Triad Views of Cognition in Intelligent Agents and Intelligent Cores for Fault Tolerance

Blesson Varghese, Gerard McKee, Vassil Alexandrov
School of Systems Engineering, University of Reading, Whiteknights Campus
Reading, Berkshire. RG6 6AY. United Kingdom.
b.varghese@student.reading.ac.uk, g.t.mckee@reading.ac.uk, v.n.alexandrov@reading.ac.uk

***ABSTRACT*:** The work reported in this paper aims to formalise two approaches based on intelligent agents and intelligent cores which are employed for automated fault tolerance in high-performance computing systems. A layered cognitive architecture is therefore mapped onto intelligent agents and intelligent cores, specifically taking into account the perceiving, reasoning, judging, responding and learning capabilities of agents and cores for demonstrating their cognitive intelligence. The mapping results in three views of cognition, namely an agent's view, a swarm's view and a landscape's view of cognition, referred to as 'triad views'. A qualitative evaluation of the mapping on the basis of a set of reference criteria is presented.

Keywords: Cognitive architecture, Intelligent agent, Intelligent core, Cognition view

Received: 4 August 2009, Revised 19 September 2009, Accepted 1 November 2009

1. Introduction

The number of computing cores employed in high-performance computing systems has risen significantly in recent times, and has resulted in an increase of recurring faults in such systems. Conventional fault tolerant mechanisms employed to isolate faults tend to be manual procedures and rely on human administrator interventions (Vallee et al., 2008). These mechanisms require large times for restarting a stalled process which was executed on the high-performance computing system prior to the occurrence of a fault (Yang et al., 2009). Hence, the need for automated fault tolerant mechanisms has risen.

Recent research in automated fault tolerant mechanisms for high-performance computing systems has led to the development of two approaches, namely the intelligent agent (Blesson et al., 2010a) (Blesson et al., 2010b) and intelligent core approaches (Blesson et al., 2010c) (Blesson & McKee, 2009) for fault tolerance. A task to be computed on a high-performance computing system is decomposed into sub-tasks and mapped onto software agents that carry these tasks onto the computing cores of a high performance computing system. In the first approach, the software agents are able to make decisions about the expected failure of a computing core and take action to relocate to a safe core. In the second approach, this intelligence is located within the computing cores, which make decisions about their possible failure and effectively push the software agents onto neighbouring cores.

Experimental results gathered from the implementation of both the above approaches on computer clusters have proved that conventional fault tolerant methods which require human administrator interventions can be replaced by these approaches that tend to be self-managing and require lesser times for reinstating process execution.

The current state-of-work of the intelligent agent and the intelligent core approaches lack formalisation. To formalise the perception, reasoning, judging, response, judging and learning capabilities of these approaches there is a need for an agent architecture to be mapped onto these approaches. The work reported in this paper, therefore aims to map a layered cognitive architecture onto the intelligent agent and the intelligent core approaches. The perceiving, reasoning, judging, responding and learning capabilities of software agents and computing cores for demonstrating their cognitive intelligence to achieve

fault tolerance is presented. The mapping results in three views of cognition, namely an agent's view, a swarm's view and a landscape's view of cognition, referred to as 'triad views'. A qualitative evaluation of the mapping on the basis of a set of reference criteria is presented. It is to be noted that no practical experimental results are presented in this paper.

The remainder of this paper is organised as follows. The second section presents a layered cognitive architecture that is employed for the mapping onto the intelligent agent and intelligent core approaches. The third section presents the triad views of cognition. The fourth section evaluates the mapping taking into account a set of reference criteria. The fifth section concludes this paper by considering future work.

2. Layered Cognitive Architectures

To formalise the intelligent agents and intelligent core approaches considered in the above sections, a layered cognitive architectures needs to be mapped onto these approaches. After an extensive search in literature relevant to cognitive architectures, a recently proposed Computational Intelligence based Architecture for Cognitive Agents (Lawniczak & Di Stefano, 2010), which will be referred to as CIACA in this paper was found. The architecture formalises the cognitive capabilities, namely perceiving, reasoning, judging, responding and learning capabilities of an agent. Therefore, a perceptual layer, a reasoning layer, a judging layer, a response layer and a learning layer are five layers constituting the architecture.

The perceptual layer perceives information from the environment by both sensing and from interacting agents (Lawniczak & Di Stefano, 2010). For example, in a traffic highway model, if a car on the highway is assumed to be an agent and the highway its environment, then information such as other cars ahead and behind and their approximate distances from the agent is acquired by perception.

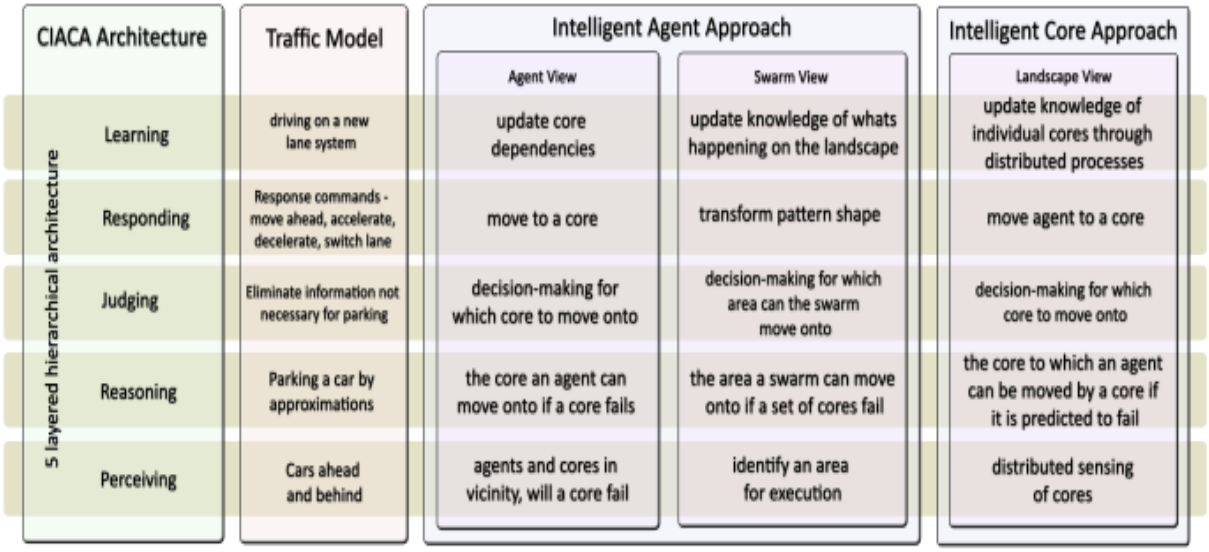

Figure 1. Illustration of the CIACA architecture for traffic models, and the mapping of the CIACA architecture onto intelligent agents and intelligent cores

The reasoning layer supports coherent and logical thinking by obtaining information from the perceptual layer and processing it using existing knowledge (Lawniczak & Di Stefano, 2010). The natural thinking process in humans comprises fuzzy logic and is therefore suggested as a useful tool for reasoning. When an agent is only able to perceive an approximate view of the environment, a result may still be inferred by applying approximate rules. For example, humans successfully park a car though many approximations are made.

The judging layer receives information from the reasoning layer and may send it back to the reasoning layer after processing or refining it to make decisions at the reasoning layer (Lawniczak & Di Stefano, 2010). For example, if a vision system is used for parking a car between two other parked vehicles, information irrelevant for successful parking which may also be obtained through the vision system needs to be eliminated.

The response layer instructs response commands to the perceptual layer after applying rules to the information obtained from the judging layer (Lawniczak & Di Stefano, 2010). For example, move ahead, accelerate or decelerate and switch lane are response commands. Additional reasoning and judging capabilities can also be implemented in this layer for exhibiting higher level of intelligence.

The learning layer modifies existing knowledge by using information gathered by the agent. The ability to learn is implemented within algorithms such that an agent is able to derive new knowledge. Extended learning by different generation of agents leading towards evolution of knowledge can also be implemented in this layer.

The above architecture formalises the cognitive capabilities of an agent in a hierarchical layered sequence and has provisions to accommodate additional computational strategies which can be implemented in these layers. Therefore, in the next section, the layers of the CIACA architecture are mapped onto the intelligent agent and intelligent core approaches for fault tolerance.

3. Triad Views: Mapping a Cognitive Architecture for Intelligent Agents and Intelligent Cores

An efficient mapping of a cognitive architecture onto the intelligent agent and intelligent core approaches can be obtained by considering three different views (or the triad-views) of the components of the approaches, namely an agent, the swarm of agents, and a landscape. The perception, reasoning, judging, response and learning capabilities of an agent, a swarm and a landscape will be explored in this section. Figure 1 illustrates the three views that are obtained by the mapping.

3.1 View 1: Agent View

In the first instance an agent's view is considered. In the intelligent agent approach, the environment in which an agent is situated comprises agents with which it can interact and computing resources. Perception in this context would mean to acquire information concerning the environment. An agent needs to answer questions such as 'are there other agents in my vicinity?' and 'which computing cores are functional in my vicinity?' To achieve this, an agent can probe the environment, i.e., by sending 'are you alive' signals to the agent and the computing resources. Perception for an agent also includes gathering information for answering the question will the core that I am situated on fail? Sensory information for predicting such failures can be gathered through sensors that consistently probe the hardware core. For example, the rise in temperature of a core can be sensed to predict a computing core failure.

Reasoning becomes necessary once an agent predicts the computing core it is situated on to fail. An agent needs to answer questions such as which cores in the computing environment would it be possible to move onto? Since an agent has options to move onto a few cores in its vicinity, an agent needs to make an appropriate choice. Hence, the agent needs to also think will the core that I will move onto fail. For this an agent should have perceived sensory information of the cores in its vicinity.

Judging for an agent is necessary for decision making. For example, an agent may think about which core do I move to, but a decision has to be made concerning the core to which an agent can move. As suggested above, the sensory information perceived by an agent aids decision making. However, in the context of the intelligent agent based approach in swarm-array computing, the judging layer necessarily need not be implemented as a separate layer; it may complement the reasoning layer.

After an agent makes a decision as to which core it can move onto, a response needs to be initiated. If the response is instructed explicitly through an external controller then the agent's cognitive capability is challenged. On the other hand, a response initiated by the agent itself is appropriate in the context of achieving intelligence by demonstrating cognitive capabilities. An instruction such as 'move to' or 'move to core x' initiated so that an agents move onto a core other than which it is situated on is an example.

A sophisticated agent also requires mechanisms whereby it learns about its environment from the perceived sensory information and uses it for decision making. For example, in the context of the intelligent agent algorithm, an agent updates its information on the cores it is dependent on. The core dependencies known to the agent and the knowledge gained from 'are you alive' signals contribute to the knowledge of an agent about the landscape.

From an agent view, the CIACA architecture maps well onto the intelligent agent based approach. This is so since the mapping on the agent level provides a microscopic view of the intelligent agent approach. The perception layer provides functionalities for acquiring information from the agent's environment. The reasoning layer enables an agent to think logically while thejudging layer assists an agent in narrowing down an agent's choice. The response layer enables an agent to initiate a response

while the learning layer updates existing knowledge of an agent.

With the current state of work in the intelligent agent approach it is to be noted that reasoning and judging need not be implemented as separate layers since decision making does not involve many choices. The CIACA architecture does not address issues such as security, resource management and providing generic services. However, since the intelligent agent approach implemented in parallel reduction algorithms considered low-level aspects, such issues did not have to be considered.

3.2 View 2: Swarm View

Having considered the cognitive capabilities of an agent there is also a need to consider how a group of agents or a swarm of agents can demonstrate intelligence in the swarm-array computing approach. Hence, the second view considered in this section is the swarm's view. The cognitive capabilities of the swarm are emergent since individual agents contribute to the swarm's behaviour. In other words, an agent demonstrates intelligence on a microscopic or individual behaviour level, whereas the swarm demonstrates its intelligence on a macroscopic or abstract behaviour level. For example, an agent might need to move onto many cores while executing a task and update its dependencies resulting in the displacement of the swarm on the landscape.

The emergent behaviour of the swarm due to the perception of individual agents is the capability to identify an area comprising a set of cores on the computing space where the swarm can situate itself to execute a task. In other words, the swarm perceives which area it can situate to execute a task and whether the set of cores can provide sufficient resources and access to resources for successful execution of the task mapped onto the swarm agents.

Reasoning on the swarm level includes questions such as 'where can I move onto if a set of cores in the computing space fail?' and 'will the new set of cores that I move onto fail?' The swarm's decision making or judging capabilities are demonstrated by the definitive and unequivocal decision it makes when a question such as 'which area on the computing space do I move to' arises after reasoning when a set of cores has been predicted to fail.

Response at the swarm level includes the capability of a swarm to displace itself from one location to another in the computing space if a set of core is predicted to fail. The displacement can occur by transforming the shape of the pattern formed by the agents whereby individual agents reposition to other cores such that the task mapped onto an agent is seamlessly executed. The knowledge of what is happening on the computing landscape aids decision making when the swarm has to move about on the landscape. This knowledge is learnt by the swarm from sensory information that is perceived and local interactions such as sending and receiving 'are you alive' signals by the agents comprising the swarm. On an implementation level, for the sake of convenience, knowledge can be acquired, maintained and updated centrally. However, a decentralized strategy for acquiring, maintaining and updating the knowledge-base is closer to the swarm concept.

In general, though the CIACA architecture mapped onto the swarm view, all layers of the architecture did not prove useful in the mapping. Perception of the swarm is an emergent behaviour and therefore is of less importance in this context. However, reasoning, judging and response can be seen on both microscopic (agent) and macroscopic (swarm) levels. Learning involves representation and storage of knowledge, which needs to be decentralized to be in similar lines of a swarm, and therefore operates on the microscopic level.

The mapping of the CIACA architecture on the swarm view is unlike the mapping of the CIACA architecture on the agent view for two reasons. Firstly, the level of abstraction for the agent and swarm view is different, since in the swarm view, macroscopic properties providing an abstract view is considered. However, in the agent view microscopic properties were considered. The CIACA architecture maps well on the microscopic level. Secondly, the interactions in the swarm level are more complex since they comprise both inter-agent and agent-environment interactions of all agents comprising the swarm. Hence, on an abstract level, the CIACA architecture proves less effective in the swarm view than on the agent view.

3.3 View 3: Landscape View

Another component whose view is considered is the landscape. The landscape plays a vital role in the intelligent core approach in which a collection of cores demonstrate intelligent behaviour.

The intelligent cores are abstracted such that they resemble a grid of computing cores. A hardware probing process on each

core monitors a parameter that could impair the functioning of the core (for example, increase of temperature on a core beyond a threshold could lead to a potential core failure). This resembles an array of distributed sensors performing a distributed sensing task, and in the case of intelligent cores, distributed sensing to predict failures. Each core also gathers and shares its information with adjacent cores, hence sharing sensory information and thereby perceiving its vicinity.

Reasoning in the landscape is required when a core predicts its failure. A process executing on the core predicted to fail will need to be reinstated onto another core in its vicinity. An appropriate choice as to which core in the vicinity should the process be moved onto will need to be made based on existing knowledge comprising sensory information gathered from the surrounding cores.

In the context of landscapes, judging is required to make a decision on which core is capable of reinstating process execution. Rules or dynamic policies that aid decision making may be employed based on existing knowledge. For example, if a core predicted to fail needs to decide between two cores that are less likely to fail, then the history of failures of the cores may be employed as a rule for decision making. If one among the two potential cores has a greater history of failures than the other, then it could be excluded from the list of potential cores. However, the judging layer need not be implemented as a separate layer but may complement the reasoning layer.

In the landscape a response is initiated by a core to move an agent situated on it onto another core determined by reasoning and judging. An instruction such as `move agent to' or `move agent to core x' which is initiated so that an agents is moved onto a core other than which it is situated on is an example. In the intelligent core approach the transfer of agents from one core to another is facilitated by processor virtualization.

The knowledge of what is happening on the computing landscape aids decision making when the agents are required to be moved about on cores in the landscape. Local knowledge is learnt by individual cores from the sensory information that is perceived and local interactions with other cores. When the landscape is viewed as a whole, knowledge is acquired, maintained and updated within the landscape on individual cores through a set of distributed processes.

In general, the landscape view maps the CIACA architecture and in this case maps well on the macroscopic level, unlike in the swarm view. This is so because the landscape is a more tightly coupled entity when compared to a collection of agents. In abstracted layer of the computing cores the dependencies of a core remain unaltered unlike in a swarm which has the need for more complex communication and coordination.

4. Quantitative Evaluation of the Mapping

This section evaluates the mapping of the CIACA architecture onto the swarm-array computing approach considered in previous sections. The set of six evaluation criteria presented by Langley, Lair & Rogers (2009) are used to perform the evaluation. The evaluation criteria are: (a) Generality, versatility, and taskability, (b) Rationality and optimality, (c) Efficiency and scalability, (d) Reactivity and persistence, (e) Improvability, (f) Autonomy and extended operation. This set of criteria is a general set of principles relevant to cognitive agent architectures and are broad in its scope of evaluation and hence adopted for the qualitative evaluation of the architecture.

4.1 Generality, Versatility and Taskability

Generality evaluates how well the architecture can support intelligent behaviour in a broad range of environments. The CIACA architecture is proposed as a general cognitive agent architecture and illustrated for traffic models. The applicability of the CIACA architecture for the intelligent agent based approach of swarm-array computing is illustrated in this paper. Though the CIACA architecture has been recently proposed yet has illustrated two applications on which the architecture can be mapped onto. To exemplify and evaluate the generality of the CIACA architecture more applications need to be investigated so that the CIACA architecture can be mapped onto.

Versatility evaluates how taxing is the process of constructing intelligent systems across a given set of tasks and environments. The architecture maps well on the microscopic level, i.e., agent view in swarm-array computing, but does not map well on the macroscopic level, i.e., swarm view in swarm-array computing. Therefore, the CIACA architecture is not necessarily versatile on an abstract view.

Taskability evaluates how an agent can carry out tasks not only by knowledge it has acquired but also by explicit communication with humans or other agents. In the intelligent agent based approach an agent's response to move off from a failing core is not only based on the knowledge it has of its environment but also from commands it may obtain as signals from other agents in its vicinity. The aim of the intelligent agent based approach is to create self-managing systems to execute a task by minimising human administrator intervention; therefore the approach does not consider receiving explicit commands from humans. However, in traffic models, humans need to make explicit commands to an agent representing a car to reach a goal.

4.2 Rationality and Optimality

Rationality evaluates how an agent's knowledge and action will lead towards its goal or in other words the relationship between an agent's goal, its knowledge and actions. In the agent view of the intelligent agent based approach, the primary goal of an agent is to execute a sub-task that is mapped onto it. To achieve this goal an agent may have to relocate on different computing cores. In this context, the degree of rationality of an agent will be based on how well an agent utilises its knowledge of the computing environment to execute a task.

Optimality evaluates whether an agent's selected behaviour yields an optimal solution. The degree of optimality will be high if an agent can successfully complete its task by being rational. In the case of parallel summation algorithms, every agent on whom the task of summation is mapped receives information from and yields information to other agents in the environment. The states that an agent can enter into, thereby showing different behaviours, is limited in this case as against agent behaviours that demonstrate different behaviours as shown in traffic models.

4. 3 Efficiency and Scalability

Efficiency evaluates the amount of time and space required by the computing system. The experimental studies on the cluster proved that the time for reinstating the execution of an algorithm once a failure occurred was significantly reduced when compared to the time taken by other existing traditional approaches. Clearly the efficiency of the approach increases with the adoption of cognitive agent architectures.

Scalability evaluates the architecture's performance in varying conditions including task difficulty, environment uncertainty and time of operation. The intelligent agent approach was implemented on a computer cluster focusing on space applications and simulated uncertainty in the environment which was sensed as a hazard by the agent. Scalability studies on other experimental platforms have not yet been explored for the approach. Moreover, the swarm-array computing approach has been implemented for parallel reduction algorithms, an important class of algorithms in parallel computing, but has not yet moved towards implementations for more complex algorithms.

4.4 Reactivity and Persistence

Reactivity evaluates how well an agent can respond to unexpected situations or events. The unexpected situation considered in the intelligent agent approach is a failure of the computing core. The failure is anticipated by an agent and the agent responds to this situation by making a decision to which core it must relocate. It is noted from experimental results that mean times taken for reinstating the execution of an algorithm if a core fails is in the order of milliseconds, and therefore confirms that agents respond and react quickly in the swarm-array computing framework.

Persistence evaluates how the architecture pursues its goals despite changes in the environment. An agent is not only affected by the failure of a computing core in its environment, but also by another agent in its vicinity. If an agent situated on a core predicted to fail is dependent on one or more agents in the environment, then dependency information needs to be circulated such that agents can continue to pursue goals despite changes in their environment.

4.5 Improvability & Autonomy and Extended Operation

Improvability evaluates the ability of an agent to perform a task with addition of knowledge when compared to the state it did not possess knowledge. Clearly, in the intelligent agent approach considered in this paper, an agent makes its decision as to which core it should move onto in the case of a predicted failure is based on its knowledge of which cores in its vicinity are not likely to fail. If the agent did not possess knowledge of its environment, an agent would make a decision that would not be necessarily optimal, i.e., moving off to a core that is likely to failure thereby requiring a further relocation at the expense of time and slowing the execution of the task.

Autonomy evaluates the personal independence of an agent. The degree of autonomy in the architecture can be evaluated

based on the cognitive ability of the agents seen in the approach rather than merely being reflexive agents.

Extended operation evaluates whether an agent can operate on its own for prolonged periods of time. To start off the intelligent approach was proposed to isolate faults when single nodes failed and continue seamless execution of a task for a prolonged period. Additional work will be required to enable the approach to handle multiple node failures, thereby extending the operation of agents for prolonged time frames in more realistic scenarios.

5. Conclusion

The work reported in this paper has considered the conceptual aspects to formalise the intelligent agent and intelligent core approaches employed for achieving automated and self-managing fault tolerance in high-performance computing systems. To formalise these approaches the CIACA architecture, a hierarchical and layered cognitive architecture is mapped onto the agents and the cores employed in the approaches. Cognitive capabilities, namely perception, reasoning, response, judging and learning, of an agent, a swarm and a landscape, referred to as the 'triad-views' are considered. The mapping of the CIACA architecture onto the intelligent agent and intelligent core approaches is qualitatively evaluated against a set of reference criteria.

Future work will consider how a cognitive agent architecture can be mapped onto a hybrid approach that incorporates both intelligent agents and intelligent cores. Further, a set of reference metrics to quantitatively evaluate the mapping will need to be developed. Immediate efforts will be made to identify how the mapping can be performed on multiple landscapes.

References

[1] Vallee, G., Engelmann, C., Tikotekar, A., Naughton, T., Charoenpornwattana, K., Leangsuksun, C., Scott, S. L (2008). A Framework for Proactive Fault Tolerance. *In: Proceedings of the 3rd International Conference on Availability, Reliability and Security*, p. 659-664.

[2] Yang, X., Du, Y., Wang, P., Fu, H., Jia, J (2009). FTPA: Supporting Fault-Tolerant Parallel Computing through Parallel Recomputing. *IEEE Transactions on Parallel and Distributed Systems* 20 (10) 1471-1486.

[3]Varghese, B., McKee, G. T., Alexandrov, V. N (2010a). Can Agent Intelligence be used to Achieve Fault Tolerant Parallel Computing Systems? accepted for publication in *Parallel Processing Letters Journal*.

[4]Varghese, B., McKee, G. T., Alexandrov, V. N (2010b). Intelligent Agents for Fault Tolerance: From Multi-Agent Simulation to Cluster-based Implementation. *In: Proceedings of the 24th International Conference on Advanced Information, Networking and Applications Workshops*, p. 985-990.

[5] Varghese, B., McKee, G. T., Alexandrov, V. N (2010c). Implementing Intelligent Cores using Processor Virtualization for Fault Tolerance. *In: Proceedings of the 10th International Conference on Computational Science*.

[6] Varghese, B., McKee, G. T (2009). Landscape of Intelligent Cores: An Autonomic Multi-Agent Approach for Space Applications. *In: Proceedings of the 9th International Conference on Applied Computer Science*, p. 117-123.

[7] Buczak, A. L., Greene, K., Cooper, D. G., Czajowski, M., Hofmann, M. O (2005). A Cognitive Agent Architecture Optimized for Adaptivity. *In: Proceedings of the Conference on Artificial Neural Networks in Engineering*.

[8] Greene, K., Cooper, D. G., Buczak, A. L., Czajkowski, M., Vagle, J. L. and Hofmann, M. O (2005). Cognitive Agents for Sense and Respond Logistics. *In: Proceedings of the International Workshop on Defence Applications of Multi-agent Systems*, p. 104–120.

[9] Franklin, S., Patterson Jr., F. G (2006). The Lida Architecture: Adding New Modes of Learning to an Intelligent, Autonomous, Software Agent. *In: Proceedings of the 9th World Conference on Integrated Design and Process Technology*.

[10] Ramamurthy, U., Baars, B., D'Mello, S. K., Franklin, S (2006). LIDA: A Working Model of Cognition. *In: Proceedings of the 7th International Conference on Cognitive Modelling*, p. 244–249.

[11] Langley, P., Cummings, K., Shapiro, D (2004). Hierarchical Skills and Cognitive Architecture. *In: Proceedings of the 26th Annual Conference of the Cognitive Science Society*, p. 779–784.

[12] Langley, P., Choi, D (2006). Learning Recursive Control Programs from Problem Solving. *The Journal of Machine*

Learning Research V. 7. 493–518.

[13] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C. and Qin, Y (2004). An Integrated Theory of the Mind. *Psychological Review* 111 (4) 1036–1060.

[14] Taatgen, N. A (2005). Modelling Parallelization and Speed Improvement in Skill Acquisition from Dual Tasks to Complex Dynamic Skills. *Cognitive Science* 29. 421–455.

[15] Laird, J. E (2008). Extending the Soar Cognitive Architecture. *In:* Proceedings of the Artificial General Intelligence Conference, p. 224–235.

[16] Nuxoll, M., Laird, J. E (2007). Extending Cognitive Architecture with Episodic Memory. *In:* Proceedings of the 22nd AAAI Conference on Artificial Intelligence.

[17] Lawniczak, A. T., Di Stefano, B. N (2010). Computational Intelligence Based Architecture for Cognitive Agents. *In:* Proceedings of the International Conference on Computational Science.

[18] Langley, P., Lair, J. E., Rogers, S (2009). Cognitive Architectures: Research Issues and Challenges, *Cognitive Systems Research* 10 (2) 141-160.

Biographies

Mr. Blesson Varghese is currently a PhD candidate at the University of Reading, UK. He was the recipient of the Felix Scholarship in 2007 and received his MSc in Network Centered Computing from the University of Reading in 2008. He graduated as the gold medallist in B.Tech Information Technology from Kerala University, India in 2006. His current research spans across disciplines such as swarm robotics, autonomic computing and parallel computing that has resulted in a novel concept for fault-tolerance in high performance computing systems referred to as 'Swarm-Array computing'.

Dr. Gerard McKee is Senior Lecturer in Networked Robotics in the School of Systems Engineering, a Chartered IT Professional (CITP) and Fellow of the British Computer Society (FBCS). He leads the Active Robotics Laboratory which conducts research in the areas of space and networked robotics. His primary research interest is in the area of modular distributed robot architectures and his primary teaching is in the area of robotics and artificial intelligence. Dr McKee is an international leader in Networked Robotics, having conducted research and published papers in the area since the early 1990s.

Prof. Vassil Alexandrov is a Professor in Computational Sciences at the University of Reading, UK. He obtained his MSc in Applied Mathematics from Moscow State University in 1984 and his PhD in parallel computing from the Institute for Parallel Processing at the Bulgarian Academy of Sciences in 1995. His main interests are in the area of simulation and modelling of complex systems, parallel scalable algorithms, collaborative, cluster and grid computing and using the advances in the above mentioned areas for efficiently solving large scale scientific and industrial problems.