Agent and Swarm Views of Cognition in Swarm-Array Computing

Blesson Varghese and Gerard McKee
School of Systems Engineering, University of Reading, Whiteknights Campus
Reading, Berkshire, United Kingdom, RG6 6AY
Email: b.varghese@student.reading.ac.uk, g.t.mckee@reading.ac.uk

***Abstract*—The current state of work in 'Swarm-array computing' requires the theoretical concepts proposed in the framework to be formalised. As a preliminary effort to this end, a recently proposed computational intelligence based hierarchical layered architecture for cognitive agents is mapped onto the intelligent agent based approach of swarm-array computing. The cognitive capabilities of two components of the intelligent agent based approach, namely an agent and a swarm, are considered and the components view of perception, reasoning, judging, response and learning are presented. The layered cognitive architecture maps well onto the microscopic or agent level than on the macroscopic or swarm level of the intelligent agent approach.**

***Keywords*-cognitive layered architecture, intelligent agents, swarm-array computing.**

I. Introduction

Research pursued in the field of cognitive agent architectures has explored the development and deployment of agents with cognitive capabilities in a computing environment. The interest in cognitive agents has increased rapidly in recent times since cognitive agents are not merely reflexive agents and if employed in parallel computing systems can provide solutions to a wide variety of scientific problems requiring memory, reasoning, and problem solving capabilities.

A recent effort to enhance the fault tolerance of large scale parallel computing systems incorporated under the swarm-array computing framework, referred to as the intelligent agent approach is one such application that can benefit from cognitive agent architectures. The current state of work requires the theoretical concepts proposed in the swarm-array computing framework to be mapped onto cognitive architectures, hence formalising the framework as layers. To this end, a layered cognitive agent architecture is required.

After an extensive survey of literature it was noted that research in cognitive architectures focused on component based architectures with minimal effort towards layered cognitive architectures. However, one recent research reported a hierarchical and layered architecture for cognitive agents [1]. The work reported in this paper is motivated towards mapping the layered architecture for cognitive agents onto the intelligent agent based approach of swarm-array computing.

The remainder of this paper is organised as follows. Section II presents the related work in the area of cognitive agent architectures. Section III considers the swarm-array computing framework, particularly the intelligent agent based approach. Section IV considers the layered cognitive agent architecture. Section V presents the formalised framework of swarm-array computing by mapping the layered cognitive agent architecture onto the intelligent agent based approach. Section VI performs a qualitative evaluation of the cognitive architecture. Section VII concludes this paper with a discussion of and consideration of future work.

II. Related Work

Among the wide variety of cognitive agent architectures presented in literature, a few relevant architectures, namely the ECLAIR, LIDA and ICARUS which are relatively recent cognitive architectures and ACT-R and Soar which are architectures that have undergone a greater development cycle are briefly reviewed in this section.

ECLAIR, otherwise known as the Engine for Composable Logical Agents with intuitive Reorganization is a recent architecture with emphasis on adaptation and learning [2]. One feature of the ECLAIR model is that it handles unknown inputs to the model by processing them as if they were known to the system. The modules of the architecture enable stimulus or perception of the world, agent awareness, two types of agent behaviours, namely reflex and plan-based, adaptation and decision making.

LIDA, the abbreviation for Learning Intelligent Distribution Agent, is a more recent cognitive agent architecture based on deriving a working model of machine consciousness [3]. An associative memory which provides a perceptual knowledge base, episodic memory for long term storage of autobiographical and semantic memory, functional consciousness that plays the role of a daemon watching for an appropriate condition for acting, procedural memory which is a graph based memory for representing an action and its context and result, and a module for high level action selection of feelings and emotions form the major components of the LIDA architecture. Multiple learning mechanisms is another feature of the LIDA architecture.

ICARUS another recent cognitive agent architecture is based on conceptual inference and skill execution, which are two approaches of handling knowledge in the architecture [4]. By conceptual inference an agent understands its state and situation by inferring from percepts and beliefs while by skill execution an agent achieves goals by decomposing them

 ISBN: 978-1-61208-108-3

into ordered subgoals. Other features of the architecture include goal selection, means-end problem solving and skill learning.

ACT-R is a cognitive agent architecture that aims to model human behaviour. The architecture comprises six sensory modules and each responsible for vision based processing, executing actions, achieving goals, long-term declarative knowledge, relational declarative knowledge and short-term memory [5]. The model is also capable of learning and updating its knowledge.

Soar is a goal oriented cognitive agent based architecture and represents long-term knowledge in the form of production rules, episodic memory and semantic memory [6]. Multiple learning mechanisms are implemented in the Soar architecture. For example, procedural long term knowledge is acquired through reinforcement learning while declarative knowledge is acquired by episodic and semantic learning.

The agent architectures presented above do not follow a layered architecture. Due to the modular or component based structure of sub systems in the above architectures, the modelling of communication and interaction between the agents tend to be taxing when compared to agent interaction and communication that could be modelled in a layered approach. Further, layered approaches enable the incorporation of additional sandwich layers for extending existing architectures by minimal modifications of the supporting layers.

Clearly there is a need for agent architectures to be developed such that they are layered. Research in the direction of developing layered cognitive architectures are sparse. However, one recent research reported in [1] has proposed a hierarchical five layered architecture for cognitive agents. The layers of the architecture are based on the sequence of activities that contribute to the cognitive capabilities of an agent.

In the next sections, we explore how the layered architecture for cognitive agents considered above can be mapped onto an approach in swarm-array computing, namely intelligent agents. The purpose of mapping the layered architecture onto the swarm-array computing approach is a part of the effort made towards formalising the theoretical concepts of the swarm-array computing framework, which is considered in the next section.

III. Swarm-Array Computing Framework

Research in swarm-array computing has progressed in the direction of applying autonomic computing concepts to large-scale distributed parallel computing systems, thereby improving the fault tolerance of parallel computing systems. The framework for parallel computing deals with constituents, namely the computing platform, the problem/task to be executed, the landscape and the swarm. Moreover, three approaches that bind these constituents, namely the intelligent agent based, intelligent core based and a hybrid approach are proposed.

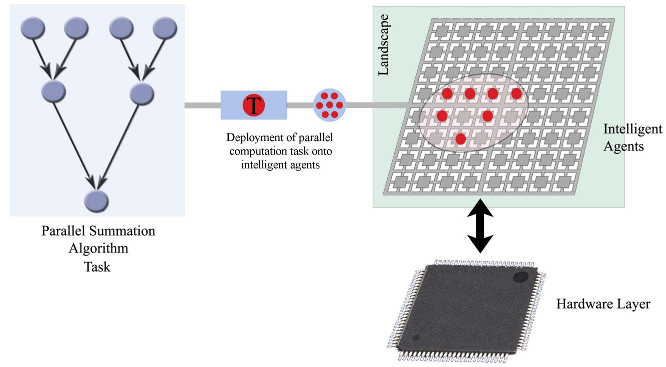

Figure 1. Illustration of the Intelligent Agent Approach in Swarm-Array Computing

In this section, the current state of work of the intelligent agent based approach of swarm-array computing is presented. Parallel reduction [7], a class of algorithms which are of importance and employed in a variety of applications in the high performance computing domain has been considered in the intelligent agent based approach. These algorithms are based on tree structures and figure 1, left, shows an example. The data flows from the leaves of the tree towards the root and at each intermediate node the converging data input is transformed into a result that is passed forward to the next intermediate node. The interconnection of a node in the tree represents its dependencies and the complexity of communication and coordination between the nodes also increases with its dependencies.

In the intelligent agent based approach, when a parallel reduction algorithm needs to be run on a high performance computing platform each node of the tree is scheduled onto a separate computing node. Since these computing nodes are susceptible to failures, there is a need to deal with the isolation of faults. Traditional methods such as checkpointing are challenged by drawbacks and reduce the efficiency of high performance computing systems [8][9].

However, the efficiency can be improved if the algorithm is self-managing such that if a node is about to fail the component of the algorithm can be moved off the node and the input and output dependencies re-established on another node. To incorporate this level of intelligence in the algorithm it would be appropriate to implement agent-like intelligence whereby a computing node can be monitored and a component moved if a failure is anticipated.

In the intelligent agent based approach, the parallel components of the reduction algorithm are mapped onto agents, such that the algorithm in effect is a payload to the set of agents. Figure 1, middle, shows the swarm with its payload. The set of agents are intelligent due to a few cognitive capabilities that they possess. Further, the set of agents

 ISBN: 978-1-61208-108-3

which carry the payload onto the computing nodes can be viewed as a robot swarm and the array of computing nodes can be viewed as a landscape. The robot swarm can then move over this terrain to find a suitable area to execute the algorithm that is mapped onto them. Moreover, if one of the nodes on which the swarm is located fails, a local adjustment can be made by the swarm agent relocating to a nearby part of the landscape and re-instantiating its dependencies, hence offers the potential to improve fault tolerance by minimizing human intervention as in traditional fault isolation methods, and therefore increase the efficiency of high performance computing systems.

These concepts have been investigated practically through both a simulation and implementation [10][11]. The implementation employed a computer cluster with thirty three compute nodes and one head node. A Message Passing Interface (MPI) [12] implementation, namely Open Message passing interface (OpenMPI) [13] was used as the middleware for the implementations. A parallel summation algorithm with fifteen nodes was implemented using both the classical approach and the swarm-array computing approach. The implementation of the approach followed a layered structure such that an abstraction layer was implemented over the actual hardware layer. The agents traversed on the abstraction layer as shown in figure 1, right. The results obtained from the implementations proved that the swarm-array computing approach improved fault tolerance as measured by the mean time taken for reinstatement of the algorithm if a node failed.

The current state of work of the intelligent agent based approach in swarm-array computing is as described above. The approach meets the aims for which it has been proposed yet lacks the formalisation of an agent architecture. Hence, a direction for progressing research would be to investigate an appropriate agent architecture that can be mapped onto the intelligent agents in swarm-array computing.

The requirements for the agents in the intelligent agent approach of swarm-array computing need to be considered before looking at cognitive agent architectures. There are atleast four requirements that agents in the intelligent agent approach need to meet so as to demonstrate cognitive capabilities, namely perception, reasoning, judging, responding and learning.

Firstly, an agent needs to be aware of its environment which includes both the computing cores on which it can carry a task onto and other agents in its vicinity and agents with which it interacts or shares information. Secondly, an agent needs to situate itself on a computing core that may not fail soon and can provide necessary and sufficient consistency in executing the task. Thirdly, an agent needs to predict core failures by consistent monitoring (for example, heat dissipation of the cores can be used to predict failures). Fourthly, an agent needs to be capable of shifting gracefully from one computing core to another, without causing interruption to the state of execution, and notifying other interacting agents in the system when a core on which a sub-task being executed is predicted to fail.

IV. Layered Cognitive Agent Architecture

The need to formalise the existing body of work in swarm-array computing requires the mapping of a generic cognitive agent architecture onto the approaches in swarm-array computing. The layered structure followed in the swarm-array computing approaches including the intelligent agent based approach of interest in this paper hence requires a layered cognitive architecture so that the approach can be formalised.

Recently a Computational Intelligence based Architecture for Cognitive Agents, referred to as CIACA in this paper, has been proposed and reported in [1]. Though the architecture cites examples of activities from highway traffic modelling, the architecture aims to formalise the sequence of activities, namely perceiving, reasoning, judging, responding and learning an agent needs to be capable of performing so that it is cognitive. One notable feature of the architecture is that it follows a hierarchical and layered structure, therefore of interest in the context of swarm-array computing. A perceptual layer, a reasoning layer, a judging layer, a response layer and a learning layer are the five layers constituting the architecture and is shown in figure 2.

The perceptual layer aims to perceive information from the environment by sensing and from information provided by other interacting agents [1]. 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 information by perception.

The reasoning layer supports coherent and logical thinking by obtaining information from the perceptual layer and processing the information using existing knowledge [1]. Fuzzy logic is suggested as a useful tool for reasoning which is a component of the natural thinking process performed unconsciously by humans. For example, humans can 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 information to make decisions at the reasoning layer [1]. For example, if a vision system is used for parking a car between two parked vehicles, then information irrelevant for successful parking obtained through the vision system needs to be eliminated. Feature extraction through edge and corner detection algorithms may prove useful.

The response layer instructs response commands to the perceptual layer after applying rules to the information obtained from the judging layer [1]. For example, move ahead, accelerate or decelerate and switch lane are response commands. Additional reasoning and judging capabilities

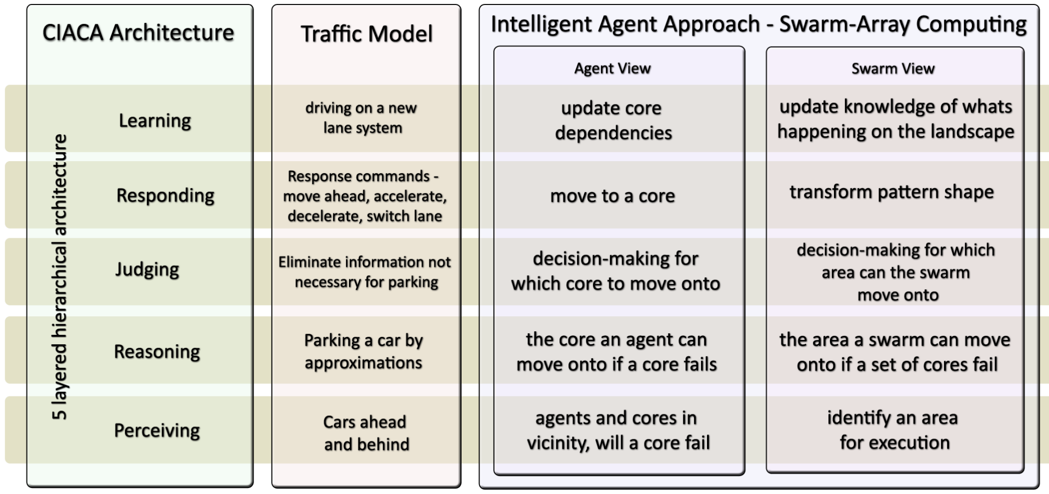

Figure 2. Illustration of the CIACA architecture and the traffic model, and the Agent and Swarm views in the Intelligent Agent Approach of Swarm-Array Computing

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. More sophisticated learning such as extended learning by different generation of agents leading towards evolution of knowledge can also be implemented in this layer.

The above layers meet the requirements for achieving an almost fully functional cognitive architecture. Further, the layers sufficiently formalise the sequence of activities of an agent in a hierarchical layered sequence such that additional computational strategies can be implemented in these layers. Moreover, the layered approach can meet an agent's requirement for demonstrating cognitive capabilities. Therefore, the concepts of the CIACA architecture are chosen to be mapped onto the intelligent agent based approach of swarm-array computing approach.

V. Mapping the Layered Architecture for Intelligent Agent Approach

An efficient mapping of a cognitive architecture onto the intelligent agent approach can be obtained only by considering different views or perspectives of the components, namely an agent, the swarm of agents, a landscape, multiple landscapes, that contribute to the approach. The perception, reasoning, judging, response and learning capabilities of an agent and the swarm will be explored in this section. The landscape and multiple landscape view, more applicable for the intelligent core based approach of swarm-array computing, will be reported elsewhere.

A. 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 and is shown in figure 2. 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 the judging 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.

B. Swarm View

Having considered the capabilities of an agent, namely perceiving, reasoning, judging, responding and learning, above, there is also a need to consider how a group of agents or a swarm comprising 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, as shown in figure 2, 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

 ISBN: 978-1-61208-108-3

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 as seen in the swarm view than on the microscopic level as seen in the agent view.

VI. Evaluating the CIACA Architecture

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 in [14] 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.

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.

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.

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) 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.

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.

VII. Conclusions

The work reported in this paper has aimed to formalise the intelligent agent based approach in swarm-array computing by mapping a layered cognitive architecture, namely the CIACA architecture, onto the intelligent agent approach. Primarily, the conceptual aspects of such a mapping has been presented in this paper. An agent view and swarm view of perception, reasoning, judging, response and learning in the swarm-array computing framework has been presented. A comparative evaluation of the mapping using the cognitive agent architecture against a set of general criteria is performed.

Future work will aim to formalise the intelligent core based swarm-array computing approach using the cognitive layered architecture. Immediate efforts will be also made to consider the intelligent core based approach for exploring the landscape and multiple landscape views.

References

[1] A. T. Lawniczak and B. N. Di Stefano, "Computational Intelligence Based Architecture for Cognitive Agents", in the Proceedings of he International Conference on Computational Science, 2010.

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

[3] U. Ramamurthy, B. Baars, S. K. D'Mello and S. Franklin, "LIDA: A Working Model of Cognition" in the Proceedings of the 7th International Conference on Cognitive Modeling, 2006, pp 244–249.

[4] P. Langley, K. Cummings, and D. Shapiro, "Hierarchical Skills and Cognitive Architecture" in the Proceedings of the 26th Annual Conference of the Cognitive Science Society, 2004, pp. 779–784.

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

[6] J. E. Laird, "Extending the Soar Cognitive Architecture" in the Proceedings of the Artificial General Intelligence Conference, 2008, pp. 224–235.

[7] J. Dongarra, I. Foster, G. C. Fox, W. Gropp, K. Kennedy, L. Torczon and A. White, "The Sourcebook of Parallel Computing", Morgan Kaufmann Publishers, 2003.

[8] J. P. Walters and V. Chaudhary, "Replication-Based Fault Tolerance for MPI Applications" in the IEEE Transactions on Parallel and Distributed Systems, Vol. 20, No. 7, July 2009, pp. 997–1010.

[9] X. Yang, Y. Du, P. Wang, H. Fu and J. Jia, "FTPA: Supporting Fault-Tolerant Parallel Computing through Parallel Recomputing" in the IEEE Transactions on Parallel and Distributed Systems, Vol. 20, Issue 10, October 2009, pp. 1471–1486.

[10] B. Varghese and G. T. McKee, "Can Space Applications Benefit from Intelligent Agents?" in the Proceedings of the 3rd International ICST Conference on Autonomic Computing and Communication Systems, Limassol, Cyprus, 2009 and in the Proceedings of AUTONOMICS 2009, LNICST 23, 2009, pp. 197–207.

[11] B. Varghese, G.T. McKee and V. N. Alexandrov, "A Cluster-Based Implementation of a Fault-Tolerant Parallel Reduction Algorithm Using Swarm-Array Computing" in the Proceedings of the 6th International Conference on Autonomic and Autonomous Systems, Cancun, Mexico, 2010, pp. 30–36.

[12] W. Gropp, E. Lusk and A. Skjullum, "Using MPI-2: Advanced Features of the Message Passing Interface", MIT Press, 1999.

[13] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, T. S. Woodall, "Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation" in the Proceedings of the 11th European PVM/MPI Users' Group Meeting, Budapest, Hungary, 2004, pp. 97–104.

[14] P. Langley, J. E. Lair and S. Rogers, "Cognitive Architectures: Research Issues and Challenges" in Cognitive Systems Research, Volume 10, Issue 2, 2009, pp. 141–160.