A Brain Inspired Architecture for an Outdoor Robot Guide

Rosamaria Barone, Irene Macaluso, Lorenzo Riano and **Antonio Chella**
Dipartimento di Ingegneria Informatica, Università di Palermo
Viale delle Scienze, 90128, Palermo, Italy
{barone,macaluso,riano}@csai.unipa.it,chella@unipa.it

Abstract

The paper describes the design of a robot architecture which allows the robot to be perceived as a social agent. The proposed architecture autonomously allocates computational resources to different activities via an evaluation of environmental stimuli and robot state based on an emotional model of attention and effort. Moreover, the robot carries out a *dramatization* of its own work, according to drama's theory and its emotions. The architecture has been tested in a project regarding a robotic touristic guide acting in the Botanical Garden of the University of Palermo. This is a challenging application due to crowd of people, large spaces involved and high noise in sensor readings.

Introduction

The current generation of systems for man-machine interaction shows impressive performances with respect to the external shapes, the mechanics and the control of movements; see for example the *Geminoid* android robot developed by Ishiguro and colleagues (Ishiguro and Nishio 2007).

However, these robots, currently at the state of the art, present only limited capabilities of perception, reasoning and action in novel and unstructured environments. Moreover, the capabilities of user-robot interaction are standardized and very well defined.

A new generation of robotic agents, able to perceive and act in new and unstructured environments, should be able to pay attention to the relevant entities in the environment, to choose its own goals and motivations, to decide how to reach them, and to suitably interact with people.

Epigenetic robotics and synthetic approaches to robotics based on psychological and biological models have elicited many of the differences between machine and mental studies of consciousness, while the importance of the interaction between the brain, the body and the surrounding environment has been pointed out. In this interaction process, a main role is played by emotions.

Empirical evidence indicated that emotions are essential for a biological system and they play a substantial function in human intelligence. Emotions are necessary for the survival of the individual and enable animals to be more effective in interacting with their environment (Kelley 2004).

These evidences suggest that an emotional system could be well suited to design robots towards which humans behave socially and that can "survive" in dynamic environments. Many researches have been conducted to achieve these goals. Kismet (Breazeal 2002) is a *social* robot whose emotion-inspired system allows it to interact with people in a socially acceptable and natural manner. (Ortony, Norman, and Revelle 2005) focused on a cognitive architecture where emotions interplay with motivations and cognition in controlling behavior. (Canamero 1997) introduced an architecture for a simulated robot with emotions as a *second order* control mechanism that may affect activation of selected behaviors, attention and perception mechanisms. These works show how the presence of an emotional system can improve robot performance and adaptation capability to dynamic environments. Moreover, recent studies show that providing the robot with the capability to express and understand emotions encourages a natural human-robot interaction (Breazeal 2002; Lisetti et al. 2004).

We designed a brain inspired architecture that allows the robot to autonomously allocate computational resources to different activities via an emotional evaluation of environmental stimuli and robot state. To this aim, we took into account the *theory of attention as a limited resource* proposed by Kahneman (Kahneman 1973) in order to design a *supervisor* block, inspired to the Kahneman model of attention and effort, that plays the role of the allocator of the computational resources of the robot.

Moreover, the emotional system increases the acceptance rate of the robot as a natural partner. In facts, the robot architecture is equipped with an emotional system that gives it the ability of emotional evaluation for each incoming stimulus. This evaluation induces robots emotional states that drive its operations. The robot then carries out a *dramatization* of its own work, according to drama's theory and its emotions: the robot puts up a *drama* in which it is the protagonist character and involves users as characters in the drama.

Experiments on the field show that in this way peoples are pleasantly involved by the robot and the acceptance of the robot as natural partner is increased.

The architecture has been tested on an effective robot architecture implemented on *Robotanic*, the outdoor successor of *Cicerobot* (Chella, Liotta, and Macaluso 2007). Robotanic is an operating outdoor autonomous robot *MobileR-*

obots Pioneer 3-AT using differential drive and equipped with a laser scan range finder, a sonar array, a stereo camera and a GPS.

The robot offers guided tours in the Botanical Garden of the University of Palermo, a public semi-structured outdoor environment, nearly 20000 sq.m. wide, characterized by several pathways sometimes bounded by short walls or plants. The environment is highly dynamic as visitors are all day walking inside the botanical garden, gardeners work alongside the plants and even workers trucks happen to pass by. The live test-bed for the proposed system was during a crowded international conference held in the Botanical Garden in September 2007 where the robot successfully covered several tours.

The Robot Architecture

The whole robot architecture is depicted in Figure 1. The proposed decomposition achieves flexibility and fault-tolerance during both the design and the implementation phase of the architecture.

The Robot System

The main block of the robot system is the *task* block. It represents an action (or a set of actions) which the robot should perform in order to accomplish a given assignment. Tasks may be composed in a recursive fashion. A task needs to be activated and it can be stopped any time. From a computational point of view, a task may be viewed as a thread. Finally, each task employes a set of parameters which define its behaviour; they can be changed at run time, as described below. Task parameters affect its own computational resources: typical parameters are the priority of activation and the allocated memory.

The tasks share a common memory: the *Short Term Memory* (STM). The STM stores information about transient data. It could be used by the tasks as a blackboard to exchange data between tasks or as a common framework for data fusion. For example, the STM may contain a representation of the robot surroundings.

Instead, the *Long-Term Memory* (LTM) block stores long-term information, which may be learned during system operation and/or provided a priori by the designer. Such an information may be, e.g., the environment representation, a set of goal or the procedures to solve a plan.

The task block includes two additional blocks: the *monitor* and the *logic sensors* (LS) blocks. The monitor supervises the task execution, detecting potential faults and activating the necessary recovery routine. If a fault is unrecoverable the monitor reports an error and the task terminates. The task monitors have a role also in the generation process of the primary reinforcers of the emotional subsystem (see below).

The LS elaborates sensors streams to extract high-level perceptual data. Examples of LSs are: laser features extraction tasks or 3D-reconstruction tasks from stereo vision. From a computational point of view, an LS is a particular task with a running context. Also an LS has parameters that can be changed at runtime. We decided to keep tasks and LSs separated to underline the different activities they carry out. LSs may be shared between two or more tasks, providing them with the same information. If some task needs to change the parameters of one LS, a new instance of it will be created to differentiate among them.

The *planner* block receives a command, or a high level goal, and it expands the command in a sequence of actions. Each action is in turn mapped into a particular task. The planner may use an inferential engine, which in turn examines actions preconditions and postconditions to generate a correct plan.

When a task reports an error, the planner has two choices: i) recovering by replanning or skipping the faulty task when possible, or ii) declaring the whole plan invalid and reporting an error. For example, the navigator task, described in the next Sects., could report an error if the robot cannot reach its destination. The planner could then decide to skip the current waypoint and to continue to the next one. But if the current waypoint is the goal one, the planner reports an error as the received command could not be fulfilled in any way.

The Emotional System

There are no accepted definitions of *mood* and *emotions*. However, there is agreement on the fact that mood refers to an affective phenomenon that differs from emotions mainly in terms of its duration. Emotions are very brief, they last few minutes while mood last many days.

Moreover, emotions are triggered by stimuli (internal or external) and they are generally associated with a "physical" expression (facial expression, voice, gesture). Mood is not triggered by a specific stimulus and it is typically a global affective state.

The proposed *emotional system* models both emotions and mood. We take into account suggestions from neuroscience and psychology (Ekman 1992), (Balkenius and Morén 2001), (Rolls 1990). A detailed description of the implemented emotional system is presented in (Chella and Barone 2008).

The Emotions Subsystem The emotions subsystem draws on the *basic emotions* theory. In particular, we refer to the six *basic emotions* proposed by (Ekman 1992): *Joy*, *Sad*, *Surprise*, *Anger*, *Fear* and *Disgust*. We also consider the neurobiological basis of emotions in order to develop an emotional learning system that gives the robot the ability to dynamically learn the emotional value for input stimuli. In particular, we refer to the computational model of emotional learning developed by (Balkenius and Morén 2001). The model is divided in two parts corresponding to the *amygdala* and the *orbitofrontal cortex*, where the model of amygdala learns the emotional response and the model of the orbitofrontal system inhibits the system output proportionally to the mismatches between the base system prediction and the actual received reinforcer.

The *AMY-OFC* block in Figure 1 implements the model with six outputs, one for each basic emotion. Primary reinforcers have been defined for each basic emotion; they are described in Tab. 1.

The input of the AMY-OFC block is the state of the robot

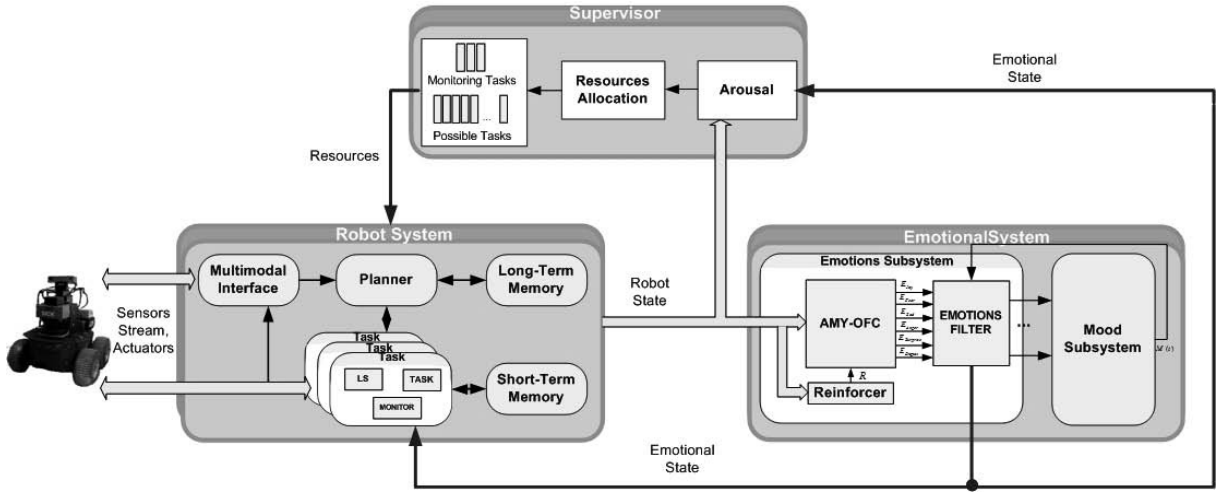

Figure 1: The robot architecture

Table 1: Primary Reinforcers

Emotion	Reinforcer
Joy	The attainment of a goal
Sad	Unsuccessful in achieving goal
Anger	Obstacles for the attainment of goals
Surprise	Unexpected stimulus
Disgust	Something revolting
Fear	Stimulus that hinds the goal of survival

that includes the environmental stimuli, the current actions, the achieved goals and so on. For each input stimulus that is repeatedly paired with a primary reinforcer, the corresponding emotion is progressively learned. Each emotion value ranges from 0 to 1.

As shown in Tab. 1, the primary reinforcers are related with the ongoing situations of the current robot tasks. In particular, they are automatically generated by the messages sent by the monitors of the corresponding robot tasks previously described. In facts, each task *monitor* generates a message describing the task situation. These messages are then employed by the emotions subsystem to generate the primary reinforcers listed in the reported Table.

For example, let us consider the *anger* emotion. Typically, the robot goal is to reach a fixed position in front of a plant to offer explanations about the plant itself. If someone obstructs the robot path then the robot encounters difficulty to reach the fixed position. This "obstacle for the attainment of its goal" situation is recognized by the monitor of the navigator task that sends a message to the emotional subsystem. This message is the primary reinforcer for the *anger* emotion. If the user repeatedly obstructs the robot path, the anger emotion increases to reach the stable value of 1. The presence of an obstacle is then associated with the anger emotion.

As another example, since the robot acts as a tour guide, one of its goal is to have many people following its tour. Therefore, an increasing number of people that follow the robot is a primary reinforcer for the joy emotion. If this reinforcer is paired, for example with the robot current action *tell an anecdote*, then the value of joy is learned for this action.

The association stimulus-emotion does not remain active forever. The robot gradually *unlearns* the link emotion-stimulus if it does not receive any more the primary reinforcement. Referring to the last example, if the robot, for the action *tell an anecdote*, does not detect an increasing of people, the joy emotion is unlearned for this action.

The outputs of the AMY-OFC block are the six values E_i corresponding to six basic emotions. An emotion is evaluated as *active* by the *EmotionFilter* block (see Figure 1) if its current value E_i is greater than the corresponding threshold Th_{E_i}. The thresholds take into account the robot mood state and also the robot *personality*. Different personalities have different emotional approaches. For example, the "choleric" type, one of the four personality types proposed by Hippocrates, is characterized by a quickly changed emotional reactivity. Instead, the "phlegmatic" type is characterized by a relatively emotionless approach (Meyer and Shack 1989). Mood also interferes with the emotions trigger: as an instance, when the robot is in a "good mood" it becomes joyful more easily.

The activation threshold for each emotion is defined as:

$$Th_{E_i} = Th_{PE_i} + \delta_i M(t) \quad (1)$$

where Th_{PE_i} is a fixed value for each emotion E_i according to robot personality P, $M(t)$ is the current mood and δ_i is a weight parameter.

A negative mood decreases the activation threshold for *negative* emotions (see Tab. 2), while it increases the activation threshold for *positive* emotions. Instead, a positive mood increases the activation threshold for negative emotions and it decreases the activation threshold for positive

Table 2: Positive and Negative Emotion

***positive* emotion**	***negative* emotion**
Joy	Sad
Surprise	Anger
	Disgust
	Fear

emotions.

The emotion with the highest value represents the current robot *emotional state*. All the *active* emotions influence the mood.

The Mood Subsystem We represent mood as an affective state with a positive or negative valence. The mood M at time $t+1$ is calculated as:

$$M(t+1) = f(M(t)) + \sum \alpha_i E_i(t+1) \qquad (2)$$

where f is a decay function, α_i are weights and $E_i(t+1)$ are the *active* emotions at time $t+1$. Tab. 2 reports the classification in *positive* and *negative* emotions. Occurrences of positive emotions contribute to induce a good mood while negative emotions induce bad mood.

The emotional state has a main role as a *context* in which tasks may be activated for the *dramatization* of the tour, and as a feedback for the *Supervisor* block in order to govern the allocation of computational resources of the tasks, as described below.

The Supervisor

The operation of the *supervisor* block is inspired to the model of *attention as a limited resource* introduced by Kahneman (Kahneman 1973). In particular, the supervisor plays the role of the allocator of the computational resources of the robot. According to Kahneman, the mobilization of effort to carry out a task is function of the demands of the task. The effort is reflected by the variation of the *arousal* level that governs the way by which the attention is allocated to different activities.

The model also assumes that continuous activities of perceptual monitoring absorb a certain amount of system capacity, labeled spare capacity. Spare capacity may be decreased to fulfill the increasing demands of ongoing tasks.

According to the Kahneman model, the supervisor modifies the computational resources allocated to the tasks depending on the current *emotional state*, i.e., the current *arousal*, and on the robot state. Therefore, the supervisor receives as input the robot emotional state that summarizes the ongoing of the robot operations, and it allocates computational resources to each task according to the received feedback (see Fig. 1). To guarantee the architecture generality the Supervisor does not have to know tasks specifications. Therefore, from the Supervisor point of view, computational resources are an abstract entity. In the current implementation we modeled computational resources as a number between 0 and 100. Each task receives the % of its own computational resources and distributes them among its subtasks by suitably varying their parameters. For example, the 3D reconstruction task may perform its computations by considering higher resolution images if the supervisor increases the computational resources for this task.

It should be noticed that a task includes all the related perceptual processes, i.e., the LSs required for its execution. The amount of computational resources that a task obtains is then propagated also to its subtasks, thus masking to the supervisor the functionality details of each task. In particular, when the allocated memory for a task is increased, the memory allocated for all of its subtasks is also increased.

Consider for example a task requiring the simultaneous processing of a certain number of incoming sensors streams, such as a navigator task which employs the 3D-reconstruction LS. In this case, the supervisor may decide to assign more computational resources, i.e., allocated memory, to the navigator task. If the resources allocated to the navigator subtasks were not increased too, the corresponding LSs resources could be decreased. The 3D-reconstructor LS will then decreases its allocated memory and correspondingly either the resolution or the frame rate, resulting in a decreasing of performances of the navigator task even if its resources were increased.

The supervisor also forces a minimal amount of resources (MSC, minimum spare capacity) to be reserved to safety tasks which guarantee the robot and the external environment.

The Robot at Work

The described robot architecture has been tested in *Robotanic*: an outdoor robotic touristic guide acting in the Botanical Garden of the University of Palermo. The robotic platform is a *MobileRobots* Pioneer 3-AT using differential drive and equipped with a laser scan range finder, a sonar array, a stereo camera and a GPS. The Botanical Garden of the University of Palermo is a public semi-structured outdoor environment, nearly 20000 sq.m. wide. It is characterized by several pathways sometimes bounded by short walls or plants. Furthermore, the roads are often covered by foliage or mud. A typical route takes 30 min to complete it (not including speaking).

The Botanical Garden environment is highly dynamic as visitors are all day walking inside the botanical garden, gardeners work alongside the plants and even workers trucks happen to pass by.

The robot has two main goals, namely: i) to guide tourists along the garden and ii) to ensure the safety of people, environment and of itself. During a route, the robot stops near exhibits of interest and it presents a description of them.

The Planner

The planner block retrieves the exact location of each requested exhibit from the Long Term Memory and it generates a navigation plan. Then, the planner activates a specific task, the navigator task, to execute the tour. During the tour, the robot offers explanations about plants or other objects of interest located near the fixed points.

The Long-Term Memory stores the environment map, which is represented as an undirected graph where each node

Table 3: Implemented tasks and subtasks

Task and subtasks		Input	Output	Parameters
Navigator	Controller	Velocities	Motor commands	Max Vel
	Local planner	Occup. grid, robot pose		Goal
Logic Sensor	Localizer	GPS, velocities	Robot pose	Covariance matrix
	Local mapper	Vision Points, odometry	Occup. grid	Dim, res.
	Local mapper	Laser Points, odometry	Occup. grid	Dim, res.
	Sensor fusion	Occup. grids	Occup. grid, discrepancy	Weights
	Ray cast	Laser readings	Laser Point Cloud	FOV
	3D-reconstructor	Stereo images	Vision Point	Res., frame rate
Emo-Dramatic Tour		Emotion, robot pose	Text unit, emotion	
Emotional Vocal Synth.		Emotion, text unit		

is a place meaningful for the robot. A place is meaningful if there is an exhibit; further nodes have been added to aid the navigation.

Each time the robot receives a new route made by a list of exhibits, the planner generates a list of nodes to be visited by performing a simple graph search over the map. The planner then generates a set of high level actions to complete the route. Each action corresponds to a task to be activated and its parameters. Next subsections detail the more significant tasks and subtasks implemented (Tab. 3).

The Navigator Task

In the current implementation, the navigator task is responsible of bringing the robot from its current position to a goal, which is provided in a global frame of reference, coherently with the frame provided by the localizer. The navigator acts as a glue between the controller, the local planner and the needed LSs. As soon as the robot is nearly enough to the goal it will report a success.

Controller The controller subtask hides the hardware interface with the motor board. It accepts translational and rotational velocities as input and it uses a PID controller to achieve the desired wheel velocities. Its monitor will report an error if there is some motor board failure. Some errors are recoverable, e.g. motor stalls due to PWM overpowering.

Local planner The local planner subtask employes the local map provided by the sensor fusion LS (see below) to generate motor commands coherent with the robot dynamics. It uses the goal and robot positions to generate sample trajectories according to Rapidly Exploring Random Trees (Kuffner Jr and LaValle 2000) algorithm. The trajectories are then weighted according to their success in keeping the robot far from obstacles and near the goal position; the best one is then picked. Each time the local map is updated, the subtask generates a new trajectory. In this way the navigator is able to handle incoming obstacles as soon as they are individuated. Its monitor will not report error, as we assume that a free path towards the goal is always available (no dead ends).

Logic Sensor Tasks

Local mapper The local mapper LS builds local metric maps based on laser ray casting and the 3D reconstruction of stereo images. The local metric map, either vision or laser based, is represented as a 2D occupancy grid. By accumulating data over time, we overcome the problems due to the limited field of view of the stereo camera. Local maps are bounded in space, as they rely on odometric noisy information. In the current implementation, we consider robot displacement of at most 3 meters, as experiments showed that our robotic odometry is unreliable for longer distances.

3D-reconstructor The 3D-reconstructor LS produces dense disparity images. The 3D-reconstructor LS output is a cloud of points referred to the local frame of reference located in the right camera. With the current stereo baseline, 3D reconstruction is performed using a maximum range threshold of 5 meters. The 3D points are then classified in floor points and obstacle points using a ground plane estimate provided by the standard *RANSAC* algorithm. The vision occupancy grid is updated by projecting down the obstacle points.

Ray cast The laser ray cast LS produces a cloud of points employed to update the laser occupancy grid.

Sensor fusion The sensor fusion LS performs a weighted sum of laser and vision local maps into a single local map (Broggi et al. 2006).

Localizer The localizer LS provides the robot with a pose estimate in a global frame of reference. We rely on GPS data to estimate the robot position. GPS gauges are easily handled by a standard EKF filter using velocity (translational and rotational) as control data and GPS data as observation. The pose provided by the localizer does not need to be very accurate, as other tasks (i.e. the navigator) will use it only as a first estimate of the robot progress towards the goal position. Its monitor will report error if the GPS stops working (hardware failure) or if the pose covariance will grow too much (localization failure).

Emo-Dramatic Tour Task

The Emo-Dramatic Tour task is based on the *drama theory* and it offers important cues to improve human-robot inter-

action and to let humans feel robots as "life-like" creatures.

The *drama* structure is characterized by: a protagonist that has a goal; the tension created by the obstacles the protagonist encounters; the antagonist that opposes the protagonist goal; the direction of the drama and the resolution at the end of the drama.

During the tour the robot puts up a *drama*. It is the protagonist of the drama and its goal is to offer a pleasant and playful tour to visitors. The changing of its emotional states creates a tension during the tour. The amount of information that the robot offers during the tour has been divided into elementary unit of text. Each unit of text has been tagged with the following information:

- Informative content: it indicates the content of the text (e.g. a botanical explanation about plants or anecdote about plants or robot "life")
- Place: it indicates the location of the garden where the information can be offered. Some texts are description of plants or exhibits and then can be exposed only when the robot is near their position. Other texts can be offered instead in any place.
- Length: it indicates the length of the text (short, medium, long).
- Flag: it indicates if the robot has just said the text.

The unit of text are dynamically selected and assembled during the tour according to a dramatic structure. The selection is driven by the robot *emotional state*. In facts, as previously anticipated, the emotional state plays a main role for the dramatization of the tour as a *context* in which robot tasks may be activated or not.

More in details, selection rules have been defined for each basic emotion E_i. Depending on robot *emotional state*, the appropriate selection rules are applied. For example, if the robot is joyful it is more loquacious and it wants to tell anecdotes. Then the selects, for the specified position (Place), long texts and texts with an informative content that indicate anecdotes. However, if the robot is anger it gives only brief and technique explanations about plants.

Emotional Vocal Synthesis Task

The Emotional Vocal Synthesis task refers to the emotional prosody speech study proposed by Schröder (Schröder 2004). Schröder formulates prosody rules that link emotional states with their effect on speech prosody. He suggests a representation of emotions for speech in a three dimensional space: *activation* A, *evaluation* E, *power* P. He defines some emotional prosody rules that allow mapping speech prosody associated with an emotion in this 3-dimensional space.

We mapped the six *basic emotions* proposed by Ekman and the absence of emotion (neutral) in this space. For five emotions categories, neutral, sadness, anger, fear and joy, we used the values of A, E and P defined by Schröder and colleagues (Schröder et al. 2001). For the other emotions we experimentally defined the A, E and P values.

The Emotional Vocal Synthesis task receives in input the emotion E_i from the Emo-Dramatic Tour task and it calculates the corresponding point $w(A_{E_i}, E_{E_i}, P_{E_i})$. The rules defined by Schröder allow mapping this point with the speech prosody parameter (pitch, intensity, duration and so on). These values and the unit of text are employed to create a suitable document that contains the information for the emotional vocal synthesis. The document is then sent to the MARY Text-To-Speech System (Schröder and Trouvain 2003) for the emotional vocalization of the excerpt performed by the multimodal interface of the robot architecture.

Figure 2: Tour snapshots

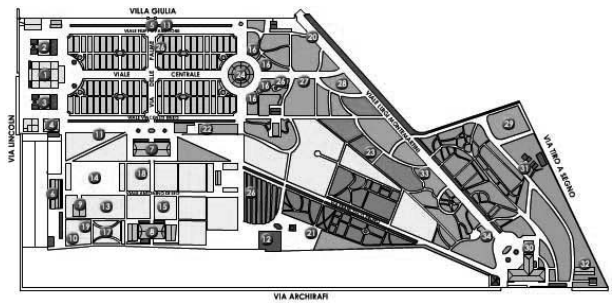

Figure 3: Botanical garden map

Experimental Results

The live test-bed for the proposed system was during a crowded international scientific conference held in the Botanical Garden in Sept. 2007. The robot performed several tours, covering more than 3 km in a day. Figure 2 shows some snapshots taken during the tour.

Odometric errors and sensor uncertainty underpinned any effort to build a map of the whole environment, leading us to choose a mapless navigation approach. We used a tourist map (Figure 3) to figure out where to place exhibit nodes and navigation aid nodes.

The latter were introduced to avoid the robot being trapped in local minima. Each node location was computed by averaging a set of GPS data taken during one whole day, to take into account satellite variations.

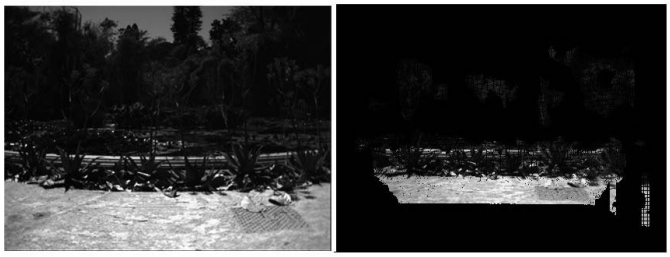

Figure 4: Basin place (left) and its 3D reconstruction (right)

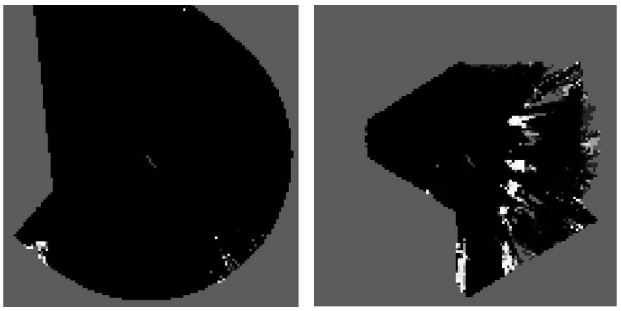

Figure 5: Laser-based (left) and vision-based (right) occupancy grid near the basin.

In the current implementation we have no exhibit recognition module, so the robot stops at fixed positions in order to illustrate the exhibits. This approach relies heavily on the localizer pose estimate. Future works will introduce some object recognition modules in order to recognize and track meaningful features.

During several routes, the laser was the main source of sensor data: it had been used for more than 80% of the time, while the 3D-reconstructor task became dominant only near the garden basin, as explained below. The robot reported several problems trying to get near an exhibit, with an average pose error of about 3 ms an and average angular error of about 13 degs respect to the desired goal.

The satellites number tracked by the GPS was 5 on average, dropping down to 3 when under heavy foliage or rising to 9 when in open space. We had no ground truth to estimate the pose error, but experiments showed that the error is not higher than 5 ms. This was an acceptable error, as both the navigator and the local planner had no problems driving the robot towards each goal.

One of the most critical sections of the Botanical Garden is near a basin placed in the middle of a large place and surrounded by short plants and flowers (Figure 4). Here, the *Supervisor* block covers a main role.

In facts, the laser was unable to detect the plants as obstacles, leading to an erroneous free space in front of the robot. This caused an increasing discrepancy between the laser local map and the vision-based local map (Figure 5).

This discrepancy value is a primary reinforcer to the fear emotion that leaded to a high arousal value. Figure 6 shows

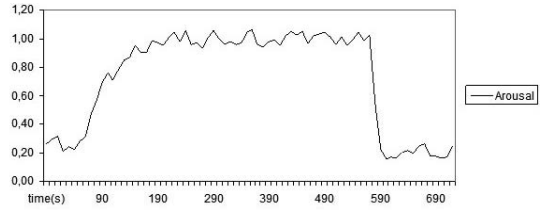

Figure 6: Arousal vs. time

Table 4: Percentage of computational resources allocated by the supervisor. MSC is the Minimum Spare Capacity.

Task	**Resources**	
	$t = 0s$	$t = 200s$
Planner	5	5
Navigator	60	90
Emo-Dramatic Tour	15	0
Emotional Vocal Synthesis	15	0
MSC	5	5

the arousal values vs. time, while the robot was negotiating the basin place. As a consequence, the supervisor allocated more resources to the navigator task. Table 4 shows the corresponding resources variation per task.

The navigator assigns more resources to the local map LS that contains more obstacles and its coupled LS. As the 3D-reconstructor LS was able to map more obstacles, its resources were increased. Table 5 shows the corresponding parameters variation (priority and image resolution for the 3D-reconstructor). Priority parameter ranges from -20 to 20. As soon as the robot managed to get around the basin, the laser resumed consistency and was used again as the main navigation sensor.

Conclusions

We described a brain inspired architecture which is flexible with respect to different environmental situations and robot tasks and which allows the robot to be perceived as a social agent. The proposed system has been tested in a real application proving its fault tolerance and social abilities.

The described architecture has points in common with the RCS architecture proposed during the years by Albus and

Table 5: Parameters varied by the navigator

Subtask	**Parameters**	
	$t = 0s$	$t = 200s$
Controller	0	0
Local Planner	5	5
Vision Local Mapper	-5	10
Laser Local Mapper	10	-5
Sensor Fusion	0	0
3D Reconstruction	-5, 160×120	10, 640×480
Ray Cast	10	-5

collaborators (Albus and Meystel 2001; Albus and Barbera 2004). In facts, our task structure has some similarity with the RCS node and also the tasks and subtasks decomposition reported in Tab. 3 have similarities with the task decomposition hierarchy typical of the RCS implementations.

However, the main difference between the two architectures is that RCS is a hierarchical architecture while our tasks are computer threads with no a priori defined activation. The activations of RCS nodes are defined by the architecture designer according to a fixed hierarchy of levels. Instead, in the proposed architecture, the *Supervisor* block dynamically allocates computational resources, i.e., the task activation priorities and the allocated memory, according the ongoing of the robot mission. In facts, as previously described, our emotional system has the role to send feedback about the robot mission to the supervisor, by means of the *emotional state*. According to received feedback, the supervisor may change the tasks computational resources.

In this way, when the robot operations are no more satisfactory, e.g., in the case of the robot operations near the garden basin, the robot emotional state lets the supervisor to change the task parameters in order to affect the activation priority and the memory allocated for the tasks and subtasks. In this way, our architecture is able to dynamically reconfigure itself in runtime.

Another main benefit for the introduction of the emotional mechanism is the increasing of the acceptance of the robot as an "entity" by people, especially by children, that perceive the robot as the owner of a personality. First experiments show that peoples find pleasing and entertaining a robot able to be angry and joyful. Future works will concern the quantitative measure of human robot interactions (Goodrich and Schultz 2007; Dautenhahn 2007) by means of the collection of visitors opinions gathered through interviews and questionnaires and the analysis of the data with the supervision of psychologists to verify the acceptability of the robot and the visitors satisfaction.

References

Albus, J., and Barbera, A. 2004. RCS: A cognitive architecture for intelligent multi-agent systems. In *Proc. 5th IFAC/EURON Symposium on Intelligent Autonomous Vehichles IAV 2004*.

Albus, J., and Meystel, A. 2001. *Engineering of Mind*. Wiley-Interscience.

Balkenius, C., and Morén, J. 2001. Emotional Learning: a Computational Model of the Amygdala. *Cybernetics and Systems* 32(6):611–636.

Breazeal, C. 2002. *Designing Sociable Robots*. Cambridge, MA.: MIT Press, Bradford Books.

Broggi, A.; Cattani, S.; Porta, P.; and Zani, P. 2006. A laserscanner-vision fusion system implemented on the TerraMax autonomous vehicle. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 111 – 116.

Canamero, L. 1997. Modeling motivations and emotions as a basis for intelligent behavior. In *Proc. of the first international conference on Autonomous agents*, 148 – 155.

Chella, A., and Barone, R. 2008. Panormo: An Emo-Dramatic Tour Guide. *AAAI Spring Symposium on Emotion, Personality and Social Behaviour* 10 – 16.

Chella, A.; Liotta, M.; and Macaluso, I. 2007. CiceRobot: a cognitive robot for interactive museum tours. *Industrial Robot: An International Journal* 34(6):503–511.

Dautenhahn, K. 2007. Socially intelligent robots: dimensions of human–robot interaction. *Phil. Trans. R. Soc. B* 362:679 – 704.

Ekman, P. 1992. Are there basic emotions. *Psychological Review* 99(3):550 – 553.

Goodrich, M., and Schultz, A. 2007. Human-Robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction* 1(3):203–275.

Ishiguro, H., and Nishio, S. 2007. Building artificial humans to understand humans. *J. Artif. Organs* 10:133 – 142.

Kahneman, D. 1973. *Attention and effort*. Englewood Cliffs, NJ: Prentice-Hall.

Kelley, A. 2004. Neurochemical networks encoding emotion and motivation: an evolutionary perspective. In Fellous, J.-M., and Arbib, M., eds., *Who Needs Emotions: The Brain Meets the Robot*. Oxford University Press, USA. 29 – 77.

Kuffner Jr, J., and LaValle, S. 2000. RRT-connect: An efficient approach to single-query path planning. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA 2000)*. IEEE Press.

Lisetti, C.; Brown, S.; Alvarez, K.; and Marpaung, A. 2004. A social informatics approach to human-robot interaction with a service social robot. *IEEE Transactions on Systems, Man and Cybernetics, Part C* 34(2):195–209.

Meyer, G., and Shack, J. 1989. Structural convergence of mood and personality: evidence for old and new directions. *Journal of personality and social psychology* 57(4):691–706.

Ortony, A.; Norman, D.; and Revelle, W. 2005. Affect and proto-affect in effective functioning. In Fellous, J.-M., and Arbib, M., eds., *Who Needs Emotions: The Brain Meets the Robot*. Oxford University Press, USA. 173 – 202.

Rolls, E. 1990. A theory of emotion, and its application to understanding the neural basis of emotion. *Cognition & Emotion* 4(3):161–190.

Schröder, M., and Trouvain, J. 2003. The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching. *International Journal of Speech Technology* 6(4):365–377.

Schröder, M.; Cowie, R.; Douglas-Cowie, E.; Westerdijk, M.; and Gielen, S. 2001. Acoustic correlates of emotion dimensions in view of speech synthesis. In *Seventh European Conference on Speech Communication and Technology*. ISCA.

Schröder, M. 2004. Dimensional emotion representation as a basis for speech synthesis with non-extreme emotions. In *Proc. Workshop on Affective Dialogue Systems*, 209 – 220. Springer.