Computational Architectures for Intelligence and Motivation

Darryl N. Davis

Abstract-- **This paper presents an overview of five years work into computational architectures for intelligence and motivation. The starting point for this research is a generic computational architecture arising from the study of the control state perspective to mind. Variations on this architecture have been used to investigate questions about the nature of autonomy, emergence, belief management and motivation (and other control states) in synthetic agents. A number of different domains have been used, for example simple a-life environments, simulated robotic factories, five a-side football, Tileworld and the game of Go. Due to the nature and scope of the work not all issues were addressed in any one domain. This raises questions about such endeavours. Given that it may be necessary to apportion fundamental research questions across different projects, can generalised conclusions be justifiably made from this division of research? The suggestion made here is that any such conclusions can only be justified in the light of subsequent integrative research. Current research directions are described in terms of these integrative questions.**

Index terms-- **architectures, control states**

I. Introduction

Intelligence can be (narrowly) defined as that quality that facilitates directed and purposeful behaviour. An agent that uses planning to achieve goals in some environment displays intelligence. Can it be said that an agent that achieves similar goals, as an emergent property of its behaviour without any similar deliberative processing, is less intelligent? In fact an intelligent agent should be able to recognise which of these types of behaviour is more applicable to any situation and respond accordingly. The research in this paper addresses the design of computational architectures that allow flexibility in both internal and external behaviour.

Underpinning the notion of achieving goals is the setting and management of goals. Goal management is one type of motivational control state. A small number of other motivational control states are identified in this paper. The thesis is that intelligence relies on motivation and similar forms of control state to provide processing contexts that enable intelligent behavior. Hence in pursuing designs for computational architectures that permit intelligent behaviour, it makes sense to consider what forms of control state are required for specific tasks and environments and what control states specific types of architecture allow. The second section of this paper presents a terse description of the theory of control state and a type of generalised architecture that will support such control states. Exemplars of this architecture are presented in the subsequent sections. The first architectures are used to address questions about the nature of emergence and the concept of autonomy. Subsequent architectures were designed and implemented in response to specific research questions arising from the study of motivation and its relation to other control states. The final architecture presented builds on this work and represents the current state of this research.

Department of Computer Science, The University of Hull, Kingston-upon-Hull, HU6 7RX, England. E-mail: D.N.Davis@dcs.hull.ac.uk

II. Generalised Architecture for Cognition

The control state approach to understanding cognition arose out of the work of Simon [1] and Sloman [2]. This theory of mind [3] considers that cognition is driven by and consists of control states. Relating this to more epistemological approaches to cognition (eg [4]) is a long standing goal of the current work. Consider an incomplete taxonomy of control states primarily centred on the concept of the motivation. Motivators are epistemologically instantiated processing contexts to assess and respond to situations in certain ways. This involves the perception and recognition of (problematic) events and states and the formation of representations and action paths to modified states of affairs. Four sub-types relevant to this work are:

- Goals defined in terms of task completion. Goals in control theory tend to involve negative feedback as a means of modifying behavior. Most artificial intelligence goals (especially in the planning literature) involve relations, predicates, states, behaviours etc. These can be mixed with soft computing techniques;
- Reflexes are ballistic mappings from input (i.e. perception) to output (i.e. behavioural response) and typically involve single actions (e.g. the knee jerk). Multiple reflex actions (e.g. walking on ice) are covered by control theory goals.

- Desires are related to beliefs, emotions and other control states and map onto the desired state defined in a goal. Rational desires involve objects, agents or events in an agent's internal world that may relate to plausible states in the external world. Desires may be irrational and relate to an implausible model of the world.

- Impulses are transient desires. However when acted on they can give rise to non-transient changes in internal and/or external worlds.

At any specific instant there may be one or more control states in various stages of emergence, dormancy, activation and accomplishment. A control state can be initiated through perceptual acts (vision, communication etc.) or through wholly internal processes (e.g. planning, belief management, other control states etc.) or through some combination (e.g. perception triggering a series of internal processes). Control states can be antipathetic or complementary. For example the reflex to catch a slippery knife while washing cutlery may be over-ridden by the desire not to cause one's self harm. Where this situation occurs frequently it may initiate the goal to purchase a dishwasher. Financial or other constraints may preclude this goal from being attained. The fixation or recurrent deliberation on unattainable states of affairs is the type of internal processes that an intelligent agent should recognise and manage.

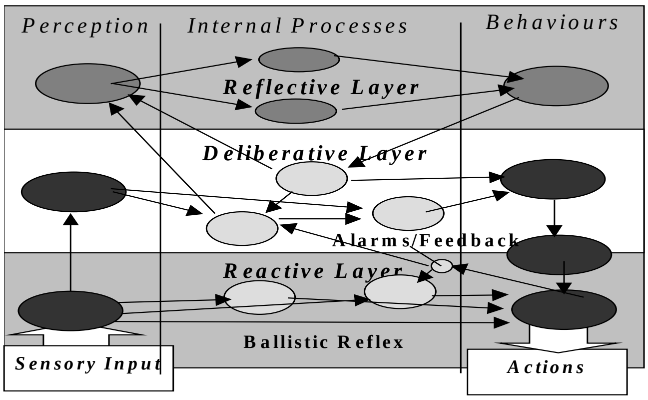

Figure 1. Generalised architecture (3 column/3 tier)

The question then arises of what sort of architecture can support such processes given that some are transient, some require immediate response to perceptual events with the possibility of control overrides and others require deliberation over more persistent representations. The generalised architecture being explored is that of a three-column, three-tier architecture (figure 1). Similar architectures are used elsewhere [5], [6]. Information typically flows from perception to action in the external world. Different processes respond to specific information in different ways. Feedback exists in a number of forms.

Reflexes are incorporated as low-level (ballistic) mappings from perception to action (in the reactive layer), but typically afford feedback in terms of alarms. Hence in the soapy knife scenario the catching reflex is over-ridden by a self-preservation alarm. In some computational experiments [3] this layer has been made separate for design and domain reasons, giving rise to a four-layer architecture. The reactive layer typically models transient control states and continually adjusts to the external world in the same way as behaviour-based systems [7]. It can be influenced in a number of ways by processes elsewhere in the architecture. The deliberative layer represents typically more slowly changing control states. As with the reactive layer not all processes may be active. Some may be activated through perceptual information, while others respond to control information arising from the second and third columns of the reactive layer. The third (reflective) layer is the meta-level of this architecture. . The reflective layer perceives control state information and acts on internal processes. It may act on deliberative processes, which in turn modify the reactive layer, or it can directly affect the reactive layer. The reflective layer acts as a form of control state feedback. Although it may, at times, act as one, the meta-level is not a central coordinator. There is no central coordinator in this generalised architecture. The apparent need for a central coordinator, arising out of the mind as an certain type of operating system with a central processor, is an illusion caused by a lack of understanding about the nature of attention in the human mind. At a theoretical level and in many computational experiments we have found no need for a central coordinator, relying on process valencing (whether quantitative or qualitative) to bring about process (i.e. control state) dominance or effective resource management.

In designing computational architectures for specific tasks in specified domains it can be helpful to categorise the types of control state that it needs to support, recognise or manage. This can be of help in identifying the minimal set of processes required for that application and help to ensure design parsimony. The following sections detail specific implementations based on this architecture.

III. AUTONOMY AND MOTIVATION

Experiments with simple a-life scenarios (predator-prey worlds) [8] were run to investigate the relation between specific control states and the two lower layers of the architecture. In particular, whether explicit motivation processes and structures are required and whether in simple (reactive) architectures (modelling the base layer of figure 1) we can rely on emergence. This led us to consider what autonomy means for the sorts of agents capable of supporting the control states described above.

Autonomy, one of four foundations for a weak notion of agency [9], has been defined as operating "…without the direct intervention of humans or others, and have some kind of control over their actions and internal state". Castelfranchi [10] categorises and discusses the various types of autonomy that a cognitive agent can demonstrate. In particular, a distinction is drawn between belief and goal autonomy. If an agent is to be autonomous, then it must set and have control over its goals. External agents, whether biological or computational, should not have direct control over the setting of an agent's goals, but can only influence these through the provision of information that affects an agent's belief set. The management of belief sets is described in terms of rationality and the credibility of incoming information. Castelfranchi's definition of autonomy is compromised where an agent relies on non-symbolic representations (for example [11]).

The Predator-Prey experiments arose from concerns about the nature of autonomy in simple and more complex agents. The experiments made used of simple agents with drives to explore, feed, reproduce and avoid threats. These drives defined a base level of autonomy. The agents had to rely on information from their senses to determine which of these drives is more important. The simplest architectures used a fuzzy valued strength for the importance of these values and ranked them in terms of this value. Certain inadequacies with these agents were noticed. As these agents had no persistent representation of their motivational state, they were in effect reacting to the current but transient state of their environment. In effect these tropistic agents did not mediate their control states through the use of long-standing or ongoing motivational constructs. An instance of this inadequacy is maundering, the switching of behaviours to facilitate seemingly incompatible goals.

Consider a simple experiment where a mobile (prey) agent perceives a predominantly static predator agent flanked by two food items. While the energy levels of the agent are high it will move away from the predatory agent. As energy levels drop, the signal strength to move towards a food item increases. At some point the signal strength associated with moving towards the food becomes sufficient that it is adopted as the primary behaviour (an implicit goal). The agent therefore moves towards one or both of the energy sources, and hence the predatory agent. The behaviour to move away from a predator is then given a higher signal strength and the agent moves in its original direction away from the predator and hence the food. Typically the behaviour of the agent fluctuates until it runs out of energy or is consumed by the predator. This agent is in effect caught in a cycle of goal conflicts. The use of planner with no further change to the agent will not help. The agent needs to be able to perform planning or some equivalent form of action sequencing that incorporates models of its internal states and external environment over time. These are associated with hysteretic agents that make use of explicit representations.

IV. EMERGENCE AND MOTIVATION

The next experiments considered how to recognise emergent patterns of behavior using the control state approach. This investigation involved the modelling of the lower levels of the generalised architecture as a society of simple agents [12]. Rather than produce a game-playing agent with a human model of how to play the game of Go, we modelled the rudiments of the game using co-operating communities of cellular automata. We hoped to demonstrate that strategic moves usually associated with deliberation could result as an emergent property of co-operating (reflexive and reactive) agents. This would then pave the way for more sophisticated agents which could recognise different patterns of emergent behavior.

A macro-agent is itself an environment within which other agents are at work This is analogous to the computer simulation of cities with agents (or actors) inhabiting those cities. Each city is itself a macro-agent within a society of cities. Such societies evolve by means of a number of processes at various levels. For example the interactions of the cities, the constraints the cities place on their inhabitants, the redefinition of the parameters of the cities by the actions of agents (or actors) within those cities and the roles the actors play in mitigating resource distribution within and between cities. Such modelling strategies may be appropriate in computationally expensive search spaces as a way of deferring computational trajectories through such spaces to higher and higher levels of abstraction. At that point where no further deferment can be obtained, the computational model may be displaying sufficiently intelligent behaviour that the search space problem becomes tractable or irrelevant. We hoped that such an approach would alleviate the computational cost of traversing the entire search space for the game of Go. While our initial experiments seemed promising there were real problems we did not solve.

The readings in [13] identify four types of emergence and specific problems associated with research into emergence. Particularly relevant to this research is adaptive emergence. The game playing agent needs to demonstrate that it can take advantage of its currently most beneficial processes. Our assumption, one of individualistic bias, was that collective (and co-operative) activity would emerge opportunistically. This was naïve. To make use of adaptive emergence the overall agent needed feedback mechanisms from the micro (individual) level to the macro (societal) level and from the societal level to the individual level (a kind of focus of attention in a macro-agent). Our initial implementations lacked this feature, and relied on information at the individual agent level to determine which move to make. For instance an individual agent using one of a number of alternative cell process models would suggest that a move at a specific location is beneficial. With a set of competing agents, each using alternative process models, a number of alternative moves would be suggested. The agent with the largest current "*drive-value*" would win and its move made. Our macro-agent lacked abstract models of these processes and would not follow any one strategy over time. It would not adapt but merely maunder according to which micro-agent was currently shouting the loudest. It needed to represent the underlying behaviors at a more abstract level. Strategies could then emerge from the adoption of localised tactical moves at this abstract level. The macro-agent is required to adapt and provide feedback to its component processes.

These simple experiments into emergence have impacted on the way we view agent architectures. The framework presented in figure 1 needs to manage perturbant processing patterns and focus the processes associated with an agent (across the first two layers) to specific categories of tasks. A macro agent's need to monitor emergent patterns of behaviour, to classify them as (potentially) useful or disruptive and manage them is a clearly analogous situation. One major challenge, which current work addresses, is how does such a macro agent recognise and harness these emergent behaviours, given that not all such patterns of behavior are known prior to the agent implementation. We are addressing this challenge by considering a computational equivalent to emotion. This is a theme that the following sections of this paper address.

V. MOTIVATION AND REPRESENTATION

Historically the first experiment [14], this architecture (quite similar to the two bottom layers of figure 1) was developed to investigate the type of structure, its constituent components and associated information handling processes that is suitable for capturing information relevant to motivation management.

Motivators do not exist in isolation. They relate the current state of an agent's environment to an alternative or future state and involve processes internal to an agent, including control states and other phenomena. Control states can be explicit, such as goals, or be implicit. Implicit control states are distributed among co-existing processes and memory structures. Our experimentation with implicit control states such as alarms (for example to replenish energy levels) were described in section 3. Here experimentation with explicit motivators in single agents is summarised.

A combination of techniques provides a means for representing behaviours and process oriented control states, for example beliefs, emotions, motivators and goals. The use of propositional calculus and finite state graphs allows the specification of the processes that manage incoming information; select and activate external behaviours; generate and manage motivators; plan; make decisions; and manage memories. An explicit motivator structure (see [3] for a full description) draws together many aspects of the control state taxonomy. The same computational structure are used for motivators at the reactive level as at the deliberative layer even though many components, for example plans, motivator rationale and resource management, are only relevant to the deliberative-layer.

The core of a motivator is a descriptor defining the desire that the agent wants to maintain, preclude or accomplish. Consider two propositions **p1** *avoid(predatorA)* and **p2** *move(localeX).* A descriptor can be a single attitude towards a single proposition, e.g. ***keep-true***(**p1***)* or ***make-true***(**p2**). In many cases there may be a need for multiple attitudes towards possibly multiple propositions, for example ***keep-true(*****p1***) AND* ***make-true(*****p2***)*. While relatively inflexible constraints can be associated with motivators, the pursuit of conjunct attitudes can lead to the relaxation of such constraints. For example, the emotional correspondence of a motivator may be activated through the attainment of one at the expense of other propositions. If *Fear* is associated with both the above attitudes, the agent can decide to risk a high level of *Fear* arising from the compromising of ***keep-true(*****p1**) to attain ***make-true(*****p2***)*.

Further components identify factors such as sub-goals, plan set names, and the environmental objects and other agents referenced by a motivator. Several components serve as motivator management information. Importance, insistence, intensity and urgency are modelled using a set of qualitative values, e.g. *high*, *medium*, *low*. These values are calculated as needed and modified during the lifetime of a motivator. For example, an avoid-threat motivator has its urgency and importance calculated when generated at the reactive layer. Such a motivator for an agent close to a possible threat would have a high importance. An agent far from the same possible threat may generate a motivator with a low importance. If the possible threat is close, the urgency value will be high. The insistence value takes the higher of these two components. The intensity of a motivator is calculated by a means of a comparison with other motivators and the agent's current belief. A decay function ensures that insistence decreases while a motivator remains passive or postponed (i.e. generated but with no further processing).

Although the motivator handling architecture showed promise, there were a number of serious flaws. The design and implementation experiments highlighted one of the major problems in complete agent research as identified by Franklin [15]; that of integrating learning. At the design level, it is possible to see how different categories of learning segue well with certain capabilities or levels of processing. For example, reinforcement learning is appropriate for the decision mechanisms in the reactive layer. However, such techniques will not be appropriate across all the architecture. There is no obvious way to integrate learning across this multi-level computational model. No one single process (however connected) could account for these learning mechanisms at a theoretical, design or computational level.

These experiments also identified a theoretical impoverishment with the concepts associated with the reflexive (meta-agent) layer. What form of currency could the reflective layer use that measures phenomena such as maundering at the reactive layer and motivator conflict at the deliberative layer? An analysis of the philosophy and psychology of mind lead to an emotion-based core for the theory and across all layers of the agent architecture [16]. This agent-oriented internal currency allows a reflective monitoring of its internal interactions and relates these to its tasks in the external environment. Internal conflicts and more harmonious states are measured in terms of qualitative emotion types (fear, anger etc) and emotion intensity (high, low etc.). The impetus for change within itself (i.e. a need to learn) is manifested as a combination of emotive states. Such states lead to the generation of motivators requiring the agent to modify its behaviour or processes in some way. The modification of an agent's internal environment is then defined in terms of an emotion (control-state) mapping between its internal and external environments and represented using a motivational structure.

VI. SHARED MOTIVATIONS

We have since considered how to use dynamic multiple agent architectures with shared motivations. In such heterogeneous communities of agencies, the initial capabilities and behaviours of an agent can be specified, but also the conditions for how its constituent processes may adapt or be tuned to the requirements of specific current tasks and their related control states. This idea of co-operative and competitive cliques of agents pulls together the threads related to emergent processing and control state architectures. Figure 2 sketches an example architecture that has been used for experimentation [17]. The deliberative agent subsumes the deliberative layer shown in figure 1. It acts as dynamic representational framework within a distributed blackboard architecture to support shared motivations across agents competing in five-aside football.

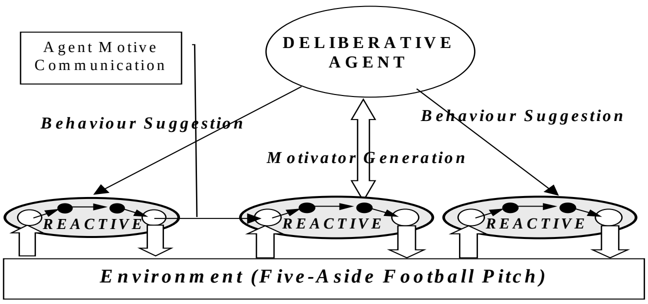

Figure 2. A societal approach to motivation.

This research used a simplified form of the representation described earlier. These agents made use of simple motivators in the form of goals and drives. A goal is a desire to move from one state to another or maintain a current state. Goals require planning and explicit differentiation between the external environment and the agent's internal environment. At any one time an agent may have multiple goals through performing certain sequences of tasks, or a certain task may spawn into multiple goals due to its behavior sequences. A drive is a transient but periodic need. *Urgency* is defined in terms of the deadline for an intended goal. *Importance* may be intrinsic (e.g. Fear is an important drive) or may be based on an assessment of the consequences of certain actions. *Insistence* combines *importance* and *urgency* to determine which action or sets of actions are to be performed.

Five types of agent communities were investigated. Behaviour-based agents with the basic capabilities to play 5-aside football (A0). Agents with the basic capabilities plus motivators (A1). Agents with the basic capabilities plus motivators and drives (A2). Agents with the basic capabilities plus motivators, drives and urgency (A3). Agents with the basic capabilities plus motivators, drives and urgency plus advice taken from their coach (A4 – the full implementation of figure 2). Communication between teammates could be switched on or off for all types. Tables 1 and 2 summarise the results for ball possession and games won when these five agent types are pitted against each other. Type A4 agents are consistently better than all others, closely followed by type 3. The difference between A0, A1 and A2 types is slight. Communication alone does not assist the agents to increase ball possession or to win the game. However, with the absence of coaching in the opponent's team, communication enables the team to draw or lose with a more marginal score difference.

Table 1. Experimental Results for Ball Possession

Possession	*A0 Agents*	*A1 Agents*	*A2 Agents*	*A3 Agents*
A0 Agents	----	----	----	----
A1 Agents	A1	----	----	----
A2 Agents	A2	A1	----	----
A3 Agents	A3	A3	A3	----
A4 Agents	A4	A4	A4	A4

Table 2. Experimental Results for Games Won

Games Won	*A0 Agents*	*A1 Agents*	*A2 Agents*	*A3 Agents*
A0 Agents	----	----	----	----
A1 Agents	A1 does not lose	----	----	----
A2 Agents	A2 does not lose	A2 wins	----	----
A3 Agents	A3 wins with high score	A3 wins with high score	A3 wins with high score	----
A4 Agents	A4 wins with very high score	A4 wins with high score	A4 wins with high score	A4 wins with marginal score

Motivators enable the agents to focus on performing the most appropriate action with more frequent ball possession but necessarily guaranteeing a win. Drives enable agents to choose between tasks. While not guaranteeing greater possession, drives enable agent teams to beat teams without drives. On the other hand, agents with drives obtained the ball less times than those without the drives. Urgency enables an agent to determine how fast it needs to perform a task, and enables such agents to defeat those without urgency by a very high score. Agents without urgency did not score one goal against an agent making use of urgency. The conclusion to be derived from the experimental results is that the coach acts as an effective co-ordinator of agent actions. Despite the possibility of agents ignoring the advice given by the coach, the frequency of agents taking this advice is high. The type A4 agent, with motivators, drives, urgencies and coach, consistently beat all other agent types. Further experiments with these architectures are required to determine the balance of communication between the reactive agents and the control communication between the deliberative adviser and the ball players.

VII. CURRENT RESEARCH DIRECTIONS

Current research on the generalised architecture is looking to pull the above threads together in a number of domains. This includes the Go playing agent built on communities of low-level (reflexive and reactive) agents plus deliberative agents to manage motivational constructs and a reflective agent. The results from experiments with earlier architectures [16] suggest the use of a computational analogy to emotion to drive the reflective layer and provide a means for the agent to valence control information across the different layers and components. Four communities of CAs make up one (low-level) reflexive agent and provide the basis for the decision-making space for this agent. Alternative (internal and communication) behaviours for the cellular automata are modelled using a society of potentially competitive agents. These agents model alternative game tactics. When run, each agent produces a list of alternative moves. Agreement across lists produced by the competitive agents suggests good moves. Disagreement requires arbitration. In either case longer term strategies may suggest a different move. The earlier macro-agent approach therefore needs to be modified so it is capable of supporting these preferred more slowly changing control states.

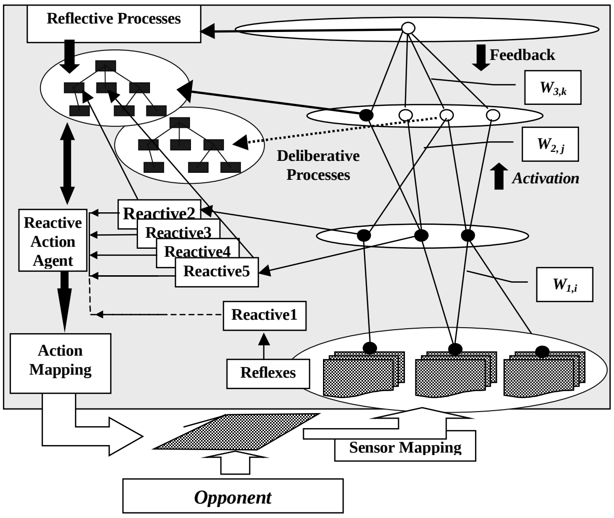

Figure 3. Sketch of the integrated Go architecture. All Boxes, Circles and Ellipses represent agents. Light Grey Box is the Macro-Society of Agents

The reflexive and reactive agents (reactive 1 to reactive N in figure 3) are in effect unaware of each other's presence. Communication at this level consists of control messages moving in hierarchical directions. One form of control message (right hand side of figure 3) activates deliberative agents in response to specific patterns or cues in either the position of stones on playing surface (e.g. singleton stone under threat of capture) or the behaviour of the agents. Multiple actions posted at the reactive level cause the reactive action agent to post an alarm to activate a deliberative agent to arbitrate. Lateral communication exists between agents in the deliberative layer. Motivations in this agent are related to game-playing strategies. For instance, there may be two opportunities on the board to capture opponent's stones. Each of these will be recognised by pattern-matching reactive agents. This will result in the same deliberative agent being activated twice to construct a motivational representation for each of these situations. The adoption of such a move provides a motivational context for the macro-agent. The viability, urgency and importance of these contexts are re-assessed over time. We hope to demonstrate that such an architecture can merge emergent and deliberative processes within a distributed model of cognition. The results of earlier experiments on theoretical architectures have fed into more application-oriented work on agents [17],[19]. We expect the same from the current research.

VIII. REFERENCES

[1] H.A. Simon, Motivational and emotional controls of cognition, *Models of Thought*, Yale University Press, 1979.

[2] A. Sloman. The mind as a control system. In *C.Hookway and D.Peterson, editors, Philosophy and the Cognitive Sciences*, pages 69--110. Cambridge University Press, 1993

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

[4] Fodor, J.A., *Concepts: where cognitive science went wrong.*, Oxford University Press, 1998.

[5] A. Sloman, What sort of architecture is required for a human-like agent? *Cognitive Modeling Workshop, AAAI96*, Oregon, 1996

[6] M. Minsky, An architecture for a society of mind, AISB 2000, University of Birmingham, 2000.

[7] R.A. Brooks, Intelligence without representation, *Artificial Intelligence*, 47:139-159, 1991.

[8] D.N. Davis, Cellular Automata, Computational Emotion and Autonomy, *International Conference on Computational Intelligence for Modelling, Control and Automata, Las Vegas, USA*. July 2001.

[9] M. Wooldridge and N.R. Jennings (Eds), *Intelligent Agents*. Springer-Verlag, 1995

[10] C. Castelfranchi, Guarantees for autonomy in cognitive agent architectures. In [9]: 56-70

[11] L. Steels, When are robots intelligent autonomous agents? *Robotics and Autonomous Systems, 15,* 1995.

[12] D.N. Davis, T. Chalbi and B. Berbank-Green, Towards An Architecture for A-life Agents:II, In: *New Frontiers in Computational Intelligence and Its Applications, ISO Press*, 1999.

[13] N. Gilbert and R. Conte, *Artificial Societies: The computer simulation of social life*, UCL Press, ISBN 1-85728-305-8, 1995.

[14] D.N. Davis, Reactive and motivational agents. In: *Intelligent Agents III: Agent Theory, Language and Architectures, Springer-Verlag*, 1997.

[15] S.P. Franklin, Autonomous Agents as Embodied AI, Cybernetics and systems, Vol. 28, No. 6, 1997, pp. 499-520.

[16] D.N. Davis, Agents, Emergence, Emotion and Representation, IEEE International Conference on Industrial Electronics, Control and Instrumentation (IECON2000), Nagoya, Japan 2000.

[17] H.A. Nunes, Investigation of motivation in agents using the simulation of 5-aside football, *M.Sc. Research Thesis, Computer Science, University of Hull, 2001.*

[18] D.N. Davis, D.N., Agent-Based Decision Support Framework For Water Supply Infrastructure Rehabiliation and Development, *International Journal of Computers, Environment and Urban Systems, vol 24, 1-18, 2000.*

[19] D.N. Davis, Y. Luo & K. Liu, Combining KADS with Zeus to Develop a Multi-Agent E-Commerce Application, *International Journal of Electronic Commerce Research, Kluwer*, 2002. (In Press)