*Cognitive Science Quarterly (2002) **1**, ..-..*

EMIB — Computational Architecture Based on Emotion and Motivation for Intentional Selection and Configuration of Behaviour-Producing Modules [1]

François Michaud
Université de Sherbrooke [2]

Over the years, intelligence has been the subject of studies by many different fields, contributing to reveal some of its mysteries. Computational architectures try to exploit in various ways these aspects for designing artificial systems. The biggest challenge is to integrate more and more properties and principles associated with intelligence, combining their advantages to minimise their limitations. With this objective in mind, we propose a computational architecture that tries to synthesise concepts about intelligence, while making sure that the underlying principles of these concepts, such as emergence, are preserved. The architecture is based on intentional selection and configuration of behaviour-producing modules. Behaviour-producing modules are used as basic control components that are selected and modified dynamically according to the intentions of the system. These intentions are influenced by the situation perceived, the need to accomplish specific goals over time, and knowledge innate or acquired about the world. Motivational and emotional variables are used to monitor the goals and the overall states of the system. The EMIB architecture is applied in designing intelligent autonomous mobile robots, as illustrated in the three experimental cases presented in this paper.

Keywords: Intelligence, Deliberative Architecture, Behaviour-Based Architecture, Emergence, Autonomous Robots.

---

[1] This research is supported financially by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Research Chair (CRC) program, the Canadian Foundation for Innovation (CFI) and the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR) of Québec.

[2] Department of Electrical Engineering and Computer Engineering, Université de Sherbrooke, 2500 boul. Université, Sherbrooke (Québec CANADA) J1K 2R1, tel.: (819) 821-8000 x2107, fax: (819) 821-7937, email: francois.michaud@courrier.usherb.ca

Received 25.03.02

Introduction

Intelligence can be addressed from various perspectives, ranging from the manifestations of intelligent capabilities to the underlying mechanisms used to generate them. Artificial intelligence, control, psychology, ethology, linguistics, biology, neuroscience and many other research fields try to gather insights on the principles, methodologies and concepts associated with intelligence. Research in these fields gives rise to theories and working principles that are translated into computational architectures, identifying the fundamental building blocks to explain intelligent decision-making processes or to reproduce them on artificial systems (also called agents).

Reviewing all of the particularities of computational architectures used in artificial intelligence and robotics is outside the scope of this paper (Arkin (1998) and Murphy (2000) present excellent overviews). However, it is possible to outline three general decomposition principles currently found in architectural methodologies:

- Hierarchical decomposition: these architectures are organised in a hierarchy of modules layered according to the *Principle of Increasing Precision with Decreasing Intelligence* (Saridis, 1983). Saridis, from the field of intelligent control, first proposed a three-level architecture (i.e., Execution Level, Coordination Level and Organisation Level), each level interacting only with the subsequent one. The intended applications are for machines operating in well-defined operating conditions.
- Functional decomposition: these architectures follow a functional decomposition of intelligent processes, by having for instance a sensory processing module, a modelling module, a planning module, a value judgement module and an execution module, as proposed by Albus (1991). Each module exchanges information between each other based on their functionality. Similar strategies can be found in multi-agent research (Werner, 1992). Such specialised modules allow complex operations to be performed (for instance, modelling the world allows an agent to plan ahead and anticipate actions to take over time), but implies strong interdependencies between the decision-making modules.
- Behaviour-based decomposition: the idea exploited in these architectures is to have multiple concurrent task-achieving processes, deriving actions from sensed conditions or internal states, and to fuse these actions using an arbitration scheme in order to determine the output of the system. By having such modules interact directly with the environment and without having dependencies between each other, it is said that a functionality emerges from these interactions that are neither a property of the agent or the environment in isolation, but rather a result of the interplay between them (Arkin, 1998). Subsumption arbitration (Brooks, 1986) and Motor Schemas (Arkin, 1989) are examples of such architectures. The approach has shown to be especially effective for mobile robots, allowing them to adapt to the dynamics of real-world environments without operating upon abstract representations of reality (Brooks, 1991).

Each of these paradigms offers interesting but different insights about intelligence, that are incomplete rather than being incorrect. Many recent computational architectures try to combine the advantages of these paradigms, especially the responsiveness, robustness and flexibility of behaviour-based approach with the use of abstract representational knowledge for reasoning and planning about the world (Arkin, 1998) or for managing multiple conflicting goals. For instance, AuRA uses a planner on top of behavioural modules (Arkin, 1998; Murphy, 2001), or 3T uses behavioural modules in the execution layer of a three-level hierarchical architecture (Bonasso, Firby, Gat, Kortenkamp, Miller et al., 1995). The basic principle for making the coupling is to dynamically reconfigure the behavioural modules according to reasoning done based on available world knowledge (Arkin, 1998).

However, in designing these hybrid architectures, it is essential that the fundamental aspects of each of their decomposition principles be preserved. One difficulty is to conserve and exploit emergent properties at all levels of decision, and not only at the behavioural level. We use the term emergence here to refer to the holistic capability arising from a collection of components interacting through the interactions the agent has with its environment (and not with themselves within the agent) (Arkin, 1998). Combining abstract reasoning processes with behavioural modules, must still allow emergence to take place at the behavioural level, and should also be present at the higher decision levels. For instance, by only using one module (like a planner) to reconfigure behavioural modules over time, a strong dependency is created on the accuracy and the adequacy of the higher decision module. However, using concurrent processes that monitor different conditions independently from one another to reconfigure behavioural modules, emergence can arise at higher abstraction levels too. This property is especially important for agents that have to plan, reason and act in unpredictable and changing environments, like in real world settings. Another use of emergence would be in deriving knowledge about the world. Most deliberative approaches derive knowledge for reasoning about the world from sensor inputs and actions taken by the agent. This results in high state space representation of the world, and does not take into consideration the context in which these sensations / actions are taken. Since behavioural modules are the low-level control blocks and that their use are driven by what emerges from the interactions with the environment, they can also serve as an abstract representation of what is experienced in the world. The purpose associated with each behavioural module can then be exploited for reasoning, grounding intentions to what the agent is experiencing in the world.

Since 1995, we have been experimenting with a three-level computational architecture that attempts to combine concepts like reactivity, deliberation and motivation, while preserving emergence at all abstract decision levels. The architecture named EMIB (for Emotion and Motivation for Intentional selection and configuration of Behaviour-producing modules) is based on the idea of intentionally selecting and configuring behaviour-producing modules. Behaviour-producing modules are used as basic control components that are selected and modified according to the intentions (as derived by the higher decision levels) of the agent. These intentions are influenced by the situation perceived, the need to accomplish specific goals over time, and knowledge innate or acquired about the world. Motivational and emotional variables are used in the architecture to monitor the goals and the overall states of the agent. One key influence in the architecture is the ability to observe how behavioural modules are used over time, gaining insights on what emerges out of the interactions the agent has with its environment.

This paper first explains the rationale behind the design choices made for the EMIB computational architecture, and then illustrates its use in three experimental cases. The objective is to design an architecture that is application- and implementation-independent, and that can exploit the appropriate decision-making mechanisms according to the agent's capabilities and purposes. Note that the architecture is not designed to model human cognitive processes, but can be inspired from them.

EMIB computational architecture

The architecture, shown in Figure 1, is made of three levels: the Behavioural Level, the Recommendation Level and the Motivational Level. The Behavioural Level of the architecture is made of behaviour-producing modules, connecting sensory information to actions. The Recommendation Level is responsible for changing the selection of behaviours or to reconfigure them to make the system behave appropriately according to its goals and the situations it encounters in the world. The Motivational Level monitors the goals and the overall states of the agent.

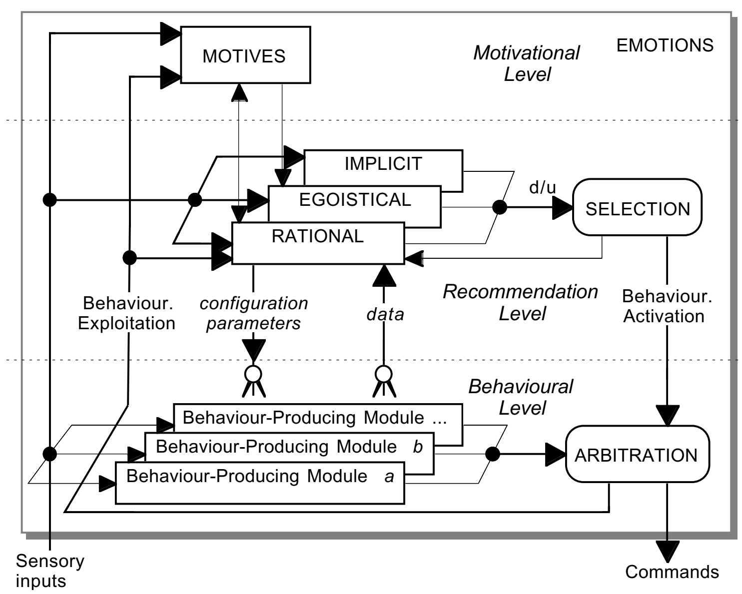


Figure 1: EMIB computational architecture.

Behavioural Level

Behaviour-producing modules allow the agent to respond in particular ways to situations encountered in the environment. These modules all run in parallel and their resulting commands are combined using an arbitration mechanism to generate the control actions of the robot. Different methodologies can be used to implement the behaviour-producing modules and the arbitration mechanism (like Subsumption (Brooks, 1986; Mataric, 1992), Motor schemas (Arkin, 1998), Fuzzy Logic (Saffiotti, 1997), etc.). Emergent computing occurs here by having modules issuing actions simultaneously on the same actuator, but based on different sensed or internal conditions.

Recommendation Level

To still allow emergent computation, the Recommendation Level uses three types of modules that formulate, concurrently, behavioural recommendations based on different monitoring conditions. The *Implicit* module recommends behaviour-producing modules by default or driven by conditions derived from sensory inputs. The *Egoistical* module selects behavioural modules according to the agent's 'needs' or goals, which are influenced by the *Motives* module described in the next subsection. In this module, goals can be prioritised for instance according to Maslow's Hierarchy of Needs Theory (Maslow, 1954), from physiological, security, to social and accomplishment needs. The *Rational* module is for behavioural recommendations based on innate or acquired knowledge about the world, to plan or to prepare the use of behaviour-producing modules exploiting such knowledge. In the case of a mobile robot, this can involve navigation using a map (learned or known a priori), communication with other agents (artificial or human), or any methods used for reasoning that can be useful for the agent. Note that knowledge about the world can also involve internal states of the agents, like motives, behavioural recommendations or data provided by the behaviour-producing modules (like behaviours used to recognise specific objects in the world (Arkin, 1998)). Rational recommendations can also involve reconfiguration of behaviour-producing modules or generation of virtual inputs, via the *configuration parameters* link.

The use of three types of recommendation modules has been influenced by the hypothesis that human behaviour is subject to three influences: the environment, the needs of the individual and its knowledge (Dolan & Lamoureux, 1990). The importance of each of these influences explains why manifested behaviour can be more reactive, egoistic or rational.

These three modules can be compared to three types of 'behaviour-recommending' modules responsible for making decisions (or recommendations) about which behaviour-producing modules must be activated or inhibited, according to what the robot wants to do and what situations it is experiencing in the world. In order to do this by concurrent processes, a common representation for making these recommendations is necessary. A recommendation for a behaviour-producing module can be of two types: a *desirability* (*d*) value (ranging from 0 to 100%), requesting that a behaviour-producing module be activated; an *undesirability* (*u*) value (ranging from 0 to 100%) indicating that a behaviour-producing module must not be activated.

Each recommendation module can derive desirability and undesirability values for each behaviour-producing module. Using these parameters, the *Selection* module determines the activation of behaviour-producing modules, i.e., by deriving the *Behaviour.Activation* parameters. Different methodologies can be used to implement this module, but the underlying idea here is to activate behavioural modules that are more desirable than undesirable. This is inspired from the hedonic axiom which indicates that the organisms direct their behaviours to minimise aversions and maximise desirable outcomes (Beck, 1983). The *d/u* parameters make it possible to prevent possible conflicts when recommending behaviour-producing modules, which may occur from the parallel and independent evaluation of these three types of recommendation modules, or from different rules in one recommendation module. With Subsumption arbitration for instance, it would allow a behaviour-producing module of lower priority to get exploited by inhibiting the higher priority modules.

Motivational Level

When operating in unpredictable and partially observable environments, an autonomous agent must examine the evolution of its general states, and try to capture what emerges from the interaction dynamics with its environment. Temporal integration of different types of observations is an important concept for doing so (McCarthy, 1995; Smithers, 1994). Works on motivational systems (Maes, 1991; Blumberg, Todd & Maes, 1996; Breazeal, 1998) have shown that a good balance between planning and reactivity for goal-management can be achieved using internal variables that get activated or inhibited by different factors. In that regard, motivations (McFarland & Bösser, 1993; Maes, 1991; Parker, 1998) and emotions (Simon 1967; Ortony, Clore & Collins, 1988; Oatley & Johnson-Laird, 1987; Albus, 1991; Breazeal, 1998) are concepts that are gaining importance in the design of autonomous agents. They reveal to be useful in making an efficient connection between adapting to the contingencies of the world and making the agent accomplish its goals. That explains why the third level of the architecture, the Motivation Level, is made of the *Motives* module and the *Emotions* module.

***Motives* module**. The term ‘motive’ refers to something that prompts an agent to act in a certain way. Similarly, the *Motives* module is used to represent the agent's goals, making it decide how to behave in the world. Motives can be influenced by the environment (from sensed conditions), the intentions of the robot (derived from behavioural recommendations), knowledge about the world (managed by the *Rational* module) and by observing the effective use of the behaviour-producing modules. Motives are used to influence the recommendations of behaviour-producing modules by the *Egoistical* module, and can also characterise special states in the *Rational* module (for example, motives experienced at particular locations can be memorised for future reference).

Even though different representations can be used to implement the *Motives* module, we have been experimenting with one that tries to integrate temporal reasoning with a symbolic representation. Each motive is associated with a particular goal (that can be accomplished using one or more behaviour-producing modules) of the agent. As shown in Figure 2, a motive *m* is characterised by an energy level *E* and a mapping function *M* that are used to determine its activation level *A* according to the formula: $A_m=M(E_m)$. The energy level and the activation level of a motive range between 0 and 100%. The energy level can be influenced by various factors: sensory inputs, exploitation of behaviour-producing modules associated with the motive, activation of other motives, *Rational* influences, emotions, and time. The energy level is computed by adding the influence of *n* factors affecting the motive, weighted by *w*. This can also be interpreted as having different increment or decrement values *w* associated with particular events *j*. For factors that occur for long period of time and that must not influence the motive for the entire period, a habituation function can be used (Staddon, 1993) to modulate its influence. The habituation strength is used to modulate the weighted input that influences the energy level of the motive. With a higher habituation rate, the same weighted input has less influence over a consecutive period of time. Finally, mapping from *E* to *A* can be determined according to different methods: directly transposed (*E*=*A*); determined using thresholds (*if E > threshold then A = x, …*); or triggered according to the energy level of other motives. The activation level of a motive can then influence other motives or be used by the other modules of the architecture.

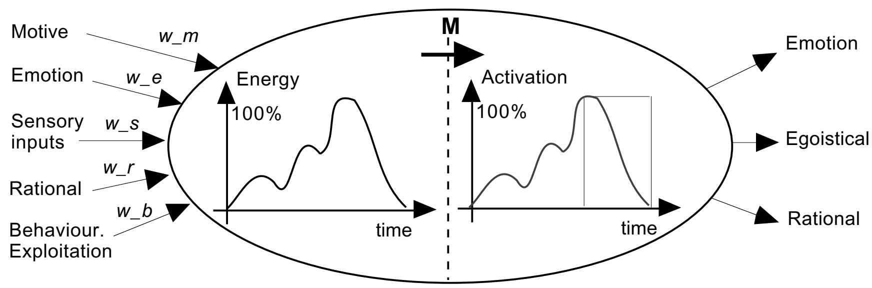


Figure 2: Schematic representation of a motive.

***Emotions* module.** The concept of artificial emotion is increasingly used in designing autonomous robotic agents (Velásquez, 1998; Breazeal, 1998). In fact, psychological evidences suggest that emotion can serve three important roles (Michaud, Pirjanian, Audet & Létourneau, 2000).

— Emotion to adapt to limitations. Emotion plays a role in determining control precedence between different behavioural modes, coordinating plans and multiple goals to adapt to the contingencies of the world (under constraints of time and other limited resources), especially in imperfectly predictable environments (Frijda, 1987; Oatley & Johnson-Laird, 1987; Plutchik, 1980).

— Emotion for managing social behaviour. Plutchik (1980) interestingly points out that emotions are in direct association with four universal problems of adaptation, which are: hierarchy (*Anger/Fear*), territoriality (*Exploration/Surprise*), identity (*Acceptance/Rejection*) and temporality (*Joy /*

Sadness). Plutchik's theory also suggests the possibility that emotions "*are functional adaptations for establishing a kind of social equilibrium. This would imply that emotions enter into every social transaction and help to establish a balance of opposing forces. These balances are always temporary and frequently change as we move through life from one conflict to another*" (Plutchik, 1980).

— Emotion for interpersonal communication. In order for emotions to regulate behaviour in social interaction, emotion also has a communicative role. Ethologists believe that emotional expression has a communicative function and acts as releasers for the coordination of social behaviour. Emotional expression promotes individual isolation (as it may be necessary in defending something) or to promote group action (as different social circumstances might require). The role of expression in emotion can be seen from three different views: the situation is evaluated by emotion that lead to an expression; expression may be a reaction to the situation that also produces the emotion; the expression may affect the emotion rather than the other way around (Mook, 1987). Emotion then serves a dual purpose: it is a communication act and it is a sensed state.

From an engineering point of view, autonomous robots would surely benefit from having mechanisms that play a similar role. Different mechanisms for implementing artificial emotions can surely be designed according to properties associated with the decision-making approach used to control the agent. One possibility is to use internal variables that have the same purposes as emotion. However, an internal variable can be named as an emotion, without playing all of the three roles associated with emotion. To avoid potential misuse of the term 'emotion', our goal is to derive an emotional model that is generic and task-independent, i.e., we would like the mechanism that derives emotional states to be the same whatever the goals pursued by the robot.

The model of emotion we are studying differs from other emotional representation in that the behaviour of the system is not oriented toward satisfying particular emotional states like in Breazeal's work (Breazeal, 1998). Artificial emotions are used to monitor how goals get satisfied or not, in a way similar to the concept of B-Brain introduced by Minsky (2002). Our model is two-dimensional and bipolar, with four emotions: *Joy/Sadness* and *Anger/Fear*, each defined from 0 to 100%. *Joy* and *Anger* are positive emotions, while *Sadness* and *Fear* and negative emotions. In our model, the energy level of motives is used as an abstraction of the progression toward the accomplishment of the goals associated with activated motives: *Joy* monitors a decrease in the energy level, indicating the accomplishment of the goal associated with the motive; *Sadness* monitors an increase in the energy level, indicating difficult progress in the accomplishment of the goal associated with the motive; *Anger* monitors oscillations in the energy level, indicating difficult progress in the accomplishment of the goal associated with the motive; and *Fear* monitors constant energy level, indicating no progress in the accomplishment of the goal associated with the motive. Monitoring the energy level of motives makes the approach generic, since the emotions can be influenced by (an emerge from) different contexts (i.e., goals) according to the motives activated and their priority, attributed following the guidelines of Maslow's *Hierarchy of Needs Theory* (Maslow, 1954).

As shown in Figure 1, the emotional capability in the EMIB computational architecture is incorporated as a global background state, allowing emotions to influence and to be influenced by all of the architecture's modules. This is related to research conducted by Oatley and Johnson-Laird (1987) indicating that emotions provide a biological solution to certain problems of transition between plans in systems with multiple goals and in unpredictable environments, by maintaining these transitions and by communicating them to ourselves and to others. For instance, emotions can be used to: change some of the parameters of behaviour-producing modules, adapting the way the agent respond to stimulus or express emotional states; influence the implicit, egoistic or rational influences of the recommendation modules, locally in each one or globally by affecting their importance in the *Selection* module; associate agent’s states with particular event, memorised in the *Rational* module; affect the goals of the agent via its motives.

Behaviour.Exploitation parameters

Behavioural exploitation refers to the observation of the effective use of behaviour-producing modules. *Behaviour.Exploitation* parameters differ from *Behaviour.Activation* parameters in the sense that the agent intends to use an active behaviour-producing module by allowing it to participate to the control of the agent, and it is said to be exploited only if it is actually used to control the agent (by reacting to the sensations using the control rules of the behaviour and going through to the arbitration mechanism at the Behavioural Level). An active behaviour is not exploited when it is not releasing commands that affect the actions of the agent.

We believe that observing the exploitation of behaviours over time is a very important source of information about the emerging functionality that comes from the behaviour-producing modules and the recommendation modules. This is explained by the fact that behaviour exploitation combines both a representation of the environment (i.e., the sensory information used by the behaviour) and of the control policy (since the exploitation of a behaviour depends on its purpose, behaviour arbitration mechanism and conditions associated with its activation). *Behaviour.Exploitation* parameters provide important feedback about the interactions the agent is having in its operating environment, interactions that emerge from the computation done in the three abstraction levels of the architecture. In the EMIB architecture, such feedback can be used to influence motives or to derive knowledge about the world as the agent experiences it.

Experimental cases

Since intelligence depends on the sensing, acting and processing capabilities of the agent, and because of the variety of aspects addressed by the architecture proposed, the experiments conducted are oriented toward validating different subsets of the properties of the EMIB architecture. The choice of EMIB’s decision modules and the mechanisms used by them are done according to the robot's capabilities and purpose. Experiments reported here were made in a simulated environment for mobile robots (Michaud, Lachiver & LeDinh, 1996; Michaud, Lachiver & LeDinh, 2001) to study the properties of the architecture without the influence of emotions, with a mobile robot that has to learn in non-stationary environment from observing the use of behaviour-producing modules (Michaud & Mataric, 1999), and with a mobile robot interacting with a human using light signals (Michaud & Vu, 1999). Detailed results for each of these experimental cases are not provided in this paper, and more information can be found in the referenced publications. The current paper reports general observations on the properties of the EMIB architecture, especially its ability to combine various mechanisms for intelligent decision-making, while preserving and exploiting emergence in all of its decision-making levels.

Monitoring incorrect behaviour and topological representation

Figure 3 shows an implementation of EMIB to make a simulated robot efficiently reach targets and recharge itself when needed. The robot does not have any a priori knowledge about the environment, and has limited memory to acquire information that can be helpful in its task. Fuzzy logic is used to implement behaviour-producing modules and for recommending them. This allows the blending of commands and of behavioural recommendations. The *Rational* module uses a topological graph to construct an internal representation of the environment, and the *Motives* module manages and monitors the goals of the robot.

Figure 4 presents one result obtained, showing how interactions between the agent and its environment can be monitored using the *Behaviour.Exploitation* link and motives. The simulated robot starts from the centre of the environment and goes directly into the lower right corner. At this corner, a conflict occurs between the behaviour-producing modules for avoiding very close obstacle (called *Emergency*), for moving away from obstacle in front of the robot (called *Avoid*) and for following the wall (called *Align*), and the robot is not able to move away from this location.

However, this situation can be monitored by looking at the exploitation of these behaviour-producing modules over time (using the *Behaviour.Exploitation* link). In this case, the simultaneous constant exploitation of *Emergency* and *Avoid* (between cycle 10 and 40) is a sign that something is not working properly for the robot: normally these behaviours should be used for short periods of time if they are successful in making the robot avoid a collision. This condition is monitored by a motive called *Distress* from which a behaviour to make the robot back up is recommended by the *Egoistical* module. In this case, *Distress* is used to monitor the emergent situation caused by the unsuccessful use of behavioural modules for avoiding obstacles and following walls. The robot is then able to move away from the corner and to continue exploring the environment.

Managed by the *Rational* module, a topological graph is learned and used by the robot as a representation of the environment. At its lowest level, the graph is constructed from topological states identified by the *Landmark* module. This procedure is similar to the work of Mataric (1992), but using a

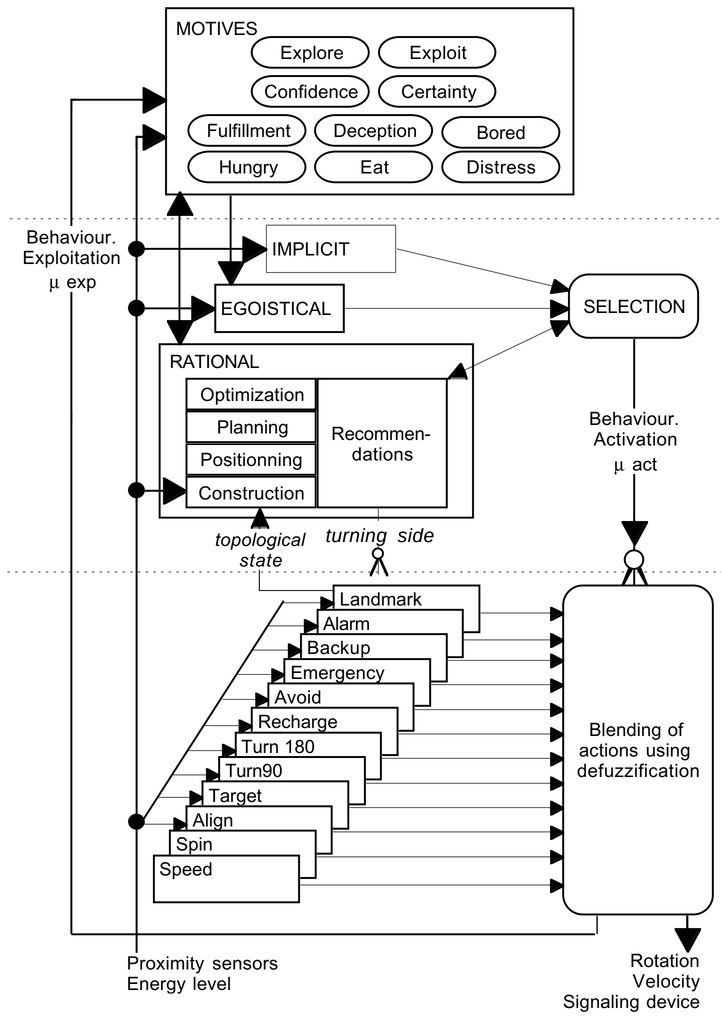


Figure 3: Implementation used for the experiments with a simulated mobile robot.

different identification behaviour. If activated, this behavioural module examines sensations coming from the front, the back and the two sides of the agent. The presence or the absence of obstacles at a certain distance in these four directions is used to infer one of 16 possible topological states. Based on these states, nodes are created to represent particular locations in the environment. Information can be stored in these nodes, like the number of visits to the node, the presence of a goal like a target or a charging station, the occurrence of a motive, etc. Adjacent nodes have links between them, making it possible to position the agent in the topological graph and to plan paths to go to particular locations. However, since only proximity sensors are used to construct this graph, optimization is an essential process. Because the similarities between nodes are evaluated based on a sequence of similar nodes and not by factors uniquely identifying each possible landmark in the environment, the graph can be composed of parallel branches or multiple nodes for the same site. The agent also has limited memory space, and so only the useful nodes are to be kept in the graph. A node is considered useful when it has been visited more than once, it is at the start of a branch or it is part of a path from a node referring to a charging station (located at the lower left corner) toward a node referring to a target (the regions delimited by circles). Figure 5 shows the topological graph before and after optimization. After optimization, the graph is reduced by approximately 50%. The paths following the boundaries of the environment are kept without being explicitly specified in the optimization procedure, and thus the useful paths emerge from the interactions the agent has with the environment (based on its behavioural capabilities) and its ability to use its topological representation.

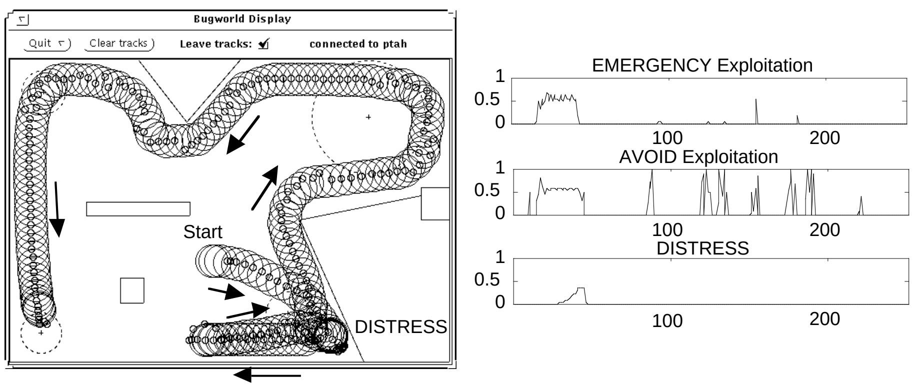
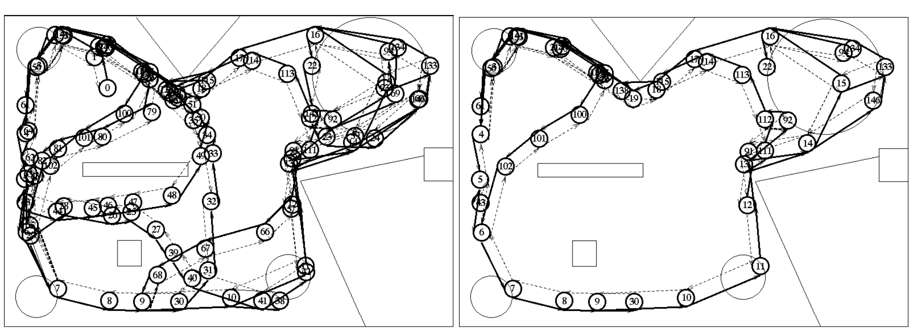


Figure 4: Example of monitoring incorrect behaviour using *Behaviour.Exploitation*.



a) Before optimization

b) After optimization

Figure 5: Topological graph before and after optimization.

Learning from the history of behaviour use

Since behaviour-producing modules are responsible for managing the direct interactions between the agent and its environment, observing their exploitation can be used to model these interactions. By knowing the purpose of each of the behaviour-producing modules and by observing their exploitations, the agent can autonomously evaluate and model the type of interactions it is having with its environment. This ability has been validated in the context of learning an “interaction model” based on the robot's history of behavioural exploitation, i.e., by representing the sequence of behaviour-producing modules exploited to characterise the interactions between the robot and the world. Figure 6 shows the modules of the EMIB architecture used for these experiments. The objective was to make the robot learn to change its behaviour selection strategy for foraging pink blocks to a homing region, in changing environments (static with different environment configurations, or dynamic with multiple robots). Figure 7 illustrates the experimental set-up. A *Rational* module is used to learn the interaction model, while an *Implicit* module is used to activate searching or homing. Using Subsumption arbitration (i.e., commands issued by the behaviour-producing modules with the highest priority subsume the ones issued by lower priority behaviour-producing modules), eight behaviour-producing modules are used to control the robot, making it able to forage and avoid obstacles. Alternative-behaviours for following the wall, for resting and for wandering are used according to the strategy learned by the *Rational* module. Whenever a behaviour-producing module is exploited, its corresponding symbol is sent to the *Rational* module (following the subsuming organisation of the behaviour-producing modules), generating the sequence of behaviour-producing modules exploited over time. This sequence is encoded into a tree representation of the history of behaviour exploitations, constructing one interaction model for searching blocks and one for going to the home region.

Learning is done in a reinforcement fashion (i.e., by trial-and-error and without being told exactly what it should do at all times), but without specifying a reward signal based on characteristics about the environment or the task. Instead, performance is based on a self-referenced method, by using the amount of time behaviour-producing modules are exploited. Comparison between the time spent exploiting behaviour-producing modules associated with the task (like *Searching* or *Homing*) and the time spent exploiting maintenance behaviours (like *Avoidance*) is used to derive the evaluation criterion. This way, behavioural selection strategy is derived from what can be learn from the experiences of the robot in its environment, without having to characterise a priori the operating conditions of the environment. Results obtained with this approach show that the robot is able to learn unanticipated (like resting in front of a static obstacle to increase the turning angle and locate a pink block placed in the centre of the pen, taking into account the unforeseen dynamic of the robot’s capabilities) and original (like yielding when coming close to other agents or following walls when the centre of the pen is crowded) behavioural selection strategy, in stationary and non-stationary conditions. By monitoring and modelling how behaviour-producing modules are exploited (not just activated but actually used based on conditions in the environment) over time, the approach tries to characterise what emerges from the behaviour of the robot in its environment.

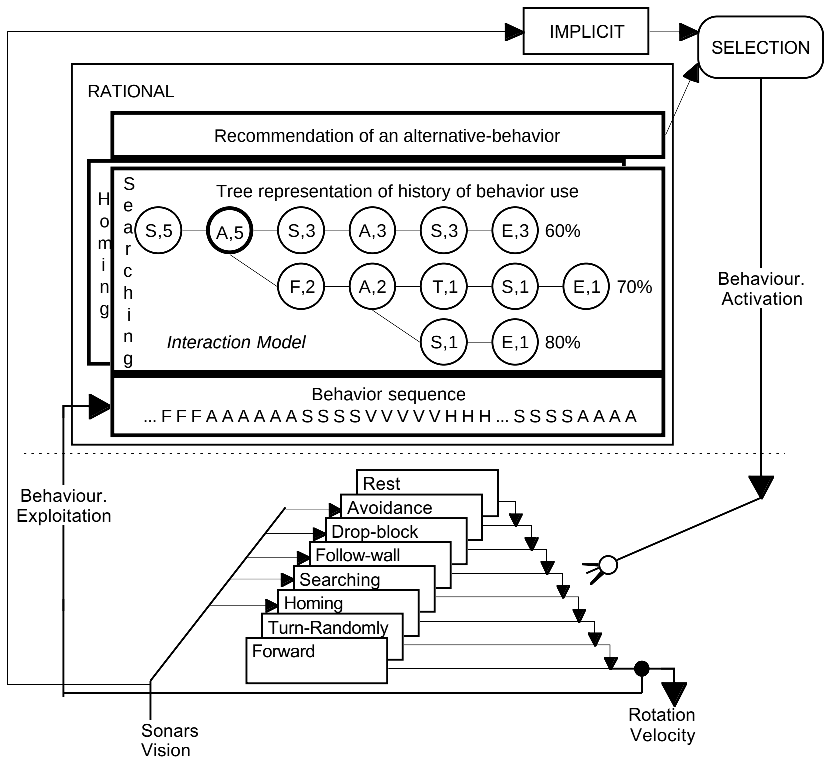


Figure 6: Implementation for the learning experiments.

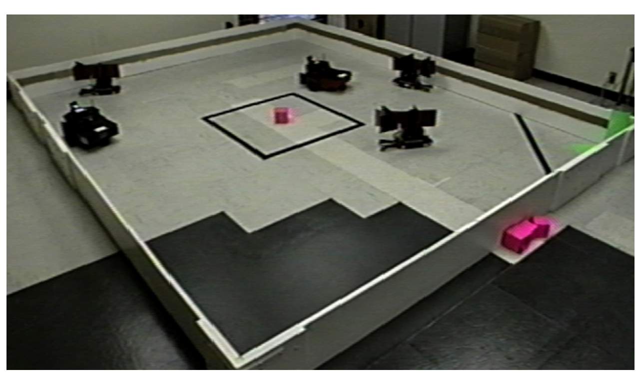


Figure 7: Experimental set-up for the learning experiments.

Goal-management using motives

An autonomous robot needs a mechanism to manage conflicting goals. Because of inherent uncertainties for robot operating in unconstrained environments and with limited knowledge and abilities, conflict management and planning between goals should emerge too from the interactions the robot has with its environment.

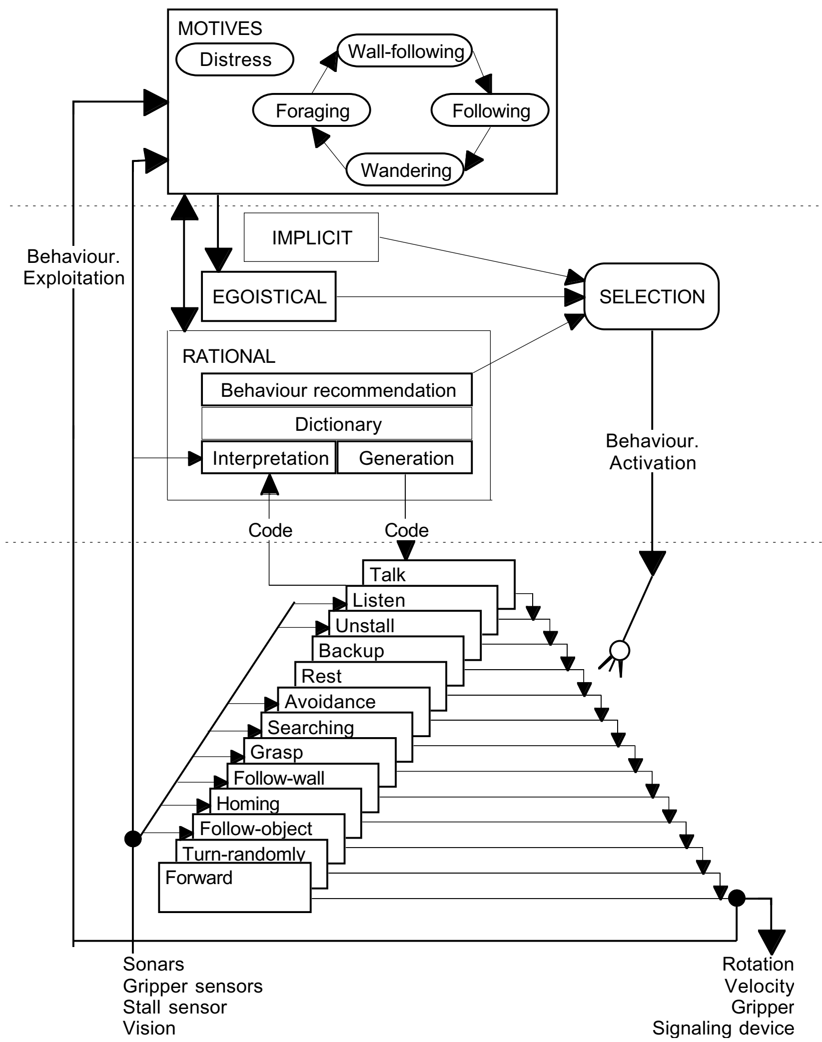


Figure 8: Implementation for the goal-management experiments.

This experimental case involves a human interlocutor interacting with a mobile robot using a light-signalling device for communication. A light-signalling device simply consists of a coloured light that the agent (the robot or the human interlocutor) can turn on or off according to a coding protocol. Perception of the signal is done by the robot using a colour-detection vision system. We chose to experiment with a light-signalling device to understand the advantages and limitations of this communication method, and how it could be complementary to standard communication media (like radio transmission). Visual communication may be limited in range, motion and bandwidth. However, having the agents relatively close to each other makes them share the same perceptual space, which allows them to sense or deduce information concerning the context of their interaction (like the location of the interlocutor, objects they perceive, etc.) without having to communicate explicitly such information. This helps establish a shared meaning between the agents.

In the experiments, the robot had the following goals: 1) Getting out of trouble (using motive *Distress*); 2) Wandering (using motive *Wandering*); 3) Following walls (using motive *Wall-following*); 4) Following a moving agent (using motive *Following*); 5) Foraging (to search for blocks and bring them to a specific location, using motive *Foraging*). The implemented architecture is shown in Figure 8. Goals 2 to 5 are task-oriented, and the robot can only accomplish each of them one at a time. Conflict between these goals is managed by activating only the task-oriented motive (*Foraging*, *Wall-following*, *Following*, or *Wandering*) with the highest energy level, setting its energy level to 100% and letting the motive remain active until its energy level drops to zero. Meanwhile, the activation levels of the other task-oriented motives are set to 0%. The *Rational* module is used in these experiments to interpret the signals communicated using the signalling device.

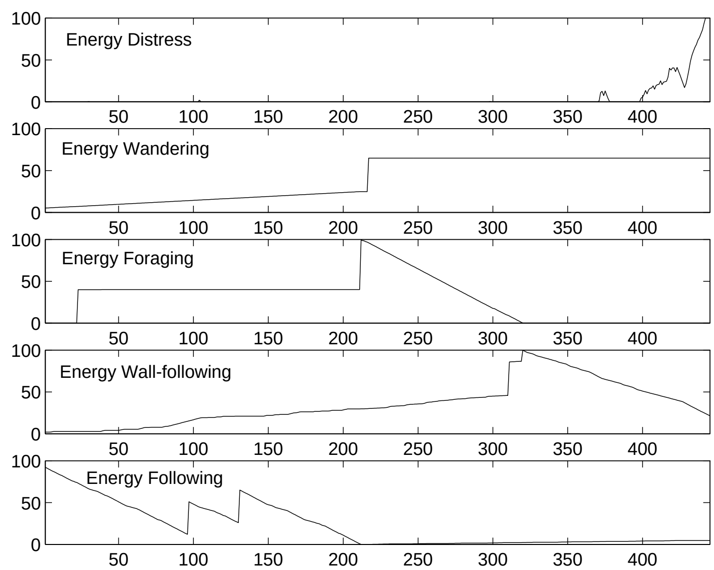


Figure 9: Energy levels of motives over time (in sec).

Figure 9 shows the energy level of task-oriented motives in one trial. The robot first decides to activate *Following*, then to activate *Foraging* (around the 210th sec) and finally *Wall-following* (around the 320th sec). Since these motives influences one another (as seen by the loop shown in the *Motives* module of Figure 8), these goals are to be accomplished regularly by the robot according to previous goals and to particular conditions experienced in the environment. For example, seeing a pink block will encourage the robot to start to forage, while no exploitation of the behaviour-producing modules associated to foraging will decrease the energy level of the motive. If the robot frequently detects walls, then *Wall-following* gets energised and eventually takes precedence over the others; it remains active for a time proportional to the ability of the robot in following the walls. It will also influence the energy level of the *Following* motive.

Requests can also be communicated by a human interlocutor, influencing the energy level of the associated motive. These requests can be seen in Figure 9 by a 40% increase in the energy level of a motive, causing it to be activated longer (as for the *Following* motive) or to take precedence over another motive (as it happens with the *Foraging* motive). Also note that the robot encountered some difficulties at the end of this trial, as indicated by the energy level of *Distress*. When *Distress* reaches 100%, the robot communicates its state, hoping to receive some outside help. So, based on this experimental case, using motives and communication add another important dimension to emergence, since the robot's goals can be influenced by the environment, the experiences of the agent and communicated information.

Conclusion

Being able to model the world, plan, predict events, deliberate from possible alternatives, learn, adapt to the contingencies of the environment and many other capabilities are important manifestations of intelligence. Designing an agent that demonstrates all of them is a complex problem usually addressed from smaller parts. This paper presents a computational architecture that tries to reach this objective by integrating modularity principles associated with hierarchical architectures, functional architectures and behaviour-based architectures.

The EMIB computational architecture has three-level: behavioural, recommendation and motivational. It distinguishes itself from other computation architectures by being able to integrate behavioural decision-making with reasoning (in a more traditional AI perspective) and also motivation and emotion. Some similarities exist with the ALLIANCE architecture (Parker, 1998). However, in ALLIANCE behavioural selection is done only using a motivational subsystem, and not from multiple sources. The EMIB architecture integrates more influences for selecting and configuring behaviour-producing modules. Albus' computational architecture (Albus, 1991) is similar to EMIB by also considering the influence of emotions (in the value judgement module), but differs in its dependence on a symbolic and central world model and in its hierarchical decomposition with different planning scopes (in time and in space, up to 7 levels).

Without prohibiting the use of symbolic knowledge representation and world modelling, the EMIB architecture preserves and exploits emergence, not only at the behavioural level, but also to reason about what the agent is experiencing in the world. This is achieved at the Behavioural Level by having concurrent use of behaviour-producing modules, and having them be selected and configured also by concurrent recommendation modules. No strong interdependencies exist between the recommendation modules, and no one has the control over the others. This adds another dimension to the emergence of functionality associated with behaviour-based approaches, by having behaviour-producing modules be selected and configured by independent sources. Using information derived by behaviour-producing modules is also another useful source of knowledge about the environment, as generated by the agent's capabilities in the topological graph experiment. Finally, looking at how behavioural modules are exploited over time is also a key factor in making agent more intelligent, as explained in the experiments. Using this information, the architecture also allows to model and to reason about what emerges from the interactions of the agent in its environment, as influenced by its decision and control capabilities. It allows the decision-making processes of the agent to be more influenced by its own reality, as defined by its own capabilities and its experiences in the world. We believe that such capabilities are required for designing autonomous agents that have to deal with complex, changing, unpredictable and partially observable operating conditions.

As illustrated in the three experimental cases presented in the paper, different implementation of the EMIB computational architecture is possible, based on the capabilities required by the agent. The architecture makes no assumptions on the mechanisms implemented in the decision modules, nor does it require that all of the decision modules be used in an implementation. The EMIB decision-making modules can then exploit different kinds of representations, adhering to the modules' purpose and the interfaces between them.

By continuing to follow the organisational principles of the EMIB architecture in the design of autonomous mobile robots, we hope to gain a better understanding of the various dimensions related to intelligence (like reactivity, planning, modelling, learning, communicating, interacting, motivation, emotion and so on) to design increasingly intelligent agents under the same conceptual framework. Currently, many projects based on the EMIB architecture are underway. One aims at formation control in complex environment and with heterogeneous robots. Another involves prolonged activity of a group of robots having to survive in an enclosed area by sharing one charging station while doing a foraging task. These projects can be associated with some of the universal adaptation problems outlined by Plutchik (1980), making possible to study the use of artificial emotion in such contexts.

Finally, we do not claim that the conceptual framework of the proposed architecture is the only one to be followed to design intelligent agents: there are still too many things unknown about intelligence to make such a claim. Improvements are still required, like having a mechanism that allows to dynamically change the priorities between the behaviour-producing modules. We also need to refine and test our model of artificial emotions. As of now, we validated the model by using it in our entry to the AAAI 2000 Mobile Robot Challenge (Michaud, Audet, Létourneau, Lussier, Théberge-Turmel et al., 2001). Our objective was to study how such generic mechanism can benefit the robot in making self-assessment of its situation in the environment and of the accomplishment of its goals, and used for human-robot interaction. The principal problem found with the model is the difficulty in adjusting the factors influencing the motives and the artificial emotions, to get an adequate response depending on the goal pursued by the agent. In future work, we hope to refine the model to significantly simplify this process, while still ensuring its generic.

Our long-term goal is to continue to integrate intelligent decision-making capabilities in the EMIB architecture and to analyse the implementation complexity of the designs and the performance manifested by the agent in complex environmental settings. We will also continue to study other computation architectures to evaluate interesting mechanisms not considered in the EMIB architecture and to compare performances using common benchmark applications. EMIB computational architecture could then be adapted and refined according to these findings. Such integration will certainly be beneficial in designing autonomous mobile robots increasingly more intelligent, capable of sharing their existence with us in real life settings.

References

Albus, J.S. (1991). Outline of a theory of intelligence. *IEEE Transactions on Systems, Man, and Cybernetics, 21*(8), 473-509.

Arkin, R.C. (1989). Motor schema-based mobile robot navigation. *International Journal of Robotics Research, 8(4),* 92-112.

Arkin, R.C. (1998). *Behaviour-Based Robotics.* Cambridge MA: MIT Press.

Beck, R.C. (1983). *Motivation: Theories and Principles.* Prentice Hall, Second Edition.

Breazeal, C. (1998). Infant-like social interactions between a robot and a human caretaker. *Adaptive Behavior, Special Issue on Simulation Models of Social Agents.*

Blumberg, B.M., Todd, P.M. & Maes, P. (1996). No bad dogs: Ethological lessons for learning in Hamsterdam. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior,* MIT Press, 295-304.

Bonasso, P., Firby, R.J., Gat, E., Kortenkamp, D., Miller, D.P., & Slack, M.G. (1995). Experiences with an architecture for intelligent, reactive agents. In *Proceedings of the International Joint Conference on Artificial Intelligence.* AAAI Press.

Brooks, R.A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation, RA-2(1),* 14-23.

Brooks, R.A. (1991). Intelligence without representation. *Artificial Intelligence, 47,* 139-159.

Dolan, S.L., & Lamoureux, G. (1990). *Initiation à la Psychologie du Travail.* Gaëtan Morin éditeur.

Frijda, N.H. (1987). Comment on Oatley and Johnson-Laird's 'Towards a cognitive theory of emotions'. *Cognition and Emotion, 1(1),* 51-58.

Maes, P. (1991). Situated agents can have goals. In *Designing Autonomous Agents: Theory and Practive from Biology to Engineering and Back,* P. Maes (ed.), 49-70. Cambridge MA: MIT Press.

Maslow, A.H. (1954). *Motivation and Personality*. Harper and Row.

Mataric, M.J. (1992). Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence, 9(2-3)*, 323-336.

McCarthy, J. (1995). Making robots conscious of their mental states. In *Proceedings AAAI Spring Symposium*.

McFarland, D. & Bösser, T. (1993). *Intelligent Behavior in Animals and Robots*. Cambridge MA: The MIT Press.

Michaud, F., Audet, J., Létourneau, D., Lussier, L., Théberge-Turmel, C., Caron, S. (2001). Experiences with an autonomous robot attending the AAAI conference, *IEEE Intelligent Systems, 16(5),* 23-29.

Michaud, F., Lachiver, G., & Le Dinh, C.T. (1996) A new control architecture combining reactivity, deliberation and motivation for situated autonomous agent. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior* (pp. 245-254). Cambridge MA: The MIT Press.

Michaud, F., Lachiver, G., & Le Dinh, C.T. (2001) Architectural methodology based on intentional configuration of behaviors. *Computational Intelligence*, *17(1)*, 132-156.

Michaud, F. & Mataric, M. J. (1999). Representation of behavioral history for learning in nonstationary conditions. *Robotics and Autonomous Systems, 29(2)*, 1-14.

Michaud, F., Prijanian, P., Audet, J., & Létourneau, D. (2000). Artificial emotion and social robotics. In *Distributed Autonomous Robotic Systems* (pp. 121-130).

Michaud, F. & Vu, M.T. (1999). Managing robot autonomy and interactivity using motives and visual communication. In *Proceedings of the Autonomous Agents Conference* (pp. 160-167).

Minsky, M. (2002). *The Emotion Machine – 1. Emotional States*. URL: web.media.mit.edu / ~minsky / E1 / eb1.html.

Mook, D.G. (1987). *Motivation. The Organization of Action.* W. W. Norton and Company.

Murphy, R.R. (2000). *Introduction to AI Robotics*. Cambridge MA: MIT Press.

Oatley, K. & Johnson-Laird, P.N. (1987) Towards a cognitive theory of emotions. *Cognition and Emotion*, *1(1)*, 29-50.

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

Parker, L.E. (1998). ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, *14(2)*, 220-240.

Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In *Emotion: Theory, Research, and Experience*, R. Plutchik & H. Kellerman (eds), *1*, 3-33.

Saffiotti, A. (1997). The uses of fuzzy logic in autonomous robot navigation. *Soft Computing*, *1*, 180-197.

Saridis, G.N. (1983). Intelligent robotic control. *IEEE Transactions on Automatic Control*, *AC-28(5)*, 547-557.

Simon, H.A. (1967). Motivational and emotional controls of cognition. *Psychological Review, 74,* 29-39.

Smithers, T. (1994). On why better robots make it harder. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior* (pp. 64-72). Cambridge MA: The MIT Press.

Staddon, J.E. (1993). On rate-sensitive habituation. *Adaptive Behavior, 1(4),* 421-436.

Velásquez, J.D. (1998). A computational framework for emotion-based control. In *Proceedings of the Workshop on Grounding Emotions in Adaptive Systems – Conference on Simulation of Adaptive Behavior*.

Werner, E. (1992). The design of multi-agent systems. In *Decentralized AI 3 – Proceedings of the Third Europeen Workshop on Modeling Autonomous Agents in a Multi-Agent World*, E. Werner & Y. Demazeau (ed.) (pp. 3-28). Elsevier Science.