Research Article Open Access

Darryl N. Davis* and Shylaja Kanaganapalli Ramulu

Reasoning with BDI Robots: From simulation to physical environment – Implementations and Limitations

https://doi.org/10.1515/pjbr-2017-0003
Received July 14, 2017; accepted September 10, 2017

Abstract: In this paper an overview of the state of research into cognitive robots is given. This is driven by insights arising from research that has moved from simulation to physical robots over the course of a number of sub-projects. A number of major issues arising from seminal research in the area are explored. In particular in the context of advances in the field of robotics and a slowly developing model of cognition and behaviour that is being mapped onto robot colonies. The work presented is ongoing but major themes such as the veracity of data and information, and their effect on robot control architectures are explored. A small number of case studies are presented where the theoretical framework has been used to implement control of physical robots. The limitations of the current research and the wider field of behavioral and cognitive robots are explored.

Keywords: cognition, BDI, affect, veracity, swarm robots

1 Introduction

There is continued interest in the development of cognitive robots [1–5] as theoretical concepts [6] and as implementations, whether laboratory based [7], or application focused [8]. Application areas for such robots, whether as single artifacts or ecologies, can include environments unapproachable or dangerous for human beings [9], assistive robots in the home, particularly as part of a larger ambient digital ecology [10], and the next generation of industrial robots [11, 12]. The focus of these studies is wide and research into robotics can seem fractured with increasing diversity and subsequent specialisation. Indeed as Al-Razgan *et al.* [13], highlighted there has not been a systematic review of the field in general for some time. However several issues are common across these many domains: the link between perception and action; task modelling; layered control; navigation; adaptation and learning; and social interaction between robots and other agencies.

This article makes a case for developing cognitive architectures as internally consistent control architectures utilizing motivation and other control states based on the veracity of sensory and communicated information. Such control architectures will be capable of a variety of behaviours and tasks in a variety of environments, and not limited to a bespoke application. The architectures developed here do this through the utilisation of metrics across control states that are realised in a probabilistic Belief-Desire-Intention reasoning model. Earlier research on cognitive architectures has focused on the control of a single agent, using either simulations [14] or a single robot [15, 16] in known and dynamic environments (typically populated by non-collaborating agents or robots). Here reasoning and representational issues that needed to be addressed, in moving from simulation and single robot work towards colonies of interacting robots in unknown environments, are outlined and discussed.

2 Background

Here the interest is in robotic control architectures and techniques that empower the robot with the ability to generate intelligent behaviour. One such area of study that aims to do this, is that of cognitive robotics. Researchers in this area admit that there are a number of considerable challenges in adopting cognitive rather than hard-coded dedicated robots [17]. In their overview of cognitive robotics, Levesque and Lakemeyer [1] build upon an earlier manifesto [18]. The ultimate aim of their analysis is

***Corresponding Author: Darryl N. Davis:** School of Engineering and Computer Science, University of Hull, Cottingham Road, Kingston-upon-Hull, HU6 7RX, UK, E-Mail: D.N.Davis@hull.ac.uk; Tel.: +44(0)1482-466469; Fax: +44(0)1482-466666
Shylaja Kanaganapalli Ramulu: Ambedkar Institute of Technology, Nagarabhavi 560056, India, E-Mail: tejushyla@yahoo.co.in

understanding how to develop high level control systems for robots that work across application domains and tasks. They address principles and conceptual issues faced in developing robot systems for uncertain and dynamic environments, and the relationship between perception, reasoning and the granularity of knowledge. Of particular relevance to the research presented here is the delineation of given knowledge from acquired domain and task knowledge, the acquisition of new beliefs from perception and reasoning, and the temporal validity of held knowledge.

Langley *et al.* [19] outline research issues and challenges in the more expansive area of cognitive architectures. Although not directly targeted at the robotics domain, the paper discusses issues that are relevant, for instance: what constitutes a cognitive architecture; the nature of learning in cognitive architectures; perception and the cognitive processes associated with it such as recognition, categorization and situation assessment; the nature of decision making and the provision for alternative choices; planning, execution and action.

These two seminal papers [1, 19] cover a large range of issues, and not all will be addressed in this paper, or indeed, as stated, are particularly relevant for embodied robots in physical environments. One way of viewing these issues is to consider the sensing to perception to reasoning to action control arc, and the types of control signals and knowledge relevant to that as the arc is expanded and deepened. Levesque and Reiter [18] phrase this in terms of the relationship between the knowledge, the perception, and the action of the robot.

A naïve approach is to view the sensing to action arc as a feedforward process where processing internal to the robot control architecture maps perceptual information onto intended action (i.e. a sense-think-act cycle). This can be trivially changed to allow perceptual actions to be the result of internal action, i.e. think-sense-think-act, particularly where the robot has a choice in how it uses its sensors. Less trivial changes involve the use of feedback from action to modify perception, or other decision processes, or indeed the type of perceptual and decision processes that are relevant to the robot's "understanding" of the current environment and its role within it. This ultimately captures the two issues (conceptual problems) that Levesque and Reiter term as "*when should a robot use perception to find out if something is true as opposed to reasoning about what it knows was true in the past?*", and "*when should the inner workings of an action be available to the robot for reasoning and when should the action be considered primitive or atomic*" [18, p. 106].

Addressing the first issue can be particularly problematic in rapidly changing, busy, cluttered or complex environments. Depending upon how the designer constructs the robot control architecture, this issue may be resolved as new sensory information is combined with existing knowledge. In non-deterministic environments, and this covers many multi-robot and multi-agent environments, the robot designer will need to address the issue that knowledge about the world that a robot inhabits is inherently incomplete and, at best, slightly out-of-date. Indeed, even in relatively simple environments totally under the control of one robot, situations may arise where actions cause unexpected side-effects, for example building a tower or bridge from blocks that subsequently collapses leading to erroneous knowledge about the position of those building blocks.

The second issue relates to the nature of action primitives. In simulation studies, or limited physical environments, it may be possible to use a framework analogous to that proposed by Minsky (in the Society of Mind [20]), where truly fundamental actions (for example, parameterised use of single motors or specific actuators, as described in the shared circuits models of Boza *et al.* [21]) are used as primitives and a network of behaviour activity is built upon them. Indeed, a similar viewpoint is a central to epigenetic (or developmental) robotics [22, 23]. Previous work has partially addressed this issue through the delineation of behaviour as reflex, reactive, deliberative and higher order behaviours [14]. Some concurrent work using simple robots in adaptive swarm topologies is approaching this issue using minimalist version of our cognitive architectures and a small repertoire of Braintenberg-like behaviours. This approach to distributed cognition will be returned to in later sections of this paper.

A fundamental issue relates to the knowledge used in these processes, and the differentiation between control, task and behavioural knowledge. Control knowledge is that which is used in processes that select different modes of action selection, reasoning or indeed initiate learning processes. Task knowledge can be thought of those types of knowledge to be found in most planning systems or Hierarchical Task Networks; although these may be embedded within the robot control architecture as reactive or other types of situated behaviour. Behavioural knowledge describes these latter type of behaviours and can be represented in a number of ways, for example finite state machines, procedural rules or as part of more encompassing representations (as discussed later in the section on motivational control states). Underpinning the use of this control, task and behavioural knowledge is the data and information gathered from sensors and communication. Research in other domains, for example machine learning and decision support [24, 25], suggests that all sources of data (including sensors and communications) should be deemed suspect, even in benign domains and situations. It is contended that belief models that fail to take this into account lack veracity and are destined to fail. In unpredictable, nondeterministic robotic domains with less than perfect sensors, and particularly where not all agencies can be trusted, for example [26], this leads to, at best, inefficient behaviour; at worst, catastrophic failure. Others, for example Kowalczuk and Czubenko [9], have also noted this.

3 Frameworks for Cognition and Noncognition

Central to the issues introduced above, is the idea of what to do next, and why. Or if a more thorough consideration is required, the what and the why (of goal management) can be extended using a framework similar to that outlined in the ontological work known as H5W [27]. Others have also tried to capture a broad spread of investigative perspectives in addressing what is meant by motivation, for example [28].

The concept of motivation, and allied affective states, has been discussed broadly in many disciplines. The requirement for affective states in complete cognitive theories and architectures was highlighted in earlier prominent research [29, 30]. For intelligent control architectures, affect is a useful concept, providing the means to relate similar and dissimilar process types across different levels of abstraction in a complex control architecture. This is particularly apt for cognitive robotic control systems where the reason for doing something can vary across tasks and domains, or even for the same tasks within a changing environment in one domain. The basis for much work on motivation has been research into the nature of emotion. This is not addressed in depth here, other than to note that theories of emotion can be typified as belonging to one of several types, as noted in earlier papers [31]. For example physiological [32], evolutionary [33], expression [34], appraisal [35] or goal based [36]. Some [35, 36] consider emotions to be a cognition centered set of phenomena, while others [37] consider them to be centered on low-level (neurophysiological) control processes that affect cognition. This is partially due to different programmatic objectives within differing academic disciplines, for example, neurophysiology, psychology, philosophy and cognitive science. Given that there is no one accepted theory or definition of emotion, the research described here has proceeded with “just sufficient” definitions to enable clear differentiation between cognition and non-cognition. This allows for clearly defined (motivational) control states, based on concepts that draw on theories of affect and cognition in biological systems. For example, as described later, a synthetic system need not model or recognise the emotive state termed “curiosity” but recognise highly valenced internal states and/or environmental affordances that give rise to motivational states and intentions about investigating a situation, and thus enable the control architecture to make intelligent decisions about what to do and how to achieve it. To paraphrase Levesque & Lakemeyer [1], in order to perform a task, what information or knowledge is required by the robot at the commencement of the task and what can be acquired via perceptual, communicative or other means during the execution of the behaviours that attempt to complete that task? Given the adage that “the study of intelligence is akin to understanding an alien in an unknown environment”, what does a robot need to know about its environment as given by its designer? It is more important and certainly more enabling to design a robot with the means to determine what is in its environment and what affordances those found entities offer. Along the way to designing control architectures that transcend robot controllers that solve a class of problems or that work in a class of application domains, it is useful to discover what are the minimal designs for fruitful behaviour and how those minimal designs can be extended with the capability to adapt.

Robot architectures designed and implemented to address these issues tend to be hybrid deliberative/reactive systems that range from adaptive parameterization to constructs that affect the operating mode of the robot. For example the fuzzy cognitive models of Kowalczuk and Czubenko [9], or the motivated robot architecture extensions to behaviour based models [38]. Without venturing too far into the argumentation over symbolic and connectionist approaches to cognition [39], and retreading existing arguments, the application of control theory [40] has seen a number of interesting approaches to the design of artificial cognitive control systems; albeit most of this work is limited to quantitative models of motivation, for example [11]. Work on modified shared circuits models [21] and TSK fuzzy models [41] are analogous to the low level Rolls’ affect models indicated above [37]. As such these provide efficient models for the selection of reactive behaviours in industrial and manufacturing environments, and indicate the growth of control theory into cognitive control systems. The reported research is indicative of the greater fluidity in control models required for twenty-first century manufacturing; the insight they bring to the design of artificial mind systems is limited by the operational frameworks of the research and seem to lack the philosophical breadth of more encompassing studies. Vascak and Hirota [42] indicate that it is possible to use these parameterised control models in an architecture that utilises richer representational modes (Fuzzy Cognitive Maps); however that work is limited to a small set of behaviours using simulated robots, albeit later extended to a greater set of behaviours with real robots in [43].

A relatively recent survey of motivational frameworks [44] presents an architectural framework for intelligent (cognitive) systems, with a specific focus on goal management. It highlights a number of problems with existing systems related to compromises between the requirements of the task environment and the design features of intelligent systems developed to fulfil those requirements. It suggests the use of deliberative planners to reason about motives, goals and the behaviour and action sequences that satisfy those goals. It further suggests the use of goal management to handle the varied states associated with goals. The research presented there is in partial agreement with this requirement as detailed in [31], and fits with the discussion above on the differentiation between control, task and behavioural knowledge. The paper presents a number of open questions for goal-driven intelligent control mechanisms that are addressed in the concluding section of this paper.

These issues and topics provide a very broad framework within which general purpose robot architectures can be discussed. Biologically Inspired Cognitive Architectures (BICA) [45] address the issues associated with the greatest sophistication of design space associated with intelligent life forms. Like GC-5 [46] before, BICA has set its aim to the challenge of creating a real-life computational equivalent of the human mind. In developing intelligent control architectures for systems that manipulate robots, or reason about actions, or perform any number of tasks, replicating the type of errors human agents make, as highlighted by Kahneman and Tversky [47], is not viewed as beneficial. There is a wide range of design spaces for cognitive control systems beyond this ambition embodied in the BICA challenge. Holland *et al.* [45] consider that emotion-like processes are possible with embodiment (i.e. in a cognitive architecture considered as part of a robot) but difficult or untenable in virtual systems. This argument is flawed in that it oversimplifies the research into emotion, and dismisses the wealth of research that has used, and continues to use, synthetic test-beds (i.e. simulations) as a precursor to physical robots, or that the intended domain is too hazardous or untenable with which to experiment.

Quirin *et al.* [48] use positive affect, rather than emotion per se, to drive switching within a multiple level robot control architecture. The affective model allows the interruption of low level behaviour modules by a more inefficient, information rich, planning system as and when the behaviour modules begin to fail, and so allow adaptive plan execution. The focus is on how to handle (unforeseen) exceptions to plans once those plans are acted upon by the behaviour modules. Although their affect model differs greatly from that used in CAMAL (Computational Architectures for Motivation, Affect, and Learning), their approach to integrating affect into a hybrid deliberative-reactive control architecture is similar.

A further paper from the BICA field is particularly relevant to the current work. Sanz *et al.* [49] address how emotion can be used to direct control within a cognitive architecture. As with the current research they are more interested in the use of emotion (or affect) as mechanisms to direct control in a sophisticated cognitive architecture, rather than model human/biological agent emotion. Emotion is used at a meta-level to configure the architecture and allow adaptation. Of particular interest are the issues that remain problematic to solve, many of which the reported research addresses, including: representation of affect in the regulatory processes and in the goal terms that manipulate; and establishing emotion dynamics without losing performance dependability.

4 Cognitive Architectures for Motivation, Affect and Learning

CAMAL (Computational Architectures for Motivation, Affect, and Learning) is an example of a general class of integrative cognitive architectures; drawing together a number of threads in Cognitive Science and Artificial Intelligence, for example perception, action, decision making, motivation, affect, and learning. CAMAL [14] has been in development over 15 years. It is essentially a design that addresses the question of what is a Theory of Cognition, as described by Newell [50], and tries to answer some of the issues that comprise Norman's Cognitive Science agenda [30]. Based on earlier work [51, 52], it has been developed to address the limitations of initial and incremental implementations [15, 16, 53, 54]. While the

CAMAL project draws on research in human cognition, it addresses many of the issues raised above and in doing so aims to produce adaptable machine intelligence, and not to produce human-like artificial intelligence. The ongoing project has not been driven by any specific testbed or application area; it is a computational design and implementation testbed with which to experiment with ideas from cognitive science and see how they map onto computational platforms, including robots of varying sophistication. Hence the design and implementations in simulation and robotic platforms have, to date, tried to remain general rather than focused on specific application issues.

The CAMAL model is a construct for investigating artificial minds that draws on formal (BDI) reasoning, probability models, affective computation, reactive machines and self-configurable architectures. The latest designs (schematically shown in Figure 1 and further developed from earlier versions) have been developed using both synthetic and real robots. A distributed blackboard combines motivator management, goal selection, behaviour determination, sense and action in a feedback cycle as it adapts given knowledge and task definitions to the found environment.

Throughout the remainder of this paper CAMAL is used to refer to the overall architecture, as indicated in Figure 1. Although initially implemented as a large single agent architecture from 1998 onwards, the conceptual framework holds true for, and has been developed for, the distributed implementations described here. The upper schematic in Figure 1 shows reflective and deliberative processes. The reflective processes monitor the colony control and the motivator and affective processes at the deliberative layer. Colony control is represented as the BDI architectures that are in effect for the current investigation (and can vary, as described in the next section of this paper). More compact versions of the architecture are mapped onto individual robots. These are individual cognitive robots referred to as SCARAB (Situated Cognitive Architecture for Reasoning About Behaviour) or cognitive swarms within the overall colony (SwarmBots). The upper architecture in Figure 1 is a "robot-less' CAMAL architecture that reasons over the data and information provided by the network of robots. The lower large architecture depicts a SCARAB instantiation. The lower two (Reactive and Reflexive) layers in the SCARAB architecture would be the only components running on one of the SwarmBots with limited on-board processing (see Figure 3 and associated text).

The currently favoured BDI set, on the abstract agency, can be modified by the motivator blackboard and result in an intended colony behaviour (seen in Figure 1 as Is). This Intention set is transmitted to the colony robots (Ri, Rj, Rk and Rl in Figure 1) with responses from that robot set according to how those intentions map onto the BDI model activated locally in the robot set (as seen for robot Rk in the lower portion of Figure). This can only occur if the robot possess BDI capabilities locally. Not all robots (see Figure 3 and associated text) are so equipped.

The lower portion of Figure 1 shows reflexive, reactive and deliberative processes within one robot (Rk); metacontrol here is limited to adaptation of the BDI parameters via behaviour Feedback Adaptation. Reflexive behaviours (for example Rx_i) are low level direct responses to sensor mappings (such as halt if obstacle detected). Reactive behaviours are suites of these reflexive behaviours that can be triggered by sensory mappings (such as Rb_i and Rb_j), or tightly coupled with BDI combinations (such as Dx-Rb_x and Dy-Rb_y). Reflexive and reactive behaviours can combine, be overridden (by BDI D_x-Rb_x couplings) or override BDI couplings according to the parametrization of the SCARAB architecture (again norms built at initialisation or modified through communication from the colony controller). The following sections provide details using examples from one set of experiments with SwarmBots and one set of experiments using more sophisticated robots running the SCARAB implementations.

The current work is driven by test-beds that comprise of a set of (over 15) non-homogeneous robots, with some development work using a simulator [55]. This has required a conceptual redesign of several parts of the cognitive architecture for several reasons. The CAMAL architecture was developed in order to address a number of research issues relating to cognition, affect and behaviour. It has become quite complex with many capabilities not required by the simpler robots, for example an extensive meta-cognitive capability. It was developed initially with a view to being implemented on single agents in mainly simulation work with limited application to the control of robots in a physical environment. CAMAL, and its variants was moderately successful in the control of single robots [15, 16]. Here we consider its limitations and the path forward to its use in the control of multiple robots of differing capabilities. The current work has led to significant changes to the architecture so not only can it control multiple robots but provide a general mechanism for the tasks those robots are expected to perform.

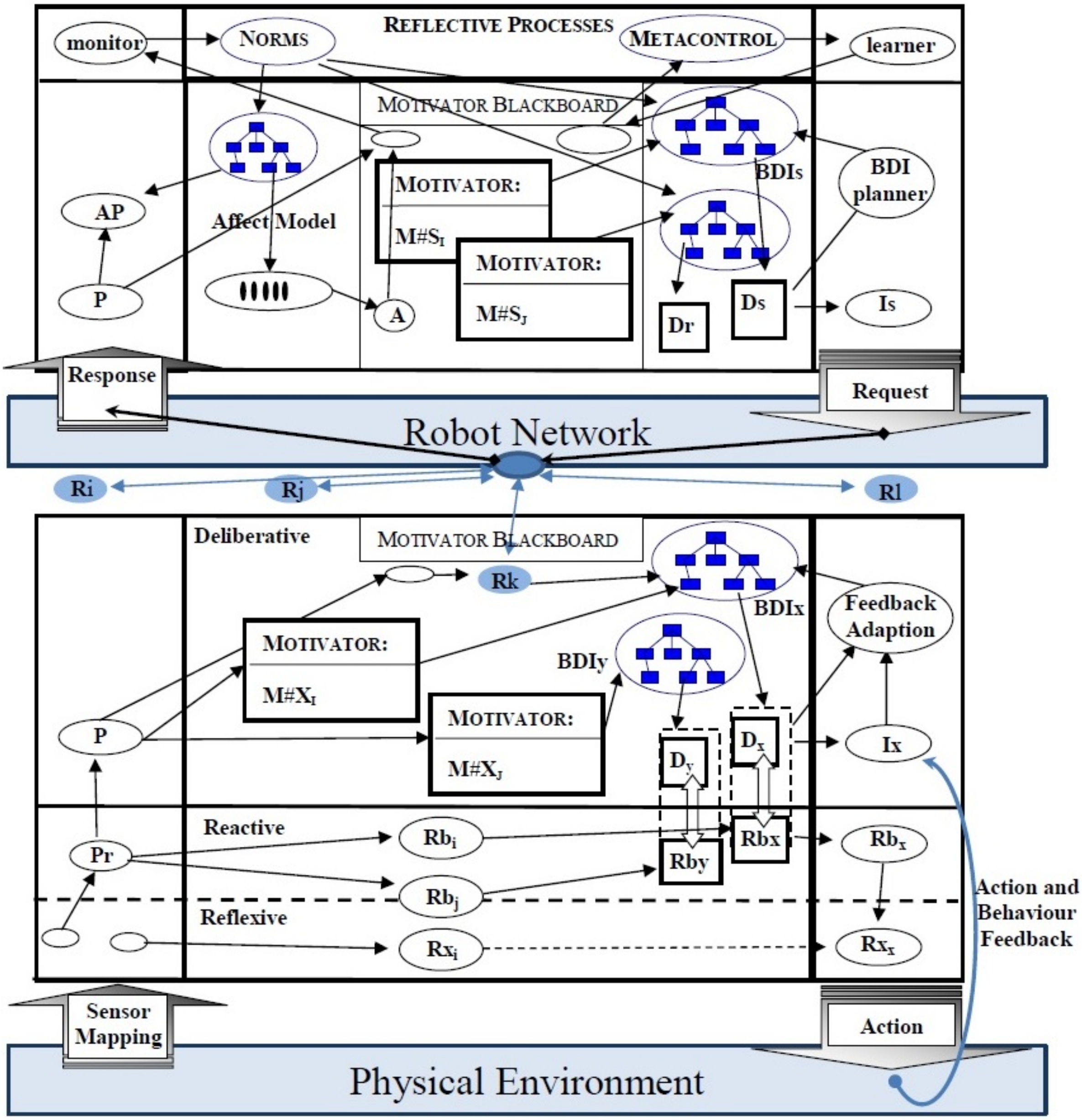


Figure 1: Robot Network and Control Architecture. The upper schematic represents an abstract agency that sits above any number of physical agents (i.e. robots) shown as the blue ellipses associated with the Robot Network (Ri, Rj, Rk, Rl); one of the robot ellipses(Rk) is expanded (as in the lower schematic) and is one of the more sophisticated robot designs (shown to right in Figure 3). The abstract agency may actually reside on a robot with sufficient processing power and/or simultaneously elsewhere across a communication network (as indicated in Figure 3).

4.1 Control of reasoning with Beliefs, Desires and Intentions

Reasoning in CAMAL occurs over multiple layers and multiple representational constructs. At its simplest and lowest level, use is made of simple parameterized actions running just above and controlling the robot microcontroller. Such actions can include sensing (using a specified sensor) and movement of the robot or some component part. These can be combined into behaviours at the reactive level. The selection of behaviours and/or actions is via a variety of Belief-Desire-Intention (BDI). The nature of the BDI reasoning is in turn controlled by higher level (deliberative) processes, and norms. Reflecting this separation are different time models. The deliberative (BDI) work on a time cycle that reflects the BDI updating process given below. Within any one BDI cycle, there exist at most one call to the reactive sub-architecture. The reactive sub-architecture has its own time model and may be constrained to a specified limit of (reactive behaviour) cycles or allow open ended processing. There are then the time models occurring immediately above the robot microcontroller plus the microcontroller model. In many cases the result of a BDI update is a call to the reactive then robot microcontroller, with a subsequent surfacing of feedback from the lower layers back to the initiating BDI processes. However the architecture does allow a more complex interaction across the layers if lower layers require higher level processing in order to complete their actions. In this case, for example the convex hull mapping described in section 7, a tightly bound 'reactive call'-percept-belief update cycle is initialised within the one BDI cycle.

The principle behind orthodox BDI models, as originating from Rao and Georgeff [56], is that as a Belief set is updated through perception, reasoning, communication or other means; this precipitates a change in the viability of a Belief, Desire and Intention set, potentially leading to a change in behaviour. Hence given a new set of Perceptions (P), a preference operator and a set of Beliefs (B), the updating of an ordered set of Belief propositions (r, s, q, p, ¬p) occurs as follows:

- $P := \{ r, s, q, p \}$ (Perception set)
- $B := \{ \neg p \}$ (Belief set)
- $P \otimes B \rightarrow B'$ (Belief Revision – Various models)
  □ Preference := $\{ p \gg \neg p,\ s \gg r \gg \neg p,\ q \gg \neg p \gg \neg s \gg \neg r \}$ (example model)
- $B' := \{ p, s, r, q, \neg p \}$ (updated Belief set)
- $D := \{ p\&s \rightarrow gz,\ r\&\neg p \rightarrow g\neg z,\ s\&q \rightarrow gx \}$ (Desire set)
- $B \otimes D \rightarrow D'$ (Desire Update)
- $D' := \{ p\&s \rightarrow gz,\ s\&q \rightarrow gx,\ r\&\neg p \rightarrow g\neg z \}$ (updated Desire set)
- $I := \{ r\&gz \rightarrow b1,\ gz \rightarrow b2,\ p\&gx \rightarrow b3, \ldots, \neg p\&g\neg z \rightarrow b5 \}$ (Intention set)
- $D' \otimes I \rightarrow I'$ (Intention Mapping)
- $I' := \{ p\&gx \rightarrow b3,\ gz \rightarrow b2,\ r\&gz \rightarrow b1,\ \neg p\&g\neg z \rightarrow b5 \}$
  Behaviour b3 preferred

The perception set (P) is combined with the existing belief set (B) and ordered according to the reference operator to give B'. The new belief set, B', contains the perceptions which have been processed in the order that accords with their significance to that individual. This preference is then propagated through Desire and Intention updating resulting in the given order of behaviours.

The BDI model used in this work originates from the model used in CRIBB (Children's Reasoning about Intentions, Beliefs and Behaviour), developed to investigate reasoning in young children [57]. This schema was implemented as a computer model to simulate knowledge and the inference processes of a child solving problems [58]. Earlier research [53] saw a variety of CRIBB (a-CRIBB) developed utilizing affect in a preference operator to prioritise between competing reasoning statements and processes. The preference operator, based on a qualitative model of affect, allows discrimination between beliefs that are derived from assumptions, perceptions or deductions. Perception can be further sub-divided to include direct perception and indirect perception (i.e. from another agent that has some degree of trust associated with it). Hence beliefs can be ordered according to the degree of trust in them. This preference then cascades through goal and behaviour selection. Subsequent research [16] has seen the a-CRIBB model extended with a belief structure allowing quantitative degrees of belief to be ascribed to belief statements, according to the source of the belief. This probabilistic model has been further developed in the research reported here, with the control flow of the BDI revised in light of structural and representational changes to the cognitive control system.

CAMAL uses this cognitive BDI schema to drive a motivational blackboard. Rather than attempt to completely and accurately model an agent's world, affect can be used to guide attention so an agent is drawn to aspects of the environment deemed to be of importance. The affect ordered belief set then drives the Desire and Intention reasoning schema, leading to the selection of action that reflects the current focus of the agent as reflected at the Belief level. The latest versions of CAMAL extend the BDI model further with the now probabilistic affect model being used to drive the entire BDI reasoning cycle. Beliefs are given a likelihood of belief measure (based on affordance and affect models), Desires (or goals) an importance value and Intentions (or associations) an insistence value.

The BDI example given earlier may now look like this (given a specified belief source preference). Given a set of sensor specific perceptions ($P_{sensor1}, P_{sensor2}$), a preference operator and a set of Beliefs (B), the updating of a set of propositions (r, s, q, p, ¬p) occurs as follows:

- $P_{sensor1} := \{ s, q \}$ (Perception set from sensor1)
- $P_{sensor2} := \{ r, p \}$ (Perception set from sensor2)
- $B := \{ \neg p \}$ (Belief set)
- $P_{sensor1} \otimes P_{sensor2} \otimes B \rightarrow B'$ (Belief Revision – no logic tautology)

  □ Preference := { $P_{sensor1}$ » $P_{sensor2}$ » B }
  □ ⇒ Preference { s » q » r » p »¬p }
- B' := { s, q, r, p } beliefs status given Affordance $\mathfrak{R}[0,1]$
- D := { p&s→ gz, r&¬p → g¬z, s&q → gx } (Desire set)
- B ⊗ D → D' (Desire Update)
- D' := { s&q → gx, p&s→ gz } (updated Desire set)
- I := { r&gz→ b1, gz→ b2, p&gx→ b3,. . . , ¬p&g¬z → b5 } (Intention set)
- D' ⊗ I → I' (Intention Mapping)
- I' := { gz→ b2, r&gz→ b1, p&gx→ b3 } Behaviour b2 preferred

The nature of belief updating can be modified from a default model, at run-time, when the architecture experiences repeated failures in performing tasks (i.e. performing actions that fail to instantiate some intention on the world with which the architecture is interacting).

The selection of belief updating models (and other parameterizations of the architecture) is performed by meta-management processes (and to a lesser extent be Feedback Adaptation as shown in Figure 1). The trigger for calling these processes is part of the motivator management process. CAMAL (and SCARAB) uses a motivational blackboard (centre of the upper and lower schematics in Figure 1). This is an affect grounded BDI equivalent to Baars' Global Workspace Architecture [59], where a number of autonomous modules communicate via the global workspace, which can be viewed as motivational control state common to all processes across the architecture.

4.2 Representation of Control States

CAMAL makes use of a hybrid reactive-deliberative architecture based on the control state approach to mind [31, 51] but developed around the a-CRIBB model described above. A control state is architecture instantiation or behaviour internal to an agent. In essence, control states involve a number of representational and process concepts, such as beliefs, desires, intentions, motivators, *etc.*. They can be considered as general behaviours within an agent, and therefore can be modified or updated at any time by other processes within the agent. Control states can exhibit external behaviours (such as obstacle avoidance) or, reflect and control internal states (such as beliefs, belief revision models *etc.*). The motivational construct that is central to the motivational blackboard is discussed in depth elsewhere [14, 31]. More recent work on probabilistic reasoning in CAMAL [16], and subsequent development, has seen the metrics associated with the BDI and motivational constructs improved in the research reported here. The following sections detail how the reasoning over these fundamental control states is achieved. All examples given are actual output from one of the robot experiments described later in the paper. Hence,

- Beliefs are of the form *belief(Semantic Content, Source, Time, Belief Status)*. Semantic Content is a proposition (using predicates of arity 0 or more) denoting a possible state of affairs, and is held to be true according to its Belief Status – a probabilistic affect value in the range $\mathfrak{R}[0,1]$. Belief status varies across different perceptual sensors, deduction models and assumption sources. Time denotes the cognitive BDI cycle when the belief was first derived. The three following statements give beliefs of each source type
  belief(sensor(sonar,range,4000), assumption, 0, 1.0)
  belief(identified(sonarbundle9, amigo-blue, 1, −1120, *2, 2172), perception, 14, 0.9)*
  belief(inspect(amigo-blue, 1944, −1116, −1, *224), deduction, 17, 0.9)*

The first is an assumption from the task model for the robot about the range of its sonar (in mm), and defined at initialisation (i.e. cycle is 0). The second is a statement that the perceptual processes have identified a cluster of sonar readings (sonarbundle9) as a specific robot (amigo-blue) from a specified location (x=1, y=−11120) with a polar reading(angle = 2 radians, range = 2172 mm). The third statement arises from deliberation over which sub-task to attend. Although this relates to the same object referred to in the perception belief, the relative pose is different as the perceiving robot has moved over the three intervening BDI cycles.

- Goals are of the form *goal(Descriptor, SuccessCondition, Importance)*, where Importance is a real value in the range $\mathfrak{R}[0, 1]$. Importance is managed by affective processes in motivator evaluation and goal revision. Descriptor and SuccessCondition both use propositions defined over the same predicates as the Semantic Content for Beliefs. For example
  goal(approach(corner),near(corner, RobotX, RobotY, Theta, Range),0.5625).

This defines a goal to approach an area previously identified as a corner and to return the relative pose (RobotX, RobotY, Theta, Range) once successfully completed.

- Associations are of the form *association(Name, ConditionList, Desire, Intention, Insistence)*, where Insistence, is a real value in the range $\mathfrak{R}[0, 1]$. Insistence is managed by the BDI affective processes, and is strengthened by goal completion using the reactive behaviours associated with the Intention component. For example *association(association145, conditions([focus(sonarobject43,corner,2113,-1454,-100,3834), pose(self,2113,-1454,-50)]), approach(corner),reactive(false, true, false, methodapproach),0.95).*

This defines the Belief-Goal-Intention mapping corresponding to the example goal given immediately above. The ConditionList can be an empty list or any combination of valid propositions fitting the Semantic Content associated with beliefs. Here it defines a condition that states if the current focus is an object detected by sonar and identified as a corner, with relative pose matching the current x and y of the robot pose, and the goal is to approach that object, then use a reactive behaviour methodapproach. This behaviour definition includes three Boolean sensor states informing the reactive sub-architecture not to use Bump Sensors, to use Sonar and not to use Camera for the behaviour.

- Motivators are of the form *motivator(Goal, Association, Time, Deterministic, Cycles, Intensity)*, where Intensity is a real value in the range $\mathfrak{R}[0, 1]$. Time gives the BDI cycle in which the motivator was generated. The Deterministic element determines the exit strategy for the behaviour modules that form the reactive component. If this is set to true then the behaviour should return as soon as it is successful or for a maximum of the Cycles element. The Cycles element gives the number of reactive cycles that the reactive component (Rb_x in Figure 1) should run for in pursuing the behaviour associated with this motivator. The example given below uses the goal and association examples given above, and attempts to run for 10 cycles and return at that point even if the behaviour is unsuccessful (Deterministic value set to false): *motivator(goal(approach(corner), near(corner, RobotX, RobotY, Theta, Range), 0.5625) association(association145, conditions([focus(sonarobject43, corner, 2113, −1454, −100, 3834),pose(self, 2113, −1454, −50)]), approach(corner),reactive(false, true, false, methodapproach), 0.95), 17, false, 10, 0.64667).*

The belief values for Belief Status, Goal Importance and Association Insistence are combined within the Intensity value for the motivational construct. The nature of this combination and the parameterisation of the deliberative aspects of the architecture are controlled through Norms.

4.3 Domain and Task Models

The architecture presented here allows the same robot to be used for multiple tasks in any one domain and again in a number of domains. Domains are generic environments that allow for a range of task types. A domain can be a type of internal environment, for example a human oriented space such as an office, corridor or, in due course, a more specialized type of environment such as kitchen, manufacturing or industrial space. Each type of environment can support multiple task models, for example mapping an unknown room, determining location, outline and type of objects, finding and tracking other robots *etc.* Any particular task model can be combined with other task models as a hierarchical BDI model.

The architecture is loaded with a set of default predicates covering how it will run. These are then modified according to the domain model, and the task model defined in a declaration file that is used to initialize a specific robot architecture. The declarations file not only includes a domain-task model but also the robot (or robot colony processor) definition. The robot definition is used to ensure at load time that of all possible robot behaviours, and associations using them, only those that are compatible with the robot sensor and processing capabilities are instantiated. This ensures not only can a robot be used for multiple tasks in a number of domains, but a particular domain-task model can be used, with some variation reflecting the robot configuration, across multiple robots.

The domain-task model defines sets of goal-association pairings. For simple or tightly focused tasks this might be just a set of root goal-association pairings. For more demanding or flexible tasks this will include not only the root pair but also other hierarchical goal-association pairs that focus on specific tasks. The task models, typically a set of three (Prolog) files defining goals, associations (or intentions) and declarations, are used to initialize the architecture for a specific robot-domain-task sub-configuration. The declarations include the predicate domain_bdi_model(Name, Goals, Associations) where Name defines a unique domain-task symbol (for example swarm-root, swarm-track1 *etc.*) and Goals and Associations are the names of files containing the Goal-Association pairings. The goal-association pairings can include context rules that denote the conditions in which that pairing is suitable for use and also specific conditions when control needs to be passed to an alternative goal-association pairing set. The architecture includes domain-task independent context switching rules for reasoning, goal and behaviour failures. These can be overridden by the domain-task model rules. Part of the distributed blackboard used in the architecture is responsible for monitoring the context switching. Central to this aspect of the blackboard is the predicate mblackboardcontrol(Mode, Root, Parent, Current, History), where Mode defines the BDI reasoning mode (as described in the following section), Root is the base domain_bdi_model Name, Parent is the previous or calling ruleset, Current the one being used and History reflects the sequence of rulesets previously used.

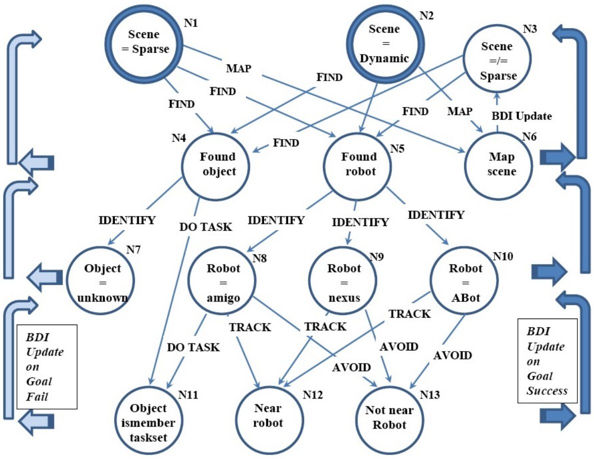


Figure 2: Example (simplified) SCARAB BDI Graph, showing alternative decision paths from two starting Belief positions (Double Circles N1 and N2) over four levels of the BDI model. Belief states shown in circles; Goal-Intention combinations as labelled arcs between Belief states. Arrows to left and right infer BDI revision upon earlier Belief states dependent upon Goal-Intention success or failure.

The goal and association files are paired and include examples fitting the definitions given above. A goal may be referenced in any number of associations with differing belief basis, allowing alternate behaviour paths for goal completion. As shown in the simplified SCARAB BDI graph in Figure 2, these BDI arcs can be easily represented as acyclic graphs. The figure shows a subset of the BDI cycle (four nodes deep) used in the mapping experiment described in section 7. The two concentric circles are starting Beliefs (the environment is dynamic and sparse) given at initialisation. As a result of one possible Goal-Intention arc, an alternative to Belief N1 (`Environment is sparse`) is shown by Belief N3 (`Environment is not sparse`); these would be examples of `p` and `¬p` in the BDI update models given in section 4.1. Goal-Intention arcs are labelled with the Intention, the terminating node represents the Goal (i.e. Belief predicates that need to be made true or false). For example, the alternative goals from node N8 (Robot found at N5 and Identified as an AmigoBot at N8). Dependent upon the preferences set for the BDI model, the strengths of the Beliefs, Goals and Intentions the robot can choose between three different goals (represented as nodes N11, N12 and N13) with associated Goal-Intention arcs (respectively, labelled as DO-TASK, TRACK and AVOID). The arrows to the left and right of the sub-graph represent the effect of a BDI update after goal success or failure.

As the experiment progresses more of the BDI graph becomes instantiated; however with the proviso that Beliefs can be erased according to the BDI norm describing belief decay. For example norm(belief,belief_decay_threshold,25), where the 25 represents the number of BDI cycles a belief can be held as true (to the extent of its associated probability). Further examples of norms governing the BDI model are given in the following section.

Task models are currently manually built and then loaded at run-time. The roboCAMAL sub-project reported in [15] made use of reinforcement learning to modify task models; current implementations lack this feature, although it is part of the sub-project described in section 6. The task model used can vary over the lifetime of a CAMAL experiment as a result of goal and intention evaluation. Currently the architecture does not support any structural learning. The models can be time consuming to build and flawed in their specification, even with bespoke logic checking software. Furthermore they do not fully support multi-robot ecologies. As this research progresses further into ambient environments supporting multiple robot ecologies, this needs to be addressed, taking into consideration the research into ontological task and domain models, for example [3, 27].

4.4 Controlling BDI and Reasoning

The architecture, BDI and motivational models are configured at run-time using domain models that allow the architecture to instantiate itself in a specific configuration of robot, domain and tasks. For any particular configuration of the architecture on a specific robot, there is a default behaviour, for example obstacle avoidance. This default behaviour (or possibly a set of default behaviours) is specified in the domain-task model and can be set to an alternative (set of behaviours) by the deliberative architecture – an example of many interactions taking place between the various layers of CAMAL. For a specific domain model, there are several behaviours associated with it; which represent intentions in the BDI model, prioritized and ranked by the affordances set by the affect and norm models. The BDI and control processes interact through the use of affect and affordances.

Norms are used to control selection and management of BDI. By default the architecture looks to perform new actions. A new action here is defined in terms of a unique belief-desire-intention arc instantiation. However it has been found that in some situations, for example a constrained robot environment with a set number of objects and dynamic agents, that this can be counter – productive and lead to no viable BDI combination that is new. The BDI generation and selection mechanisms catch this (via meta-reasoning) and change the norm for BDI generation and selection from generative to regressive mode, allowing previous BDI arcs to be chosen. For example finding another robot that was found but lost some time ago. Alternatively, specific behaviour modules may generate a set of beliefs causing the architecture to pursue a preferred direction.

The affect system and reasoning processes associate their actions with the difference between the current and desired state of the motivator. In all cases, these metrics can change over time as the robot adjusts to the world it inhabits and the domain model loaded. These modifications are the result of a parameterized evaluation processes and reasoning models. The parameters and current reasoning modality is defined through the use of Norms as selected by default or modified by the Meta-Control or BDI Feedback Adaption processes. Motivator evaluation modifies the values associated with Goals and Associations based on the success or failure of the motivator, and the current Norms controlling the BDI. The adaption parameter, as explored in the experiments presented in [15], is a real valued number in the range [0, 1].

- Norms to ascribe initial metrics to Belief:
  - Source – assumption, deduction, perception (variety). These nominal tokens are associated with an affordance mapping given in the initialisation parameters. These can be changed, depending upon the current BDI set, so that probability values associated with a belief source can change at run-time.
- Norms to control Belief Updating
  - Belief decay, beliefs can be managed in a number of ways;
  - Belief deletion (transfer to memory) so ¬p and p cannot co-exist in current belief set (logic tautology disallowed). The preferred belief selected, by default, upon recency (newer beliefs) but the BDI model can be modified to prefer belief importance, or exception models for specific predicates;
  - Belief models associated with specific task models and BDI rulesets.
- Norms to control BDI are:
  - Akin to conflict set resolution in rule based systems, for example recency, specificity, redundancy *etc.*
  - BDI chosen on a selection mode (currently five) across the affect values associated with beliefs, desires and intentions. Variations on this allow for recently generated, historically generated or recurrent (time-based) BDI arcs to be preferred.

5 Robotic Platforms and Architecture Implementation

Previous robot systems such as roboCAMAL [15] used the generic architecture tailored to a specific robot experiment. In the current work a more general architecture (as seen in Figure 1) has been developed from earlier simulation work that allows alternative task models to be loaded that configures the architecture for a specific robot or robot colony and its duties. Figure 3 shows the information and control flow paths for the major modules for the implementation of the design in Figure 1 for the suite of robots shown in Figure 4.

The current research looks towards building a distributed model of cognition, with the organization of the robot swarm adapting to the needs of the colony, the demands of the current task and the nature of the environment. Not all the robotic platforms used have the same sensor or onboard processing capabilities (see image in Figure 4). All robots are fitted with sonar, although the number and type vary across the different robot platforms. A Pioneer and five AmigoBots (see

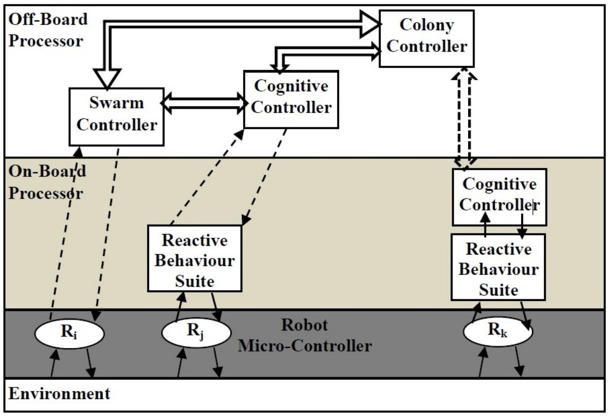


Figure 3: Architecture Schematic with more sophisticated robots to the right of figure. Single arrows represent control and information flow. Walled arrows represent information flow. Dashed lines represent communication over the network. Solid lines represent communication within a processor.

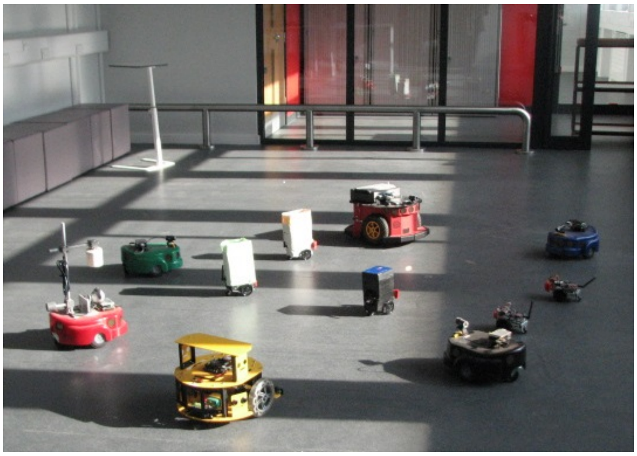


Figure 4: Part of the full complement of the robots used in the developments described here.

Adept MobileRobots [60]) are fitted with cameras so they are capable of supporting visual perception. One of the AmigoBots (as discussed in [15]) makes use of an omni-directional vision system; the others use directed cameras. A powerful processor (Matrix 5i) is used for the larger (Pioneer 3-DX) robot, enabling it to be developed as a mobile vision processing centre (and shown schematically as the configuration associated with robot R_k in Figure 3). A further off-robot system is being developed for coordination (the Colony Controller to the top of Figure 3). A less powerful and smaller set of robots form part of the larger colony. Some of these ActivityBots (see [61]) are equipped with Raspberry Pi with an associated Pi camera.

All the AmigoBots and a number of ActivityBots have an onboard Raspberry Pi for local control (and these are shown schematically as the configuration associated with robot R_j in figure 3); other ActivityBots have no such onboard computer and rely on their robot microcontroller for limited local control under the direction of an external controller (as indicated by the configuration associated with robot R_i in figure 3). Communication across the robot colony enables even the simplest of the robots to change the focus of their onboard processing and alter their current behaviour. The full control architecture can contain sub-colonies (swarms), such as a number of ActivityBots which can be used for fast exploration of an environment to establish the presence of objects of potential interest. Indeed some of the research makes use of small colonies of AmigoBots or ActivityBots running a very reduced SCARAB architecture; typically Braitenberg reactive Behaviours plus sensors and a limited form of the Belief system introduced above (as described in section 6 below). Hence a wide range of behaviour capabilities and control systems ensue across the robot colony. The robots can be situated in any number of indoor environments; the latest of these hosts an environmental camera system with its own dedicated PC to enable localisation of objects and robots. All robots, embedded computers, and off-board computational devices can communicate with each other over a dedicated wireless network as shown in the central part of Figure 1 (and in Figure 3).

The control architecture for any robot (with sufficient on-board processing power) can be localised or distributed (see Figures 1 and 3); comprising of a cognitive control architecture (implemented in swi-prolog), and a local behaviour control. The behaviour control architecture for those robots with a raspberry pi is implemented in a networked C++ or C module that interacts locally or remotely with the immediate and remote cognitive processes.

6 Collaborative Tasks and Shared Information using swarms

Experiments with part of the architecture depicted in Figures 1 and 3 involved the use of five ActivityBots [61], seen in Figure 4, with a limited set of Braitenberg [62] behaviours to investigate the use of adaptive swarm topologies. The robots were designed with four Braitenberg behaviours activated according to their affective (non-cognitive) state; nominally that of curiosity. The project is to be described in full elsewhere (Shylaja

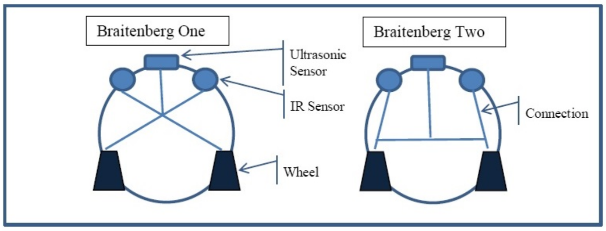


Figure 5: The two Braitenberg Vehicles used as designs for the behaviours of the ActivityBots showing alternative connection topologies.

and Davis, submitted for publication); here an overview is presented to show how aspects of the CAMAL architecture map onto colonies of less complex robots.

Here, curiosity is defined as a psychological construct [63, 64] that leads to exploratory behaviour. Curiosity is a motivated (to be desired) cognitive state that links learning and explorative behaviours in biological systems. Curiosity arises when there is lack of coherence across cognitive processes about the origin of some perception, thought, belief or desire. It drives animals to explore and find new knowledge that can bridge the gap of non-coherence. There are a number of other projects that have investigated the use of this motivational state [65, 66]; here the deliberate adoption of a constrained behaviour set is used to investigate a minimal configuration of the CAMAL architecture.

Five ActivityBot robots were designed as Braitenberg vehicles, each with two forward facing Infra-Red sensors (left and right) with a central Ultrasonic sensor (as illustrated in Figure 5). Two connection topologies were used as shown: crossed (Braitenberg One) and parallel (Braitenberg Two) but with the connection modulated by the signal from the central Ultrasonic sensor.

This arrangement plus the use of inverse and proportional function on the signal connectors gives rise to four behaviours (plus stationary or rest):

- B-MTAC MoveToAndCollide Cross connection, Proportion function
- B-MTAS MoveTowardsAndStop Parallel connection, Inverse function
- B-MAAS MoveAwayAndSlow Parallel connection, Proportion function
- B-MAAO MoveAwayAndOscillate Cross connection, Inverse function

At the level above the Braintenberg behaviours, a four state Finite State Automata is used to determine which of the behaviours is active. The principal drive (or state) around which others are closely linked is Curiosity; commonly defined as “a quality related to inquisitive thinking such as exploration, investigation, and learning” [67]. Here it is defined in terms of the swarm and the individuals that comprise the swarm as a cognitive state that makes an individual explore its capabilities to address a given situation or problem (i.e. an inquisitive approach to goal-directed behaviour). The other three drive states are Normal; Rest; and LowMood (or Frustration or Boredom). The robots switch from one state to another based on their internal parameters and external events. These events map onto three real valued metrics which form the basis for state changes: Number of attacks; Energy-level; Number of rest cycle.

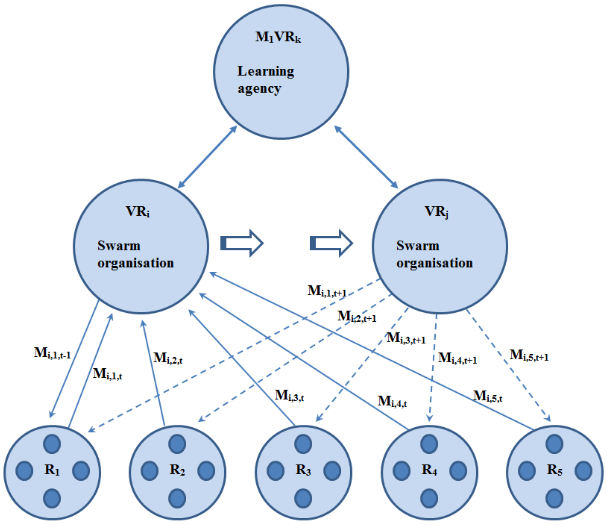


Figure 6: Communication across three levels of a swarm topology, with a swarm consisting of five robots (R_1, R_2, R_3, R_4, R_5; each in one of four possible states). The second level (the VR_i and VR_j agency) is the swarm organisation; the third level (the M_1VR_k agency) is the learning framework for the swarm.

In figure 6, the swarm of five robots (R1, R2, R3, R4, R5) is shown connected via communication to a first level meta-agency (for example VirtualRobot i or VR_i). Each of these robots is in one of the four possible states (Curiosity, Rest *etc.*) described above. The swarm organisation changes over time as the robots and the macro-swarm (first level) agency modify their states. In the figure just two of the possible swarm topology organisations are shown (VR_i and VR_j). The third level shows one meta-level macro-swarm agency (M_1VR_k); the learning agency for the swarm. Other meta-level agencies are possible; see [31] or [68] for further reading. Any message passing between any two levels is time stamped;

the accumulation of these messages becomes part of the swarm's history (or memory).

Messages between the robots and the first-level swarm organisation (VR_i) are defined as:

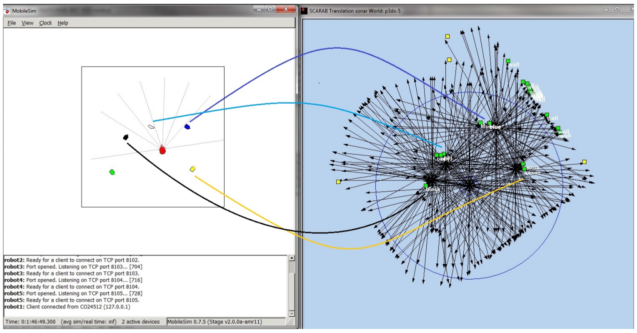
$M_{i,r,t}$:- <$B_{i,r,t}$, $S_{a,r,t}$>,

where $M_{i,r,t}$ is message M_i from robot r at time-stamp t defined over behaviour valence B_i (with B_i defined within the interval $\Re[0.0,1.0]$) of robot r at time-stamp t with the robot in one of four states as defined by state S_a of robot r at time-stamp t.

The specifically chosen number of robots within the swarm allows the swarm topology organisation (VR_i) to determine a mode state for the swarm ($S_{a,t}$); plus an average Behaviour valency ($B_{i,t}$). These together with the reported environmental state Env_t are then compared to a "goal-swarm history" profile to determine corrections to the robots; i.e. the message from the swarm organisation to each robot as $M_{i,r,t+1}$:- <$B_{i,r,t+1}$, $S_{a,r,t+1}$>. The meta-level learning agency uses the "goal-swarm history" to find optimal behavioural changes for the swarm as a whole and each individual within the swarm in order to further the goals of the swarm. This work is ongoing but initial designs and implementations have shown interesting results. With the return of this paper's co-author to her native country, there is now the opportunity to investigate this approach further using different robot types as well as the development of the original swarm as part of the larger CAMAL project (and then form the vertical coupling to the right in Figure 3). This research becomes part of the larger investigation into the veracity, validity and verification of perceptual data as it affects a distributed model of cognition which supports multi-modal BDI models.

7 Sonar Mapping of an environment and its objects

One of the development testbeds used in moving from a bespoke simulation to physical robot control is the MobileSim testbed [60]. This simulation allows control and communication of selected robots in a simple environment (such as shown to the left in Figure 7) or more complex maps to be loaded which reflect the intended robot environment. Control of one or more of the robots in the environment is via Aria [60] calls to the robot microcontroller. In the current work a suite of parameterised behaviours (written in C++) were used within the Aria framework (at the design level, the Reactive Behaviour Suite referred to in Figure 3) and reflect the behaviours R_i, R_j *etc.* referred to in the lower part of Figure 1. A listening cycle within the Aria software awaits communication from higher level control architecture (written in swi-prolog) and selects the appropriately parameterised behaviour (as given in the calling message). Once the reactive behaviour has completed (or failed) a returning message is sent. Communication between these software packages can be via cable or wireless network. A full implementation of this has been tested with a suite of robots but for simplicity a task using the simulation software is described here.



```
robot2: Ready for a client to connect on TCP port 8102.
robot3: Port opened. Listening on TCP port 8103... [704]
robot3: Ready for a client to connect on TCP port 8103.
robot4: Port opened. Listening on TCP port 8104... [716]
robot4: Ready for a client to connect on TCP port 8104.
robot5: Port opened. Listening on TCP port 8105... [728]
robot5: Ready for a client to connect on TCP port 8105.
robot1: Client connected from CO24512 (127.0.0.1)
Time: 0:1:46:49.300  (avg sim/real time: inf)  2 active devices   MobileSim 0.7.5 (Stage v2.0.0a-amr11)
```

Figure 7: Sonar map from use of SCARAB control of Pioneer 3-DX in mobilesim. Colour coded arcs represent a mapping from objects located to their position in the simulator. The sonar map is rotated as it reflects the orientation of the robot at the start of the simulation run.

The left image in Figure 7 shows a simple uncluttered world with one object, four AmigoBots (yellow, blue, black and green) and the principal controlled robot, a Pioneer 3-DX [69], at the start of the run. Over 15 BDI cycles the Pioneer 3-DX has identified, approached and mapped the convex hull of the object, three of the other robots and one of the opaque walls (seen as green squares in the right image of Figure 7 – the yellow squares represent objects as yet not mapped). The ultrasonic beams, from one time stamp, are marked in the left image; the right image shows the ultrasonic mapping of all sonar readings taken over the short 15 BDI cycle run (the simulation has run for 1000's of BDI cycles; the physical robots for several hundred).

The SCARAB generated right image (the fifth one generated by the robot over the 15 cycles) clearly shows five main loci for the ultrasonic beams. The starting position is the central lower position. The other four loci relate to the approach and sonar mapping of each of the objects identified and mapped. The green AmigoBot has been located and the empty space surrounded by longer arcs to the lower left in the right image reflects its position. However as the controlled robot has yet to approach and model its convex hull it is not labelled in the image.

The task model used in this experiment is a variation on that given in Figure 2; except instead of just tracking any other robot, the controlled robot is tasked with exploring the boundaries of its environment, locating objects, identifying them and if possible mapping their complete convex hull using sonar. The human user is required to confirm the identity of the found objects as one from a set of known objects (or as something new) as no machine vision is used in the simulator. High level norms provide the criteria for selecting which object to investigate; those being nearest, then new, then unknown aspects of a possibly known object. On approaching an object the controlled robot determines the extremities of that object and positions itself parallel to a straight arc half way between the found extremities. It then moves clockwise or anti-clockwise keeping itself approximately the same distance away from the object as given by one of the side ultrasonic sensors. If the object moves or the pathway to continue mapping is blocked the robot completes this mapping task at that point. Otherwise the controlled robot navigates completely around the object being inspected to give a full ultrasonic convex hull of the object. These behaviours are managed by the BDI control layer (in a tightly bound cycle) with the ARIA level behaviours returning pose and ultrasonic readings to the BDI layer. If a sequence of behaviours fail, the meta-level architecture is activated to parameterise the BDI so alternative BDI sequences are followed (as explained in section 4.1). In a full CAMAL architecture with multiple controlled robots the robot-centric sonar maps of individual robots are relayed to the abstract agency (top of Figure 1) for it to reason about and so provide an absolute sonar map that can be shared across the robot colony. The top-level agency must reason about how to resolve differences across individual and combined ultrasonic maps. How it does that is ongoing research but analogous to previous work on landmark identification in medical images [70].

8 Current Limitations

In moving from our own simulations to MobileSim to real robots a number of issues have arisen. Many of these will be well-known to robot practitioners such as noisy data from sensors, the need for robust and fast vision processing, the issues surrounding pose and localisation; robot parts failing, limited battery life *etc.* Some of these are being currently addressed; other issues require that the architecture be rethought in terms of the deliberative-reactive-motor mapping associated with any specific BDI arc. While the research project, in its broadest sense, attempts to remain true to the original aim of developing a (now distributed) model of cognition that will generalise across tasks, robot platforms and environments, specific sub-projects, and investigations, sometimes we sacrifice generality for efficiency and timeliness of implementation. On completion, the sub-project is then mapped back to the over-arching project to allow generality at the expense of efficiency. As Wei and Hindriks [71] found this can be time-consuming and ultimately needs to incorporate learning mechanisms. Their major stated obstacle would appear to be the veracity of information arising from perceptual (vision) systems. This issue is one that, as described earlier, has no easy solution if the architecture is to work well in a multitude of scenarios.

Visual perception in simulation work can be shallow, in fact in many cases non-existent. However for physical robots, it is a useful source of perceptual information, and when working efficiently a relatively reliable means of locating and identifying objects. In earlier implementations [15], a reduced colour-coded environment was used for an investigation in symbol anchoring based on visual information. That research was limited to the control of a single robot and three objects of interest (blue ball, black robot and red robot). The (omni-directional) vision system used there is at best incomplete, at worst inadequate, for the current work.

Initial investigations into the use of 2.5D images (using Kinects) proved troublesome. Only the larger Pioneer 3-DX robot (plus some other teaching robots not used in this research) had the computational power to use the Kinect. The smaller AmigoBots could carry the Kinect but the Raspberry Pis used on these robots proved computationally inadequate. The smaller ActivityBots could not carry such a camera and lacked the processing capability to run the devices. For the smaller robots carrying a Raspberry Pi, a Pi camera was installed. For the Pioneer 3-DX and AmigoBots relatively cheap webcams (as seen in Figure 4) were installed. As the initial work required only that the suite of robots be identified and other objects were to be classed as unknown (but still mapped using sonar), a variety of algorithms from OpenCV were used for blob matching using HSV and RGB profiles acquired from trials with one robot sensing the other robots in a variety of situations and illumination levels. The principal investigator has sufficient experience in machine vision to develop these algorithms but unfortunately time constraints meant that many of the problems associated with developing existing (or new) algorithms for this application remain unsolved.

Figures 8 and 9 illustrate the current state of development of the vision system. Both figures were

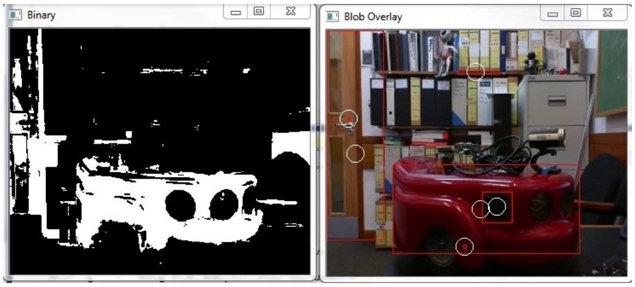


Figure 8: The left image is binary version arising from HSV and RGB segmentation of the original. The right image is original with blobs selected from binary segmentation overlaid. In this instance there is relative good identification of the main object of interest (the red AmigoBot)

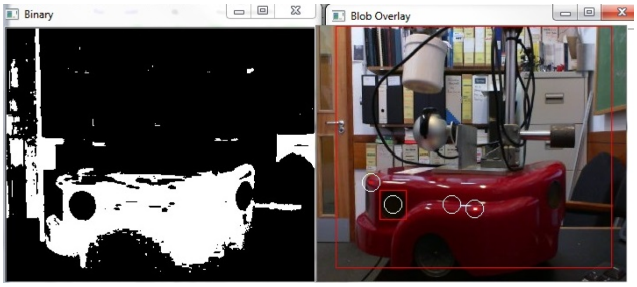


Figure 9: The left image is binary version arising from HSV and RGB segmentation of the original. The right image is original with blobs selected from binary overlaid. In this instance there is relative poor identification of the main object of interest (the red AmigoBot with the omnidirectional camera)

taken with the same camera under identical illumination levels. Figure 8 shows a red AmigoBot with minimal surface infrastructure. In this instance there is relative good identification of the main object of interest (the red AmigoBot). Figure 9 shows a red AmigoBot with more significant surface infrastructure supporting an omnidirectional vision system. In this instance there is relative poor identification of the main object of interest (the red AmigoBot with the omnidirectional camera). Due to time constraints a decision was taken to pursue the sonar mapping described above as spatial reasoning is deemed more important for the robot colony than object identification via visual means. Spatial reasoning over sonar images, and the correlation of robot location and found objects from the individual maps (as described above) was deemed to be of greater interest. The research, for the time being, sidesteps this issue by using other means to identify robots, such as colour coded LEDs or Wi-Fi tags. Alternatively, we can 'cheat' on this issue if required by labelling the robots so the robot lab ambient vision system in the recently installed laboratory can identify the robots. Meanwhile a further sub-project [72] is approaching the vision system by taking the ideas from earlier work on anchoring and applying them to robot colonies. The, as yet unresolved, visual perception problems will be addressed when the full robot colony architecture has progressed sufficiently with mechanisms for exacting the veracity and validation of perceptual data and beliefs.

There are a number of other failings in the current architecture, beyond those listed above. For example, experimental studies that explore the effect of different task models across a controlled set of environments, the effect of varying how intentions are modelled, and the lack of any structural learning mechanisms. A number of these shortcomings will be addressed when a new robotic laboratory (as seen in figure 4) comes on-line.

9 Conclusions

The principal message that the movement of this research into cognitive agencies from bespoke to generally available simulations to physical robots is that it is (unexpectedly, sic) troublesome and time consuming. What is relatively trivial in simulation work, such as how to assess the quality of sensor data in perceptual processes, becomes a major sub-project in its own right. It is also tempting at times to sacrifice generality of process and knowledge for engineering expediency. This may not be an issue within any one sub-project, but the mapping of the results back to the parent project, and regain the generality of process and knowledge, is not a task undertaken lightly, as many of the references cited throughout the paper also state.

It is unclear how far closer the field of cognitive (and behavioural) robotics is to the goal of a generalised robotic intelligence as envisaged ten years ago by Levesque and Lakemeyer [1] and reviewed by Duch *et al.* [73]. As noted by Al-Razgan *et al.* [13] in their systematic review of robotics, the field of robotics research has become fragmented. The same can be said of the wider domain of cognitive architectures as noted in the recent survey of the research field [74]. Funding models for research in the domain require that specific engineering solutions to problems be found. If the focus is on assisted living, for example Beetz *et al.* [75], or the use of cognitive robots in industrial settings, for example Pedersen *et al.* [12], then the research output reflects the engineering of solutions appropriate to that funding stream. Alternatively it may seem that some particular topic, for example the computational modelling of emotion [76] as described earlier and reviewed in Robinson and El Kaliouby [77], finds favour. This is to be expected but it is at the expense of more challenging research directions such as the use of imagery in reasoning [78] or the establishment of truly generic tests of robot ability and generality.

Acknowledgement: Past funding for the work presented here has come (in ascending chronological order) from University of Birmingham Vice-Chancellor Grant (Architectures for Intelligent Agents), Japanese Society For Promotion of Science (Affective Agents), University of Hull Alumni Development Fund (Pioneer 3-DX Robot), University of Hull PhD studentships (S.J. Lewis, J. Gwatkin, M.V. Vijayakumar, H. Miri) and Government of India visiting scholar award to Shylaja Kanaganapalli Ramulu for her research at the University of Hull.

Author Contributions

The first author is responsible for the research (including theoretical framework, designs and implementations) presented in this paper. The co-author is responsible for design and implementations of part of the work presented in the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] H. Levesque, G. Lakemeyer, Cognitive Robotics, In F. van Harmelen, V. Lifschitz, B. Porter (Eds.), Handbook of Knowledge Representation, Elsevier, 2008

[2] M. Lee, U. Nehmzow, M. Rodrigues, Towards cognitive robotics: robotics, biology and developmental psychology, In D. McFarland, K. Stenning, M. McGonigle, D. Hendry (Eds.), The complex mind: an interdisciplinary approach, Basingstoke, Palgrave Macmillan, 2012, 103-126

[3] M. Tenorth, M. Beetz, KnowRob: A knowledge processing infrastructure for cognition-enabled robots, International Journal of Robotics Research, 32(5) (2013), 566-590

[4] S. Karapinar, S. Sariel, Cognitive robots learning failure contexts through real-world experimentation, Autonomous Robots, 39(4) (2015), 469-485

[5] Y. Wang, On Cognitive Robotics and Theories of Abstract Intelligence, 3rd International Conference on Automatic Control, Soft Computing & Human-Machine Interaction (ASME '15), Salerno, Italy, June 27-29, 2015

[6] Y. Wang, Abstract intelligence and cognitive robots. Journal of Behavioural Robotics, 1(1) (2010), 66-72

[7] P. Haazebroek, S. van Dantzig, B. Hommel, A computational model of perception and action for cognitive robotics, Cognitive Processing, 12(4) (2011), 355-365

[8] R. Bemelmans, G. J. Gelderblom, P. Jonker, L. de Witte, Socially Assistive Robots in Elderly Care: A Systematic Review into Effects and Effectiveness, Journal of the American Medical Directors Association, 13(2) (2012), 114-120

[9] Z. Kowalczuk, M. Czubenko, Intelligent decision-making system for autonomous robots, International Journal of Applied Mathematics and Computer Science, 21(4) (2011), 671-684

[10] R. Li, B. Liu, K. D. McDonald-Maier, Cognitive assisted living ambient system: a survey, Digital Communications and Networks, 1(4) (2015), 229-252

[11] A. Widyotriatmo, K.-S. Hong, Configuration Control of an Autonomous Vehicle under Nonholonomic and Field-of-View Constraints, International Journal of Imaging and Robotics, 15(3) (2015), 126-139

[12] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, Ole Madsen, Robot skills for manufacturing: From concept to industrial deployment, Robotics and Computer-Integrated Manufacturing, 37 (2016), 282-291

[13] M. Al-Razgan, L. F. Alfallaj, N. S. Alsarhani, H. W. Alomair, Systematic Review of Robotics Use Since 2005, International Journal of Mechanical Engineering and Robotics Research, 5(2) (2016), 129-132

[14] D. N. Davis, Control States and Complete Agent Architectures. Computational Intelligence, 17(4) (2001), 621-650

[15] D. N. Davis, J. Gwatkin, RoboCAMAL: A BDI Motivational Robot, Journal of Behavioural Robotics, 1(2) (2010), 116-129

[16] D. N. Davis, H. Miri, Probabilistic BDI in a Cognitive Robot Architecture, International Journal of Computer Science and Artificial Intelligence, 2(3) (2012), 1-10

[17] A. Chella, U. Kurup, J. Laird, G. Trafton, J. Vinokurov, B. Chandrasekaran, The challenge of robotics for cognitive architectures, In Proceedings of the International Conference on Cognitive Modeling, Ottawa, ON, Canada, July 11-14, 2013, 287-290

[18] H. Levesque, R. Reiter, High-level robotic control: Beyond planning, AAAI Spring Symposium on Integrating Robotics Research, Working notes, Palo Alto, CA, March 1998

[19] P. Langley, J. E. Laird, S. Rogers, Cognitive architectures: Research issues and challenges, Cognitive Systems Research, 10 (2009), 141-160

[20] M. L. Minsky, The Society of Mind, William Heinemann Ltd., 1987

[21] A. S. Boza, R. H. Guerra, A. Gajate, Artificial cognitive control system based on the shared circuits model of sociocognitive capacities. A first approach, Engineering Applications of Artificial Intelligence, 24(2) (2011), 209-219

[22] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, *et al.*, Autonomous mental development by robots and animals, Science, 291 (2001), 599-600

[23] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, *et al.*, Cognitive developmental robotics: a survey, IEEE Transactions on Autonomous Mental Development, 1(1) (2009), 12-34

[24] N. Friedman, Learning Belief Networks in the Presence of Missing Values and Hidden Variables, Proceedings of the Fourteenth International Conference on Machine Learning, July 8-12, 1997, 125-133

[25] M. M. Rahman, D. N. Davis, Machine Learning Based Missing Value Imputation Method for Clinical Datasets, Chapter 19 In G.-C. Yang, S.-L. Ao, L. Gelman (Eds.), IAENG Transactions on Engineering Technologies, ISBN 978-94-007-6190-2, Springer, 2013

[26] P. De Bourcier, M. Wheeler, The Truth Is Out There: the Evolution of Reliability in Aggressive Communication Systems, Proceedings of the Fourth European Conference on Artificial Life, 1997, 1-10

[27] S. Lallee, P. F. M. J. Verschure, How? Why? What? Where? When? Who? Grounding Ontology in the Actions of a Situated Social Agent, Robotics, 4 (2015), 169-193, DOI:10.3390/robotics4020169

[28] P. F. M. J. Verschure, C. M. A. Pennartz, G. Pezzulo, The why, what, where, when and how of goal-directed choice: neuronal and computational principles, Philos. Trans. R. Soc. Lond B Biol. Sci., 369: 20130483 (2014), http://dx.doi.org/10.1098/rstb.2013.0483

[29] H. A. Simon, Motivational and emotional controls of cognition, Reprinted in Models of Thought, Yale University Press, 1979, 29-38 (originally Psychological Review, 74(1) (1967), 29-39)

[30] D. A. Norman, Twelve issues for cognitive science, Cognitive Science, 4 (1980), 1-33

[31] D. N. Davis, Cognitive Architectures for Affect and Motivation, Cognitive Computation, 2(3) (2010), 199-216

[32] R. Plutchik, The Psychology and Biology of Emotion, Harper Collins, New York, 1994

[33] R. Plutchik, The Nature of Emotions Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice, American Scientist, 89(4) (2001), 344-350

[34] P. Ekman, The Nature of Emotion: Fundamental Questions, Oxford University Press, New York, 1994

[35] K. Scherer, Appraisal Considered as a Process of Multilevel Sequential Checking, In K. Scherer, A. Schorr, T. Johnstone (Eds.), Appraisal Processes in Emotion, Oxford University Press, New York, 2001

[36] K. Oatley, Best Laid Schemes, Cambridge University Press, 1992

[37] E. T. Rolls, The Brain and Emotion, Oxford University Press, 1999

[38] R. C. Arkin, Moving Up the Food Chain: Motivation and Emotion in Behaviour-based Robots, In J.-M. Fellous, M. A. Arbib (Eds.), Who Needs Emotions? The Brain Meets the Robot, Oxford University Press, 2005

[39] R. Sanz, J. Gómez, Vindication of a Rigorous Cognitive Science, Journal of Mind Theory, 0(1) (2008), 5-9

[40] E. D. Sontag, Mathematical Control Theory: Deterministic Finite Dimensional Systems, Second Edition, Springer, New York, 1998

[41] R.-E. Precup, H.-I. Filip, M.-B. Radac, E. M. Petriu, S. Preitl, C.-A. Dragos, Online identification of evolving Takagi–Sugeno–Kang fuzzy models for crane systems, Applied Soft Computing, 24 (2014), 1155-1163

[42] J. Vascak, K. Hirota, Integrated decision-making system for robot soccer, Journal of Advanced Computational Intelligence and Intelligent Informatics, 15(2) (2011), 156-163

[43] J. Vascak, M. Pal'a, Adaptation of Fuzzy Cognitive Maps for Navigation, Purposes by Migration Algorithms, International Journal of Artificial Intelligence, 8(12) (2012), 20-37

[44] N. Hawes, A survey of motivation frameworks for intelligent systems, Artificial Intelligence, 175(5-6) (2011), 1020-1036

[45] O. Holland, A. Diamond, H. G. Marques, B. Mitra, D. Devereux, Real and apparent biological inspiration in cognitive architectures, Biologically Inspired Cognitive Architectures, 3 (2013), 105-116

[46] GC5, UK Computing Research Committee Grand Challenge 5 (GC-5): Architecture of Brain and Mind, University of Birmingham website, 2008, http://www.cs.bham.ac.uk/research/cogaff/gc/

[47] D. Kahneman, A. Tversky, Prospect theory: An analysis of decision under risk, Econometrica, XLVII (1979), 263-91

[48] M. Quirin, J. Hertzberg, J. Kuhl, A. Stephan, Could positive affect help engineer robot control systems?, Cognitive Processing, 12(4) (2011), 375-378

[49] R. Sanz, M. G. Sánchez-Escribano, C. H. Pérez, A model of emotion as patterned metacontrol, Biologically Inspired Cognitive Architectures, 4 (2013), 79-97

[50] A. Newell, Unified Theories of Cognition, Harvard University Press, 1990

[51] A. Sloman, The Mind as a Control System. In C. Hookway, D. Peterson (Eds.), Philosophy and the Cognitive Sciences, Cambridge University Press, 1993, 69-110

[52] D. N. Davis, Reactive and Motivational Agents, In J. P. Muller, M. J. Wooldridge, N. R. Jennings (Eds.), Intelligent Agents III: Agent Theories, Architectures, and Languages, Springer Verlag, 1996

[53] D. N. Davis, S. J. Lewis, Computational Models of Emotion for Autonomy and Reasoning, Informatica, Special Edition on Perception and Emotion Based Reasoning, 27(2) (2003), 159-165

[54] G. Bourgne, Affect-Based Multi-Agent Architecture for a 5-aside Football Simulation. M.Sc. Thesis, Department of Computer Science, University of Hull, UK, 2003, http://blacklight.hull.ac.uk/catalogue/b1858572

[55] MobileSim, Adept Mobile Robots (2015), http://robots.mobilerobots.com/wiki/MobileSim (Last accessed July 2017)

[56] A. S. Rao, A. P. Georgeff, BDI Agents: From Theory to Practice, Technical Note 56, Australian Artificial Intelligence Institute, April 1995, http://www.agent.ai/doc/upload/200302/rao95.pdf

[57] K. Bartsch, H. Wellman, Young children's attribution of action to beliefs and desires, Child Development, 60 (1989), 946-964

[58] S. Wahl, H. Spada, Children's Reasoning about Intentions, Beliefs, and Behaviour, Cognitive Science Quarterly, 1 (2000), 5-34

[59] B. J. Baars, In the Theater of Consciousness, Oxford University Press, 1997

[60] Adept MobileRobots, http://www.mobilerobots.com/ResearchRobots.aspx, OMRON ADEPT MOBILEROBOTS LLC 2016 (Last accessed July 2017)

[61] Parallax, ActivityBot Robot Kit, https://www.parallax.com/product/32500, Parallax Inc., 2017

[62] V. Braitenberg, Vehicles: Experiments in synthetic psychology, Cambridge, MA: MIT Press, 1984

[63] D. E. Berlyne, A theory of human curiosity, British Journal of Psychology, 45(3) (1954), 180–191

[64] A. Manguel, Curiosity. New Haven, CT: Yale University Press, 2015, ISBN 978-0-300-18478-5

[65] L. Macedo, A. Cardoso, The exploration of unknown environments populated with entities by a surprise-curiosity-based agent, Cognitive Systems Research, 19-20 (2012), 62–87

[66] J. Schmidhuber, Developmental Robotics, Optimal Artificial Curiosity, Creativity, Music, and the Fine Arts, Connection Science, 18(2) (2006), 173-187

[67] Wiki, 2017, Wikipedia: https://en.wikipedia.org/wiki/Curiousity (Last Accessed July 2017)

[68] M. V. Vijayakumar, D. N. Davis, K. R. Shylaja, Design of Metacontrol and Metacognition mechanisms in SMCA Using Norms and Affect rules, International Conference on Artificial Intelligence and Pattern Recognition (AIPR-09), Orlando, FL, USA,

July 13-16, 2009

[69] Pioneer 3-DX, OMRON ADEPT MOBILEROBOTS LLC 2016 http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx (Last accessed 30 August 2017)

[70] D. Forsyth, D. N. Davis, Automated Cephalometric Analysis. European Journal Of Orthodontics, 18(1) (1996), 471-478

[71] C. Wei, K. V. Hindriks, An Agent-Based Cognitive Robot Architecture, In Programming Multi-Agent Systems, part of the Lecture Notes in Computer Science, 7837 (2015), 54-71

[72] S. Balding, D. N. Davis, Combining depth and intensity images to produce enhanced object detection for use in a robotic colony, 18th Towards Autonomous Robotic Systems (TAROS) Conference, University of Surrey, July 2017

[73] W. Duch, R. J. Oentaryo, M. Pasquier, Cognitive Architectures: Where do we go from here? Proceedings of the 2008 conference on Artificial General Intelligence, 2008, 122-136

[74] I. Kotseruba, O. J. A. Gonzalez, J. K. Tsotsos, A Review of 40 Years of Cognitive Architecture Research: Focus on Perception, Attention, Learning and Applications, Cornell University Library, 2016, https://arxiv.org/abs/1610.08602

[75] M. Beetz, L. Mosenlechner, M. Tenorth, CRAM — A Cognitive Robot Abstract Machine for Everyday Manipulation in Human Environments, 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 18-22, 2010, Taipei, Taiwan

[76] R. Sun, N. Wilson, M. Lynch, Emotion: A Unified Mechanistic Interpretation from a Cognitive Architecture, Cognitive Computation, 8(1) (2016), 1-14

[77] P. Robinson, R. El Kaliouby, Computation of emotions in man and machines, Philosophical Transactions Royal Society B, 3441 (December 2009), 3441-3447

[78] S. Wintermute, Imagery in cognitive architecture: Representation and control at multiple levels of abstraction, Cognitive Systems Research, 19-20 (2012), 1-29