01 Jan 2007

Understanding Behavior of System of Systems Through Computational Intelligence Techniques

Nil H. Kilicay

Cihan H. Dagli
Missouri University of Science and Technology, dagli@mst.edu

2007 1st Annual IEEE Systems Conference
Waikiki Beach, Honolulu, Hawaii, USA April 9-12, 2007

UNDERSTANDING BEHAVIOR OF SYSTEM OF SYSTEMS THROUGH COMPUTATIONAL INTELLIGENCE TECHNIQUES

Cihan Dagli
University of Missouri-Rolla
Rolla, MO 65409
573-341-4374
dagli@umr.edu

Nil Kilicay
University of Missouri-Rolla
Rolla, MO 65409
573-341-6556
nkilicay@umr.edu

Abstract – The world is facing an increasing level of systems integration leading towards Systems of Systems (SoS) that adapt to changing environmental conditions. The number of connections between components, the diversity of the components and the way the components are organized can lead to different emergent system behavior. Therefore, the need to focus on overall system behavior is becoming an unavoidable issue. The problem is to develop methodologies appropriate for better understanding behavior of system of systems before the design and implementation phase. This paper focuses on computational intelligence techniques used for analysis of complex adaptive systems with the aim of identifying areas that need methodology customization for SoS analysis.

INTRODUCTION

In today's world, business and government applications require integrated systems that exhibit intelligent behavior. Their success depends on the successful interaction between different groups of systems together. Conventionally, the style of operation for businesses and government was to develop or build what they can do and subcontract when they did not have the capabilities. Now, the operation style is to be the lead system integrator where business or government gets the best systems the industry develops and focuses on system engineering, integration, planning and control to provide a System of Systems.

System of systems describes the interaction between different independent and complex systems in order to achieve a common goal. There are many definitions of SoS depending on the application area and focus [2], [11], [14].Future Combat Systems (FCS), NATO, trans-national virtual enterprises, intelligent transportation systems are some of the networked systems that we are observing in governments and commercial enterprises. These networked systems consist of people, organizations, cultures, activities and interrelationships. The semi-autonomous systems (people, organizations) are integrated through cooperative arrangements. These systems are referred to as network-centric systems.

In System of system analysis, the architecture efforts are focused on the evolution of the existing communications and processing systems, moving towards the creation of an integrated system that can provide a seamless physical, information and social network. This brings the focus on understanding the system level behavior emerging from these sub-systems. It is feasible to understand any System of Systems as an artificial complex adaptive system [5]. The relation of SoS characteristics and CAS characteristics are outlined in [5]. Computational intelligence tools have been successfully used in analysis of Complex Adaptive Systems. Since System of Systems is collections of several Complex Adaptive Systems, we can utilize these tools for analysis of SoS behavior. Therefore, this paper aims to review some of the computational intelligence tools that are potentially suitable for analysis of SoS. The next section discusses the importance of architecture development in SoS and identifies several challenges associated with SoS analysis. The following section reviews the computational intelligence tools that system architects can utilize for their analysis. This section also provides several studies conducted in Smart Engineering Systems Lab at University of

Missouri-Rolla. These studies share similar characteristics as System of Systems. Finally, methodology customization for System of System analysis is discussed.

COMPLEX SYSTEMS ARCHITECTING

Complex systems architecting is an attempt to integrate complex meta-systems. Component systems operate independently, but there is some degree of interdependence between systems in order for SoS to achieve its mission. System of system and its components are also distributed, adaptive, cooperative and competitive at the same time. Therefore, system architects should consider both the complexity associated with the meta-system and the complexity of the potential component systems.

In the SoS environment, architecture has more influence on requirements than it does in an environment dominated by one complex system. In a complex system, architecture is the implementation solution for the requirements. However, in a SoS environment the architectural constraints imposed by existing systems can have a major influence on overall capabilities, objectives and requirements. Therefore, architecture becomes more important in defining the problem [4].

Systems of systems are open systems where a SoS does not have fixed and stable boundaries. All component systems need a physical global interface to function. Global Information Grid (GIG) is seamless communications architecture and forms the basic interface for creating meta-architectures of SoS. A Global Information Grid represents the system formed by the distributed collections of electronic capabilities that are managed and coordinated to support some sort of enterprise (virtual organization). Different independent systems are connected to the GIG to create a network-centric architecture. Therefore, an evolving physical architecture is created by connecting systems to GIG. This net-centric architecture is also evolving to meet the changes in requirements and objectives. A dynamically changing meta-architecture for System of Systems can be defined as a collection of different systems that are readily available to be plugged into the evolvable net-centric architecture. Figure 1 illustrates this meta-architecture. It is the dynamically changing architecture that creates the best net-centric systems not the data that passes through it, although it is a necessity for the system to function. Development and existence of SoS is evolutionary and the evolutionary architecture leads to emergent behaviors and capabilities.

Key performance parameters for a network centric architecture are joint interoperability, adaptable, flexible, modular and robust networked operations, as well as sustainability, reliability, affordability and survivability. Interoperability helps to create meta-architectures that can adapt to changes. However, achieving interoperability is a challenge. Architectural frameworks, technical architectures and levels of information system interoperability are some of the interoperability enablers. However, there are missing linkages between these enablers and system engineering processes. There is a need for better upper level descriptive and analysis frameworks and tools for SoS analysis so that the missing linkages between interoperability enablers and system engineering processes can be identified.

Seamless integration and dynamic adaptation to changing environments are common characteristics. These characteristics are true for both defense and commercial systems. Resulting systems are complex; their behavior can be understood through Computational Intelligence, Artificial Life approaches and Complexity Theory. Following section reviews some of the computational intelligence tools that are suitable for SoS analysis.

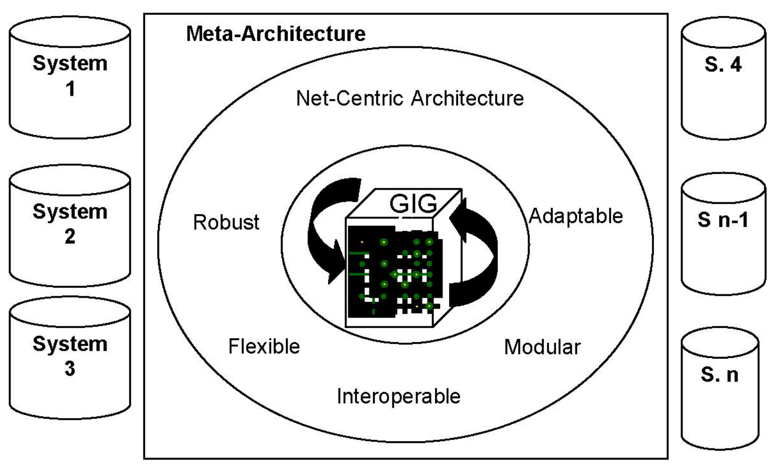

Figure 1: Dynamically changing meta-architecture for System of Systems

ARCHITECT'S TOOL BOX: THE ROLE OF COMPUTATIONAL INTELLIGENCE

Multi-agent systems, distributed artificial intelligence focus on system design whereas evolutionary modeling, swarm intelligence focus on system behavior. The intersection of these two views provides a full representation for SoS analysis. Following sub-sections briefly provide some of the tools that are essential for SoS analysis.

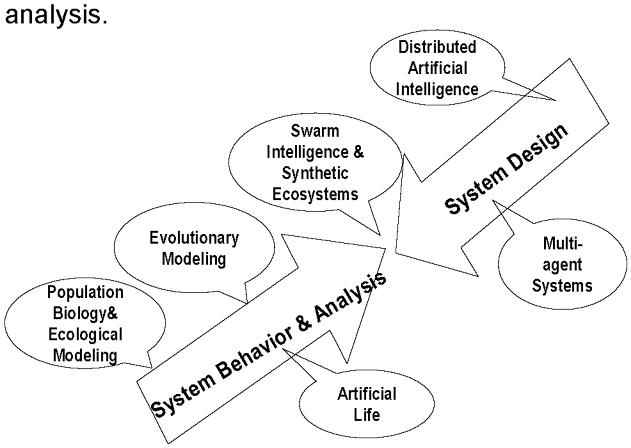

Figure 2: Distributed systems modeling research

Cognitive Agent Architectures

Cognitive architectures seek to understand the building blocks of human cognition. Cognitive architectures are designed to exhibit diverse set of tasks across many application domains. They have the capability to exhibit intelligent behavior at the systems level rather than at the component level designed for specialized tasks.

Since most problems are not purely rational or purely reactive, hybrid architectures in the form of layers are presented. These architectures have several layers to deal with different level of abstractions. Soar and ACT-R are two hybrid cognitive architectures that support most of the cognitive mechanisms. Soar is developed from artificial intelligence viewpoint; ACT-R is developed out of experimental psychology viewpoint. ACT-R is composed of sensory modules, action modules, intentional module for goals and each module has buffers for short- term memory. The long-term production memory coordinates all the modules in ACR-R [9]. Soar also has a long-term memory consisting of production rules. It also has a semantic memory for holding previous states and an episodic memory holding previously seen facts. Soar has several learning mechanisms such as chunking, reinforcement learning, semantic and episodic learning [8].

H-Cogaff cognitive architecture [15] is another hybrid human like information processing architecture. The H-Cogaff architecture would meet the requirements of a complex adaptive system analysis because it represents a combination of the cognitive architecture and the multi-agent systems conceptual frameworks. It provides a framework for describing different kinds of architectures and sub-architectures. It consists of perception, central-processing and action components. The central-processing component has three tiered sub-architectures which are reactive, deliberative and meta-management mechanisms. The reactive components generate goal seeking reactive behavior, whereas middle layer components enable decision making, planning and deliberative reasoning. The third layer supports monitoring, evaluation and control of internal process in the lower layers.

Cognitive architectures have been used on the front-end analysis portion of systems engineering [10]. Cognitive architectures represent a promising approach to explaining mental processes and human behavior with error generation mechanisms. Cognitive architectures embedded within system architectures are useful in identifying the effect of human errors on the overall system behavior.

Multi-agent Modeling

Systems of systems are decentralized, distributed and dynamic in nature. Consequently, these systems cannot be analyzed successfully by modeling them as single cognitive agents because capacity of an agent is limited by its knowledge and computing resources. Multi-agent modeling by its modular characteristic provides a valuable engineering abstraction for analysis of System of Systems.

It is a loosely coupled network of components (agents) that work together to solve problems that are beyond their individual capabilities. Agents in Multi-agent systems (MAS) contain processes for behavior generation, world modeling, sensory processing and value judgment together with a knowledge database.

Multi-agent studies focus on design aspects of the agents. Some of the major study areas can be described as follows [6]:

- Agent architectures
- Agent-System architectures
- Agent Infrastructures

Agent architecture studies focus on internal architecture of agents such as perception, reasoning and action components. Since multi-agent systems are constructed without any global control, one way to prevent chaotic behavior of the system is to design perception and reasoning into agents. Therefore, cognitive architectures provide the underlying foundations for a multi-agent intelligent system.

Agent-System architecture studies analyze agent interactions and organizational architectures where agents operate and interact under specified environmental constraints. Agent Infrastructure studies focus on interface mechanisms of the multi-agent systems which is mainly communication aspects between agents. These studies try to achieve common agent communication language and protocols, common format for the content of communication and shared ontology between agents.

Multi-agent systems have been successfully used in what-if scenario analysis and system structure analysis for System of Systems such as Future Combat Systems.

Evolutionary modeling

Models of elements of System of Systems deploy all types of evolutionary algorithms. Learning classifier system is a suitable tool for SoS analysis and design because it provides a good combination of exploitation and exploration. The learning classifier system has three main components: the performance component, the reinforcement component, and the discovery component [16]. Unmanned vehicles and autonomous robots are an important component of Future Combat Systems. Learning Classifiers is utilized in [3] to model control mechanisms of robots in unknown environments. In their study, classifier rules consist of obstacle avoidance and target capture behaviors. Their results reveal that robot controller is able to generalize properly thus robots are able to adapt to different environments.

Another important task in SoS environment is data analysis and prediction. There are many challenges for prediction under Distributed Information Sharing (DIS) environment. First of all, no supervised learning technique can handle highly non-stationary evolving system's forecasting. There is no good explanation why historical patterns should be expected to repeat. The enhancement for existing teacher-based learning such as Artificial Neural Network, Fuzzy Association is either too time-consuming or too hard to maintain consistency. Therefore, there are a few solutions that can be applied to solve prediction problems under DIS environment. One solution is that core prediction engine can adopt supervised learning which is assisted by reinforcement learning architecture. Adaptive Critic Designs and Q-learning can be considered as potential reinforcement learning candidates. Supervised learning blocks such as ANNs, Fuzzy Association System can be used to make coarse learning and the "normal" system patterns can be generalized from historical data. After sufficient coarse learning, fine learning is applied which employs reinforcement learning (RL) algorithms. RL agent will mimic system evolution. The offset will be generated to show the effects of "unexpected" events. Figure outlines this approach using Adaptive Critics method which combines supervised learning with reinforcement learning. Figure 3 illustrates this architecture.

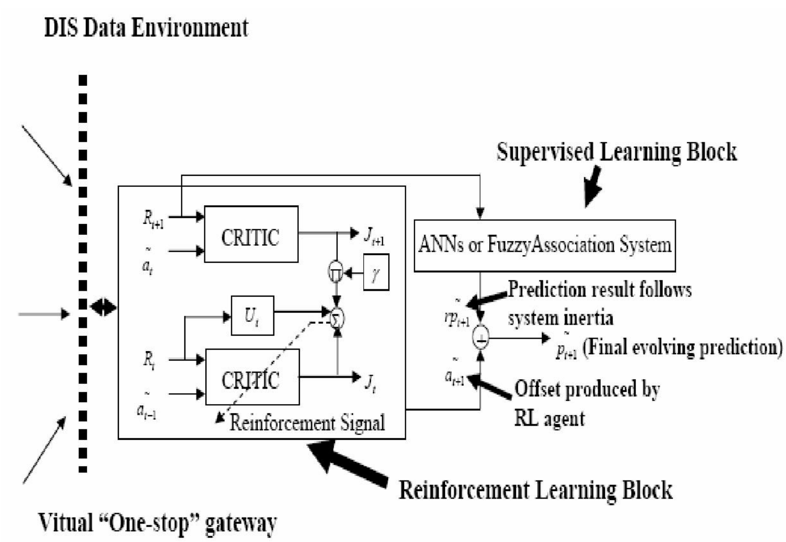

Figure 3: Supervised learning assisted reinforcement learning prediction architecture

Swarm Intelligence

Swarm intelligence specifically focuses on collective intelligence. Centralized and personalized communication is not allowed. Many agents run concurrently performing actions which affect the behavior of other agents. There has to be a well-specified task set for the entire distributed system that requires maximizing some utility function. System of Systems can benefit from swarm intelligence tools because characteristics of swarm intelligence are common with characteristics of System of Systems. Various applications utilized swarm intelligence for modeling and solving system of system problems including swarm robots, swarm routing in communications networks and military swarm scenario modeling [1].

A self-adaptive, swarm-based control model for real-time part routing in a flexible flow shop environment is presented in [13]. The control model is a multi-agent system that exhibits adaptive behavior, which has been inspired from the natural system of the wasp colony. The production problem consists of assigning trucks to paint booths in real-time in a flexible flow shop environment with the objective of throughput maximization and minimization of number of paint flushes accrued by the production system, assuming no a priori knowledge of the color sequence or color distribution of trucks is available. The flexible flow shop environment is a dynamic environment where the product mix is not known prior to production. Table 2 summarizes how swarm intelligence is used for the advanced manufacturing system analysis.

Table 2: Manufacturing System and Swarm Intelligence

	Wasps	Manufacturing System
Goal	Maximize food collection Egg laying Nest building	Maximize throughput Minimize number of setups incurred Minimize average cycle time
Agents	Wasp	Machines
Work Specialization	Gather food or Build nest or Lay eggs	Process particular part types and Avoid additional setups
Force	Force variable of wasp	Remaining processing times, setup
Threshold	Threshold of wasp	Setup requirements
Stimulus	Scent of food	Waiting times of parts

This is a good example of a system where changes in environmental conditions are not known and the system has to adapt quickly to changes.

Dynamic path planning must be used in systems such as Mars exploration systems. Future missions to Mars will have multiple cooperating rovers working on the Martian terrain collecting rock samples and traveling between multiple base stations. Therefore is it important to clearly define the architecture for such a system. An architecture representation for this problem by utilizing swarm intelligence is developed in [7]. One architecture representation is the team of mobile robots that operate in the unknown environment navigating from target to target. The sub-system of mobile robots is decentralized in architecture; i.e. each robot makes its own decisions about the action it has to take. Each robot has data processing and storing capability. Using its map of the environment, each robot navigates itself through workspace and visits the targets. There is no explicit communication between the robots. The only form of communication is in the form of the shared localization map where the pheromone information is updated. The architecture is shown in Figure 4.

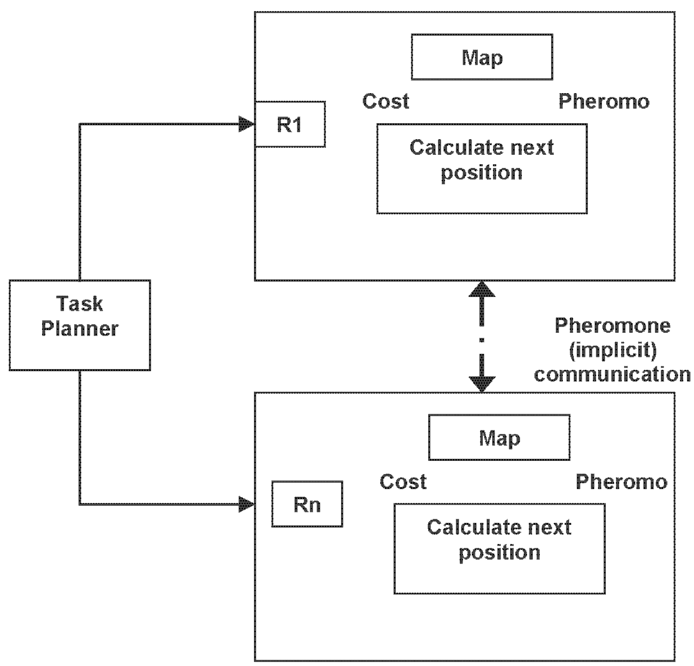

Figure 4: Collaborative robot architecture

METHODOLOGY CUSTOMIZATION FOR SOS ANALSIS

Computational intelligence plays an important role in analysis and design of System of Systems. However, methodology customization for SoS analysis is necessary:

- New modeling and simulation algorithms based on biologically inspired approaches should be added to systems engineer's tool box to cope with modeling and analyzing emerging systems.
- Cognitive architectures can be used as a blue-print for modeling sub-systems of a SoS. Specifically a combination of deliberative and reactive reasoning provides a flexible architecture. Additions of different modules such as learning, long-term memory, short-term memory, imitation provide sub-systems to be evolvable.
- Multi-agent systems combined with cognitive architectures provide a useful methodology for modeling SoS. Relationship between cognitive modeling and multi-agent modeling should be explored more because these models are important in analyzing the emergent system level behavior of SoS analysis.
- Many systems in nature deal with dynamically changing environment by forming swarm architectures. Swarm intelligence provides robust and scaleable solutions for interactions between agents. More focus should be given to the use of swarm intelligence for building SoS. Swarm intelligence can provide a solution to the gap between cognitive agent architectures and multi-agent architectures.
- Both structural and object-oriented analysis is required for comprehension of SoS. Simulation tools that combine various modeling paradigms (discrete, agent-based, system dynamics) should be used in analysis of SoS to capture different behavioral views.
- Ability to learn and evolve new architectures from the previously generated ones, based on systems performance values, need to be incorporated in modeling and simulation process.
- Evolutionary modeling and learning are essential components of System of Systems. Current supervised learning techniques cannot handle rapidly evolving SoS. Reinforcement learning candidates combined with other learning mechanisms such as a supervised learning assisted reinforcement learning architecture or classifier systems are more suitable for modeling system evolution analysis in SoS.
- The concept of “value” needs to be integrated into the systems engineering process. Some of the value oriented perspectives are requirements engineering, architecting, design and development, verification and validation, planning and control, risk management, quality management and people management.

CONCLUSION

SoS engineering is interdisciplinary. It spans through [12]:

- Systems engineering
- Operational analysis
- Decision analysis
- Modeling and simulation
- Value engineering
- Cognitive modeling
- Collaboration-social domain

This paper focused on computational intelligence tools for analysis of SoS. Computational intelligence tools derived from biological systems provide promising analysis methodologies for SoS. These tools have been successfully used in complex system analysis. Modular architectures that utilize one or a combination of these methodologies promise more adaptability and robustness for SoS design and analysis. Swarm intelligence, agent technology and artificial intelligence can be applied, wherever applicable to achieve robustness, reliability, scalability and flexibility in SoS.

Natural systems do not forecast or schedule. They respond quickly, robustly and adaptively. The solution to successful SoS lies on designing systems that can self organize and adapt itself without any outside control. How can we achieve this is the challenge that today’s system engineers face to solve.

References

[1] Bonabeau E., Dorigo M. and Theraulaz G., (1999) *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, New York.

[2] Carlock, P.G. and Fenton R. E. (2001) "System-of-Systems (SoS) Enterprise

Systems for Information-Intensive Organizations" *Systems Engineering*, Vol. 4, No. 4, pp. 242–261.

[3] Cazangi R., Zuben F. and Figueiredo M. (2003) “A Classifier System in Real Applications for Robot Navigation” *The 2003 Congress on Evolutionary Computation*, Vol.1, 8-12 Dec. 2003, pp.574-580.

[4] Cole R. (2006) “The Role of Requirements and Architecture in Systems Engineering*” Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering*, Los Angeles, CA, USA-April 2006.

[5] Correa Y. and Keating C. (2003) “An Approach to Model Formulation for Systems of Systems” *IEEE Conference on Systems, Man and Cybernetics*, Vol.4, Oct. 2003, pp. 3553-3558.

[6] Ferber J., (1999) Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, Addison-Wesley

[7] Gopalakrishnan K., Ramakrishnan S. and Dagli C. (2006) “Optimal Path Planning of Mobile Robot with Multiple Targets Using Ant Colony Optimization” *Intelligent Engineering Systems through Artificial Neural Networks*, Vol. 16, pp.25-30.

[8] Jones R. M. (2004) “An Introduction to Cognitive Architectures for Modeling and Simulation” *Interservice/Industry Training, Simulation and Education Conference*, http://www.soartech.com/pubs/tutorial-iitsec-2004.PDF (last accessed: 03/11/07)

[9] Langley P., Laird J. and Rogers S. (2006) “Cognitive Architectures: Research Issues and Challenges” working paper, http://cll.stanford.edu/~langley/papers/final.arch.pdf (last accessed:03/22/07)

[10] Madni A. M., Sage A. P. and Madni C. (2005) “Infusion of Cognitive Engineering into Systems Engineering Processes and Practices” *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, 10-12 Oct. 2005, pp.960-965.

[11] Maier M. W. (2005) “Research Challenges of System-of-Systems” *IEEE International Conference on Systems Man and Cybernetics*, Vol. 4, 10-12 Oct. 2005, pp. 3149-3154.

[12] Meilich A. (2006) “System of Systems Engineering & Architecture Challenges in a Net Centric Environment” *Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering*, Los Angeles, CA, USA-April 2006.

[13] Meyyappan L., (2006) “Domain Adaptive Control Architecture for Advanced Manufacturing Systems” *Dissertation submitted to University of Missouri-Rolla*, July 2006.

[14] Sage A. P. and Cuppan. C. D., (2001) “On the Systems Engineering and Management of Systems-of-Systems and Federations of Systems” *Information, Knowledge, Systems Management*, Vol. 2, No. 4, pp. 325–45.

[15] Sloman A. (2002) “Architecture-Based Conceptions of Mind” In the *Scope of Logic, Methodology, and Philosophy of Science (Vol II)*, Eds. P. G¨ardenfors, K. Kijania-Placek & J. Wole´nski Kluwer, Dordrecht, Vol. 316, pp 403–427

[16] Wilson W. S., (1995) "Classifier Fitness Based on Accuracy", *Evolutionary Computation*, vol. 3, No. 2, pp. 149-175