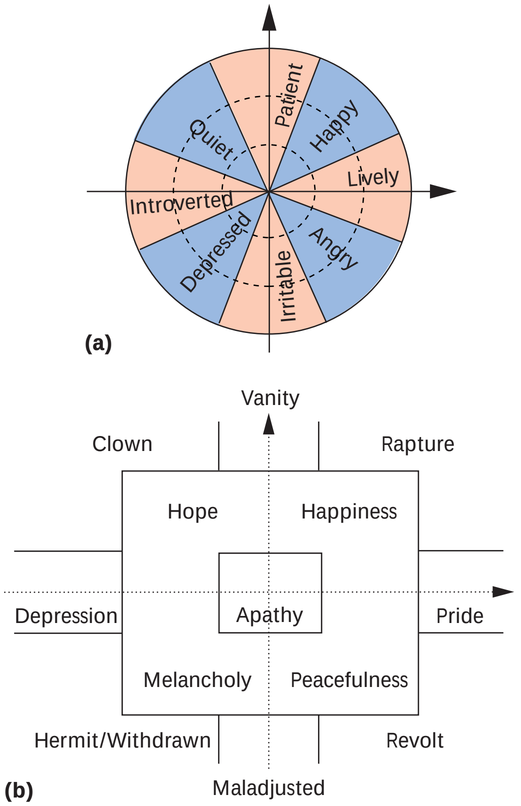


Figure 3. Two examples of emotional spaces.

The Emotional component

Emotional holds the agent's emotional state, whose temporal evolution is governed by emotional stimuli from Input, Reactive, and Rational. An interesting computational realization of an emotional state, among the many possible, consists in coordinates of a point in some "emotional space." The coordinates can change step by step according to emotional stimuli or even according to some physical metaphor where emotional stimuli constitute forces acting on a point mass. The emotional space is usually partitioned into zones characterized by synoptic, symbolic names such as happiness, sadness, and excitement. So, different emotional stimuli tend to move the mass toward different zones. Borders between zones, as well as emotional coordinates and forces, can be fuzzy. Figures 3a and 3b show two emotional 2D spaces of these kinds of computational realizations.[2,8] While we have found these examples of Emotional very useful and interesting, we stress that our architecture makes no commitment to them and in fact allows very different realizations of Emotional.

Unlike Reactive and Rational, Emotional does not produce any "emotional output" for Output. However, the state of Emotional influences the processing of Output through the emotional-output parameter. This parameter, as well as the emotional-input, emotional-reactive, and emotional-rational parameters, contains information about the emotional state (such as the name of the zone where the mass lies or a fuzzy vector of a point coordinate membership to zones), which Emotional updates as its state changes.

In addition to receiving emotional stimuli from Rational, Emotional can be influenced by the rational state through the rational-emotional parameter. For example, having faster dynamics of a robot's emotional state changes—such as emotional stimuli resulting in greater forces upon the mass—proves useful in situations of close interaction with humans.

An application of the architecture

We're successfully employing our architecture in "Città dei Bambini," (Children's City) a permanent science exhibition at Porto Antico in Genoa, Italy where children, by playing interactively with various devices and games, learn about physics, music, biology, and many other subjects.[9] Our lab conceived and realized the Music Atelier of this exhibition, described in the sidebar "The Music Atelier of Città dei Bambini." We modeled the software for our Atelier as a population of five communicating emotional agents, one for each game. Messages exchanged by agents—besides informing the robot how visitors are playing the other games—help the agents make adjustments to the system dynamically (such as adjusting sound volumes of different games to avoid interference). In addition, the agents help the Atelier work as an integrated game by making the current

The Music Atelier of “Città dei Bambini”

“Città dei Bambini,” (Children’s City) a 3,000-square-meter interactive permanent science museum exhibit for children, recently opened at Porto Antico in Genoa, Italy. The exhibit consists of two main modules, one developed by La Villette (Paris) and the other by Imparagiocando (formed by the University of Genoa, National Institute for the Physics of Matter, National Institute for Cancer Research and Advanced Biotechnology Center, Arciragazzi, and the International Movement of Loisirs Science and Technology). The Music Atelier, part of the latter module, was conceived and realized by the Laboratory of Musical Informatics of the Department of Informatics, Systems, and Telecommunications (DIST) at the University of Genoa. The Atelier (see Figure A) consists of five games characterized by multimedia-multimodal interaction involving music, movement, dance, and computer animation.

In the first game, called Let Your Body Make the Music, visitors can create and modify music through their full-body movement in an active sensorized space. This biofeedback loop helps visitors learn basic music concepts such as rhythm and orchestration. This game also serves as a “buffer” for visitors. In fact, it can be played by a group of visitors while they wait their turn for the rest of the Atelier, which no more than five people can visit simultaneously.

The second game, Cicerone Robot, features a navigating and speaking (as well as singing) robot (see Figure B) that guides groups of up to five visitors through the remaining three games, explaining how to play, while simultaneously interacting with the visitors. The robot can change mood and character in response to whatever is happening in the Atelier. For instance, it gets angry if visitors do not play the other games in the way it explained the games to them. Its emotional state is reflected in the way it moves (swaggering, nervous, or calm), its voice (inflection; different ways of explaining things, from short and tense to happy and lively), the music it produces (happy, serene, or noisy), and environmental lights in the Atelier (when sad, the light becomes blue and dim; the angrier it gets, the redder the light becomes). This game aims to introduce children to artificial intelligence concepts and dispel the myth of science-fiction robots by introducing them to a real robot.

continued on p. 30

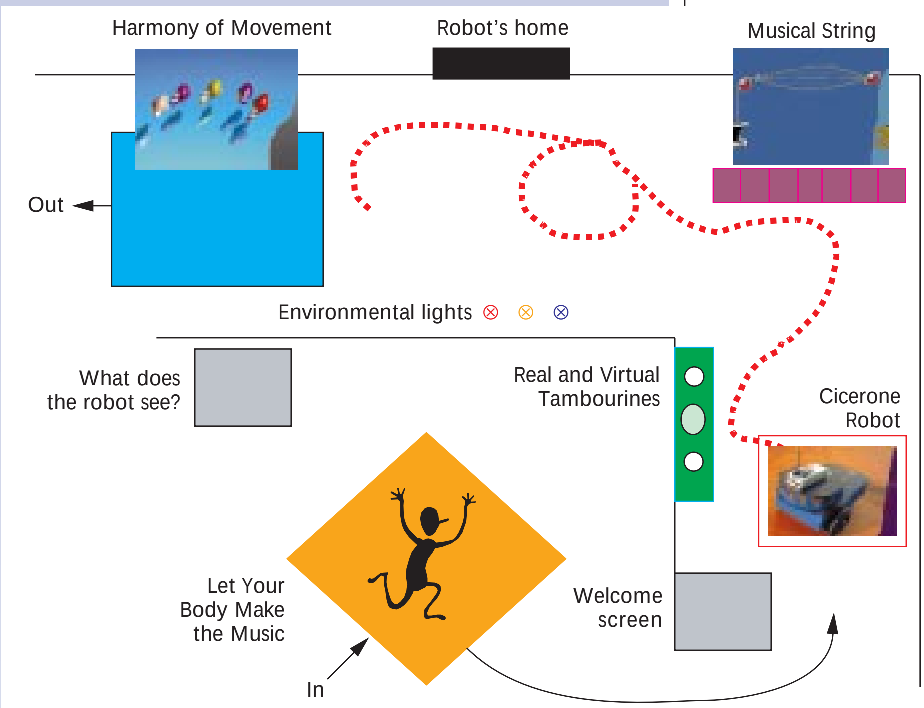


Figure A. The map of the Music Atelier at “Città dei Bambini.”

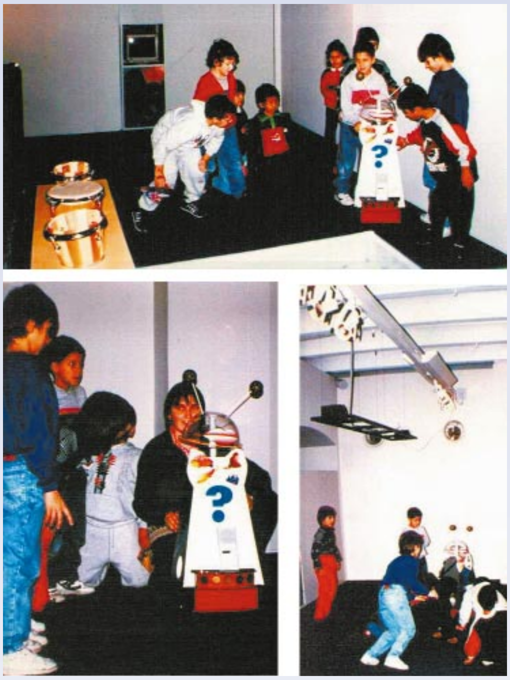


Figure B. The Cicerone Robot, the guide (or games partner) for visitors, at work in the Music Atelier at “Città dei Bambini.”

continued from p. 29

The game Real and Virtual Tambourines consists of a set of three percussive musical instruments. Only one has a real membrane, while the other two have a computer-simulated sound source (vibration of the membrane) controlled by sensors. This game introduces visitors to virtual instruments.

Musical String explores the nature of musical timbre (see Figure B3 for the physical structure on the ceiling for this game). Visitors' movements excite a (simulated) string, then dynamically change parameters of its physical model. The results can be heard and seen on a computer screen showing an animation of the vibrating string.

Harmony of Movement explores some principles of musical language without using traditional notation. Instead it employs a visual metaphor with animated characters (little colored fishes) on a computer screen instead of notes on a stave. Visitors experiment and reconstruct a simple five-voice musical piece by means of their movements, receiving musical and visual feedback. Each voice corresponds to a fish. As soon as a visitor stands on one of the five colored tiles ("teletransport stations") arranged in a semicircle, a fish of the same color appears on screen and the melody associated with that character begins to take form. The fish's "alter ego" moves around and plays the melody following the visitor's movements on the teletransport station. After some time, the fishes form a shoal, and the visitors can move all over the semicircular area, working together to guide the shoal. This game aims to obtain the best musical result possible, which also corresponds to the most harmonic and coordinated movement of the shoal, halfway between total chaos and perfect order.

game's operation depend on how previous games were played. This ensures coherence and continuity in a visitor's tour of the Atelier.

We implemented the emotional agent for the robot and found that children are generally impressed by its performance. For the other games, we initially realized "nonemotional" applications, which we're now extending toward full emotional agents as part of the continuing growth of the Music Atelier. Our architecture has proven useful in designing and implementing the other four agents in a clear and reusable way. These agents and applications are written in C++ and run on the Microsoft Windows NT operating system. Next we'll describe the main features of the robot agent's current implementation.

The emotional agent for the robot

Physically, the robot is an ActiveMedia Pioneer 1, which we dressed and equipped with on-board remote-controlled loudspeakers and additional infrared localization sensors. The Pioneer 1 comes with Saphira navigation software (see http://www.ai.sri.com/~konolige/saphira), which provides a high-level interface to control robot movements. Inputs and outputs for the agent include data from and to this navigation software. Inputs also include messages from the Atelier's other software applications. In addition, outputs consist of speech and music data for the on-board loudspeakers and control data for environmental lights.

The robot's emotional state is a pair of coordinates in a circular emotional space shown in Figure 3a.[8] The horizontal coordinate measures the robot's self-esteem, the vertical one the esteem of other people (that is, visitors of the Atelier) for the robot. These coordinates divide the space into eight sectors of moods. The circle also can be divided into three concentric zones corresponding to the intensity of the moods (weak, average, and strong). So, the coordinate space has 24 distinct zones.

Emotional stimuli are carrots (rewards) and sticks (punishments) either from the robot or from people. Generally, a carrot or stick from the robot moves the point in the coordinate space right or left (because self-esteem increases or decreases), while a carrot or stick from people moves the point up or down (for analogous reasons). The dynamics of this model are nonlinear. When the point lies in the happiness sector and in the external (strong) ring, a carrot has little or no effect, while even a single stick can catastrophically move the point into an opposite zone. For example, carrots and sticks from people are generated by Input for Emotional from messages coming from the other games—carrots if visitors are playing them as the robot explained, sticks otherwise. Carrots and sticks from the robot originate from Rational for Emotional, respectively, when the robot achieves a goal (such as reaching a location) and when it cannot achieve it (for instance, because visitors block its way).

In the absence of carrots and sticks, the point moves toward a spot in the happiness sector (that

How the Robot Conveys Emotions and How Visitors React

The robot communicates to children through lights, style of movement, speech, sound, and music. Such outputs depend on the emotional state and how the visit is going so far. Groups of colored lights (yellow, red, and blue) are displaced in the Atelier (see Figure B3). A happy, lively robot will produce full lights of all colors—red means angry, blue depressed, yellow serene, and so on. Further, the robot embeds an on-board small light that pulses at a frequency related to its emotional state and tension. Such a light is hidden inside the robot so that only gleams of pulsing light remain visible. This helps children remember the robot is “active” even when it’s standing still (for example, it may feel apathetic or be in a difficult navigational situation).

The “style” of movement is another important communication channel. For example, the robot can go from, say, moving in a straightfoward, courageous manner (getting close to obstacles before avoiding them) to a slow, shy-like pace (it doesn’t want to reach its goal) to a tail-wagging, gregarious gait (it might deviate from its navigational path or evoke some kind of “dance”). Tail-wagging usually means an open, friendly, lively, and happy mood. Slow versus fast and sharp versus smooth trajectories are important for expressing the robot’s inner tension.

Different spoken sentences can describe the same content. Furthermore, the robot’s speech is slightly modified by modulating it with intensity dynamics or using different sound source(s) with digital audio filtering (such as a harmonizer). For example, for an important message, the on-board loudspeaker system is reinforced by the other games’ environmental loudspeakers. This makes the voice louder and dynamically spatialized to command children’s attention in critical or joking situations.

Music and environmental sounds are also very important and strictly related to the robot’s emotional state. A computer dynamically re-creates and re-orchestrates music in real time, using ad hoc compositional rules conceived for the musical context. Further, the music the robot generates takes into account the sound and music of the other games (agents) with which the robot interacts. This means that all games possess rules concerning their musical interaction, that is, their “merging” and orchestration when the robot enters the audio range of another game. The overall model is cooperative-competitive: a music output can be temporarily turned off to emphasize another performing a high-priority audio task, or both can be modified to play as two voices of the same music. Therefore, music is not a mere back-

continued on p. 32

is, the robot tends to be happy naturally). A module of Output reflects mood changes as color changes in analogically controlled environmental lights (through the emotional-output parameter). Particular colors are associated with some points in the emotional space (for example, a deep red with a point in the anger sector), and interpolation determines colors for the other points. For more information see the sidebar “How the Robot Conveys Emotions and How Visitors React.”

We realized Rational as a state machine. Rational inputs, generated by Input from data coming from the robot navigation software, cause state transitions. For example, after the robot reaches the string game, its state changes from “moving to string” to “explaining how to play string.” Some state transitions generate rational outputs for Output and carrots and sticks for Emotional (see above). Rational outputs, besides data to the navigation software, include commands to say explanatory sentences about games. The current zone in the emotional space influences the exact choice of the sentence (through the emotional-rational parameter). Different zones result in the same sentences expressed in different moods, such as a sharp tone for the angry sector, a lively tone for the happiness sector, and so on.

Reactive inputs are generated by Input for Reactive when the robot encounters an obstacle, which forces it to stop. When this happens, Reactive generates a reactive output to utter an exclamation. The current Emotional zone determines (through the emotional-reactive parameter) a set of slightly different exclamations, one of which the system randomly chooses to avoid tedious repetitiveness. If Reactive instructs Output to utter an exclamation while the robot utters an explanatory sentence, the latter is aborted (quite a human-like behavior) and an internal feedback is sent to Input so that Rational can instruct Output to say the sentence again.

continued from p. 31

ground effect, but an active component to communicate the overall emotional state.

Finally, to further enhance emotional communication, we're also experimenting (in a recent public art installation) with visual output from the robot. A video camera on top of the robot acquires the face of a child standing in front of it. Our software, Virtual Mirrors, projects a deformation of the face on a large screen reflecting the robot's emotional state—the angrier the robot, the more distorted the mirrored face.

Does our robot communicate emotions to the general audience? Our results from experimental work encourage us to think so. Our experience confirmed that children make a good test audience. They adapt very quickly to new things, but they're also very good at discriminating the faults in your work and explaining it to you more freely than adults. We collected feedback from a large number of children through some written questionnaires and direct verbal communication. Overall, we found that children are impressed by the robot's performance. We observed that small groups of children (up to three) are prone to follow and collaborate with the robot. Also, small groups are often more impressed by the performance. When a full classroom of children enters our installation, the children don't generally understand the robot because it requires more attention. In fact, in our musical installation the robot is left alone with children, and no intermediate person is active in the space.

What about the robot's reactions to children? The robot is "scared" by a large number of obstacles, so it performs poorly if more than three or four children are around. The performance degenerates because the robot can't reach the goal (reaching the next game and explaining it to the children or completing an evolution while navigating in the space). This negative stimulus will eventually make the robot angry and then depressed. With a large number of children, the robot changes its goal. Instead of accompanying the children to the other interactive games, it becomes a game partner, a visitor itself touring around with children until it reaches a calmer situation.

Concluding remarks

The architecture we propose offers a very powerful, flexible, and useful way to structure software agents that embed artificial emotions. The state of Emotional—the agent's emotional state—is affected both by the external world and the agent (through emotional stimuli from Input, Rational, and Reactive) and influences how the other four components work via the four parameters. The state of Rational represents the agent's rational knowledge, which is affected by and affects the external world (through inputs from Input and outputs to Output, respectively) and influences how the other four components work through the four parameters. Reactive reactively interacts with the external world, while Input and Output translate between raw data of the external world and higher level information inside the agent.

We note that the idea of producing outputs from inputs "in parallel" through fast and simple computations (Reactive) and complex and intensive ones (Rational), is not new (see, for example, Ferguson[10]). Our main contribution consists in introducing emotional computations (Emotional) and integrating them with Rational and Reactive. Besides working in multimedia-multimodal systems, our emotional agents seem suited to other fields as well (for example, the kind of agents Bates[3] described).

While providing a clear separation of concerns among the agent components and the data exchanged by them, our architecture grants many degrees of freedom in realizing specific agents. As we mentioned, our architecture provides only a conceptual blueprint, since it doesn't mandate how the individual components work or the data they exchange, thus allowing a wide variety of concrete realizations. For example, our architecture puts no restrictions on how the five components execute relative to each other (that is, no restrictions exist on the flow of control inside the agent). Generally, they can run concurrently, and even modules inside Input and Output can run concurrently. When the agent must exchange data with the external world in real time, it's often necessary to execute Reactive and (part of) Input and Output in a single thread of control, which cyclically activates Input, Reactive, and Output with a cycle time sufficiently small to meet the application's real-time constraints.

Our architecture finds its roots in Camurri's past work in several research projects, including the three-year Esprit long-term research project 8579 Miami (Multimodal Interaction in Advanced

Multimedia Interfaces),[1] which resulted in concrete real-world applications such as interactive multimedia-multimodal systems used in public concerts, in Luciano Berio's first performance of his opera "Outis" (at the Opera House Teatro alla Scala, Milan), and in an interactive art installation at Biennale Architettura, Venice. This past work inspired us to imagine how the systems built for those projects would be realized as emotional agents and to ensure that such re-realizations would be natural and convenient. MM

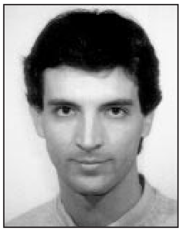
Acknowledgments

We thank Claudio Massucco for useful discussions. We also thank Paolo Coletta, Claudia Liconte, and Daniela Murta for implementing the emotional agent of the robot for "Città dei Bambini."

References

1. A. Camurri, "Network Models for Motor Control and Music," *Self-Organization, Computational Maps, and Motor Control*, P. Morasso and V. Sanguineti, eds., North-Holland Elsevier Science, Amsterdam, 1997, pp. 311-355.
2. A.Camurri, P.Ferrentino "Interactive Environments for Music and Multimedia," to be published in *Multimedia Systems* (special issue on audio and multimedia), ACM and Springer-Verlag, Berlin, 1999.
3. J. Bates, "The Role of Emotions in Believable Agents," *Comm. ACM*, Vol. 37, No. 3, 1994, pp. 122-125.
4. J.A. Sloboda, *The Musical Mind: The Cognitive Psychology of Music*, Oxford University Press, Oxford, 1985.
5. H. Sawada, S. Ohkura, and S. Hashimoto, "Gesture Analysis Using 3D Acceleration Sensor for Music Control," *Proc. Int'l Computer Music Conf. (ICMC 95)*, Int'l Computer Music Assoc., San Francisco, 1995.
6. A. Camurri et al., "An Architecture for Multimodal Environment Agents," *Proc. Italian Assoc. for Musical Informatics (AIMI) Intl. Workshop Kansei: The Technology of Emotion*, AIMI and DIST (University of Genoa), Genoa, Italy, 1997, pp. 48-53.
7. A. Sloman, "Damasio, Descartes, Alarms, and Meta-Management," *Proc. IEEE Intl. Conf. Systems, Man, and Cybernetics (SMC 98)*, IEEE Computer Society Press, Los Alamitos, Calif., 1998, pp. 2652-2657.
8. A. Camurri et al., "Toward Kansei Information Processing in Music/Dance Interactive Multimodal Environments," *Proc. Italian Assoc. for Musical Informatics (AIMI) Intl. Workshop Kansei: The Technology of Emotion*, AIMI and DIST (University of Genoa), Genoa, Italy, 1997, pp. 74-78.
9. A. Camurri, G. Dondi, and G. Gambardella, "Interactive Science Exhibition: A Playground for True and Simulated Emotions," *Proc. Italian Assoc. for Musical Informatics (AIMI) Intl. Workshop Kansei: The Technology of Emotion*, AIMI and DIST (University of Genoa), Genoa, Italy, 1997, pp. 193-195.
10. I.A. Ferguson, "Touring Machines: Autonomous Agents with Attitudes," *Computer*, Vol. 25, No. 5, May 1992.

Antonio Camurri is an associate professor in the Department of Informatics, Systems, and Telecommunications (DIST) at the University of Genoa, Italy. His research interests include computer music, multimodal intelligent interfaces, interactive systems, kansei information processing and artificial emotions, and interactive multimodal-multimedia systems for museum, theatre, and music applications. He earned an MS in electrical engineering in 1984 and a PhD in computer engineering in 1991 from the University of Genoa. He is the founder and scientific director of the Laboratory of Musical Informatics at DIST. He is president of the Italian Association for Musical Informatics (AIMI), member of the executive committee of the IEEE Computer Society technical committee on computer-generated music, and associate editor of the *Journal of New Music Research*.



Alessandro Coglio is a research scientist at the Kestrel Institute in Palo Alto, California, where he is working in the field of formal methods for software engineering. He was previously involved in various research projects at the University of Genoa in the fields of theorem proving (in collaboration with Stanford University, Palo Alto, California), Petri nets and discrete event systems, and emotional agents. He received a degree in computer science engineering from the University of Genoa in 1996.

Readers may contact Camurri at DIST, University of Genoa, Viale Causa 13, 16145 Genova, Italy, e-mail music@dist.unige.it, http://musart.dist.unige.it.