# A Methodology for Developing C2 Complex Systems Simulator

**Kamal Mirzaie[1] and Mehdi N. Fesharaki[2]**

[1,2]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[*]**Corresponding author**: Kamal Mirzaie; Email: k.mirzaie@srbiau.ac.ir

**Abstract:** The design and development of an appropriate simulator for C2 complex systems is field of research. Researchers in this field are trying to create simulation models to describe the behavior of C2 complex systems. Like any other application softwares, the design and development of C2 complex systems simulator needs a methodology to make analysis, design, and development more accurate and produce a high quality software simulation. C2 simulator software usually models and simulates C2 complex systems using multi agent approach. However, the multi agent simulator is basically different from the agent based business applications. Therefore for developing former one we need a special methodology. In this paper, a methodology based on Capra cognitive framework for modeling and simulating C2 complex systems has been proposed. This methodology determines the main steps for the analysis, design, and development of C2 simulator. In addition, since the proposed methodology utilizes the basic concepts of complex systems, it may be used for modeling the main concepts and components of other complex systems such as ant colony and stock market.

[Kamal Mirzaie and Mehdi N. Fesharaki. **A Methodology for Developing C2 Complex Systems Simulator**. *Life Sci J* 2012;9(4):377-386] (ISSN:1097-8135). http://www.lifesciencesite.com. 58

**Keywords:** Bio-cognitive, Capra Framework, Command and control (C2), Complex system, Methodology, Multi Agent Systems (MAS), Living Systems.

## 1. Introduction

Today, the design and development of an appropriate simulator for complex systems modeling is one of the fields of research [1-5]. Researchers in this field are trying to extract the appropriate concepts, providing frameworks and computational methods and mechanisms in order to create simulation models to describe the behavior of complex systems. A complex system is made of many components and as a result of the interactions between them, the emergent behavior would appear [6-8].

Living systems such as cells, organizations, society, and the earth in which there is the concept of life and evolution are all examples of complex systems. In modern biocognitive, the main feature of living systems is life and life process is intertwined with cognition in such a way that they are inseparable and synonymous. In this view, all actions and reactions of living systems with the environment are cognitive. The living systems can be biological or social. In living systems, there are interactions between components for survival and evolution. The living system is a complex system that adapts with its surrounding environment throughout its life for survival and evolution. Adaptation means how a system responds to the changing environment and adapts with it [9].

The modeling of complex systems usually leads to a simulation software, in which researchers can simulate and test their models and theories [10]. Multi agent systems (MAS) are a suitable approach for complex systems modeling [11-12]. Today, there are different methods, frameworks, architectures and methodologies that use the concept of MAS. It should be noted that many of the methodologies proposed based on the concept of multi agent systems are mainly for developing agent based business applications. It means that the main motivation for these methodologies is to design and develop the business application that is used in the real environment and not for an artifact which can be a simulation software. On the other hand, few methodologies that are available for designing and developing complex systems simulation are mainly based on ecology and biological viewpoints and have not taken much advantage of engineering principles. For example, Aumann has proposed a methodology for developing simulation models of complex systems [13]. This methodology has three parts: The first part of this methodology places the modeling process in the context of general research planning and thus emphasizes the process of synthesis. This part combines the constituent pieces of knowledge into a unified description of the entities and processes comprising the system to be modeled. The second part involves operationalizing this synthetic description into a composition of smaller models specified over three scalar hierarchical levels. The hierarchical structure of the decomposition also structures the explanations given about system behaviors in that the "mechanism" for a behavior arises at the lower levels while its "purpose" is found at higher levels. The third part of the methodology involves using model assessment to explicitly establish the veracity of the links between the implemented model, the model design specification, and the synthesis.

As the design and development of multi agent simulation software is not completely the same as multi agent business applications, in this paper, we propose a special-purpose methodology for designing and developing C2 (Command and control) complex systems simulator. The proposed methodology is based on Capra's framework. This methodology is used for simulator application development to explore capabilities efficiently for overcoming C2 complex systems. C2 is a concept for military operations. Command refers to the intelligent and, usually, human decision-making process. Control refers to the process of actual execution of the decisions [14-16]. C2 complex systems have been analyzed and modeled with variety of methods and approaches in various papers and reports [17-20]. On the other hand, C2 components cover the entire battlefield, so its modeling can help to examine the behavior of forces and analyze the efficiency of scenarios.

The paper is organized as follow. First, multi-agent systems, agent-oriented methodologies, frameworks and tools are introduced. Then Capra's framework and the proposed methodology are explained. The main ideas of this methodology have been derived from Capra's framework. It should be noted that the term framework in multi agent frameworks and Capra cognitive framework are referring to two different issues. In the former, framework is a software with which multi agent systems are designed and implemented. Multi agent frameworks are special software for developing multi agent systems. In the latter, framework is a kind of cognitive perspective for understanding biological and social phenomena. Having explained the proposed methodology, we will use it to describe modeling and simulation of C2 complex systems.

## 2. Multi agent Systems and Agent Oriented Methodologies

The concept of agent is a powerful and convincing way to describe a software entity that can act with a certain degree of autonomy in performing its duties. Although the agent concept is easily understandable, it has different definitions and interpretations in the papers and books on artificial intelligence. In some definitions, any object can be an agent and some others have narrowed this definition. Russell and Norvig, for example, define agent as an entity that senses and percepts the environment using sensors and based on this perception acts on the environment with actuators [21].

In general, the agent concept is a tool for understanding complex concepts. The concept of multi agent systems is formed when a set of agents interact with each other for problem solving. In other words, such systems solve a problem with a data oriented approach. That is, this approach is based on holistic thinking. However, top-down approach or reductionism can be used to analyze, design and develop multi agent applications [22]. There are various definitions and interpretations for multi agent systems. But, most authors and researchers assume multi agent systems as a system that consists of several agents interact with each other to achieve a certain goal [22-24]. Today, multi agent systems are used as an advanced method for simulation of the real world. This type of simulation provides the test bed for examining theories raised in different areas such as psychology, sociology, economics and military [10].

### 2.1 Agent-oriented Methodologies

Taking advantage of software engineering principles in the design and development of multi agent systems is very important. We need a methodology for analysis, design, and development of multi agent systems. Using the methodology, each of the main phases of software engineering (analysis, design, development, and maintenance) is divided into smaller steps. Nowadays, different agent oriented methodologies have been proposed for analysis, design and implementation of multi agent systems. Some of these methodologies are as follows: Gaia [25], MaSE [26], MESSAGE [27], AUML [28] and Tropos [29].

Gaia is a general methodology that supports both the micro-level (agent structure) and macro-level (agent society and organization structure) of multi agent system development. In the analysis phase, this methodology finds roles and interactions between agents in two separate steps. Roles consist of four attributes: responsibilities, permissions, activities, and protocols. Design phase consists of three steps: The first step is to map roles on to agent types, and then to create the right number of agent instances for each type. The second step is to determine the service model needed to fulfill a role in one or more agents. The final step is to create the acquaintance model for the representation of communication between the agents.

MaSE is similar to Gaia with respect to generality and supporting the application domain. The MaSE methodology is divided into seven phases in a logical pipeline. The first phase, capturing goals, transforms the initial system specification into a structured hierarchy of system goals. The second phase creates use cases and sequence diagrams based on the initial system specification. Refining roles, the third phase, create roles

that are responsible for the goals that are defined in first phase. Generally speaking, each goal is represented by one role, but a set of related goals may also map to one role. The fourth phase, creating agent classes, maps roles to agent classes in an agent class diagram. The fifth phase, constructing conversations, defines a coordination protocol in the form of state diagrams that define the conversation state for interacting agents. In the sixth phase, or assembling agent classes, the internal functionality of agent classes which is based on the type of agent architectures is created. System design, the final phase, creates actual agent instances based on the agent classes. The final result is presented in a deployment diagram.
AUML methodology is based on UML that extends UML for analysis and design of multi agent systems [30]. This methodology has tools for the representation of the agent class, describing the interaction protocol between agents, the internal behavior of an agent, describing the role, and has also a deployment diagram. As AUML methodology takes advantage of UML for modeling, it can be a modeling language as well. In addition to AUML, there are some other special purpose languages based on UML such as AML [31]. AML is used for specifications, modeling, and documentation of multi agent systems.
Tropos is an agent-oriented software methodology created by a group of authors from different universities in Canada and Italy. One of the noticeable differences between Tropos and other methodologies is its strong focus on requirements analysis. This methodology consists of five phases including Early Requirements Analysis, Late Requirements Analysis, Architectural Design, Detailed Design and Implementation [29].

### 2.2 Agent based Frameworks

In order to implementing the theoretical concepts of multi agent systems, it needs an intermediate layer. This layer provides fundamental components to manage the resources of agents. With this layer, developing an agent based applications becomes much simpler, because it provides facilities such as, agent identity, autonomy, communication, mobility and life cycle management. These facilities are supported by the agent based frameworks. Sometimes frameworks are called "platform" [23]. Cougaar , JADE , JACK, and RESTINA are examples of frameworks ([32- 34].
Cougaar is a Java-based architecture for the construction of large-scale distributed agent-based applications. In other words, the goal of Cougaar is to provide developers with a framework to implement large-scale distributed multi agent applications. For this goal, Cougaar supports key functions such as Directory Services, Messaging, and Conflict Management.
JADE framework is a Java based agent platform used for developing multi agent systems. Similar to Cougaar, this framework has facilities such as Directory Services, Messaging, and Mobility. JADE, in other words, is utilized for developing multi agent distributed applications, and as a middleware, provides a set of services and Graphics User Interface (GUI) for debugging and testing the programs. In recent years, this framework has been extensively used in academic and industrial organizations [35].
RETSINA is developed by the Robotics Institute of Carnegie Mellon University. RETSINA is developed in C++ but is platform independent. Agent systems can be run on a number of different platforms and hosts (including handheld systems) and can be implemented in a number of languages. This framework has Directory Services, Messaging, and Semantic Interoperability. In Semantic Interoperability, which is a shared dictionary, the meaning of content is defined by KQML messages.
Agent based frameworks can be used for developing business applications. Thus, if the purpose of designing multi agent systems is developing business applications, agent based frameworks are suitable. These frameworks are mostly used in a real environment.

### 2.3 Multi agent Simulation Toolkits

Agent based frameworks such as JADE and Coagaar are mostly knowledge based systems. Knowledge in these frameworks is represented by one of the methods such as logic, rule, or a method that is supported by a given framework. However, some of the knowledge for modeling and simulation of complex systems such as C2 is implicit and cannot be represented by explicit knowledge. Multi agent frameworks, on the other hand, focus on modeling individual agents rather than social agents. To simulate complex systems with multi agent approach, each agent is described with a set of important and influential parameters. Therefore, simulation toolkits for agent-based modeling and simulation of complex systems are presented. These toolkits can model and simulate wide range of subjects including such as medicine, geology, sociology, and computer science. Examples of multi agent simulation toolkits are SWARM [36], RePast [37], and NetLogo [38].

SWARM is one of the first general purpose multi agent simulation toolkits that has been developed at the Santa Fe Institute to study complex adaptive systems. This toolkit has been designed and developed based on swarm intelligence. The goal of SWARM project was making a general purpose simulator for the simulation of artificial life. SWARM environment is made of many objects. An object in Swarm has three main characteristics: Name, Data, and Rules. An object's Name consists of a unique ID tag that is used to send messages to the object. The Data are whatever local data the user wants to have in the agent (e.g. internal state variables). The Rules are a set of functions that handle any messages that are sent to the object, including the "step" message.

RePast is a free open source toolkit created at the University of Chicago. This toolkit has a set of software libraries for constructing, running, and visualization. RePast borrows many concepts from SWARM agent-based modeling toolkit so they are similar in many cases.

NetLogo was originally designed for simple educational purposes but now many researchers use it as well. Many colleges have used it as a tool for teaching agent-based modeling. NetLogo environment enables exploration of emergent phenomena. It comes with an extensive models library including models in a variety of domains such as economics, physics, biology, chemistry, psychology, system dynamics, and many other natural and social sciences. Furthermore, NetLogo enables the quick and easy authoring of models. It is particularly well suited for modeling complex systems over time. Modelers can give instructions to hundreds or thousands of independent "agents", all operating concurrently. This makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerges from the interaction among many individuals.

In addition to these toolkits, some other simulation tools have been introduced [39].

### 2.4 Agent Based Distillations

Agent based distillations (ABDs) represent an emerging technology within the field of combat simulation. Generally, distillations can be defined as a simulator attempting to model warfare scenarios by implementing a small set of rules that allow agents to adapt within each scenario [40-41]. Distillations are far less detailed than traditional simulations and rely on sensible global behavior to emerge naturally, while traditional models require explicitly this behavior to be programmed. This simplicity gives distillations the characteristics of speed, transparency, ease of configuration, and the ability to use the systems with minimal training. Today, there are a number of ABD systems such as ISSAC [42], EINSTein [43], MANA [44], CROCADILE [40] WISDOM-I, WISDOM-II [45] and ACOMSIM [46].

## 3. Capra Cognitive Framework

The living systems that can be biological or social are assumed by many researchers as complex systems [47]. Hence, theories and frameworks for living systems can be utilized for understanding and modeling complex systems. In a living system, there are interactions among components for survival and evolution. Likewise, a complex system has many components and interaction between them forms an emergency complex behavior. In the study of living systems, we can use Capra cognitive framework. Capra has presented a unique framework for understanding the biological and social phenomena in four perspectives. Three out of four perspectives is about life and the fourth one is meaning. The first perspective is pattern that includes various relations among system components. The organization pattern of a living system defines the relation types among the system components which determines the basic features of the system. Structure, the second perspective, is defined as the material embodiment of system pattern. The Structure of a living organism evolves in interaction with its environment. The third perspective is the life process integrating the pattern and the structure perspectives [9].

When we try to extend new understanding of cognition to the social life, we immediately encounter many misleading phenomena - rules of behavior, values, goals, strategies, intentions, designs and power relations - that often do not have a role in non-human world, but they are essential for human social life. For expanding life to the social domain, meaning perspective is added to three other ones. Thus, we can understand social phenomena from four perspectives: pattern, structure, process, and meaning. Culture, for instance, has created and preserved a network (pattern) of communication (process) with embedded meaning. Material embodiment of culture includes art and literary masterpieces (structure) that transfer meaning from one generation to another.

According to Capra cognitive framework, any complex phenomena can be discussed and studied in four perspectives. In order to close these four perspectives to the terminology of complex systems modeling, we replace "pattern" with "network" and "structure" with "agent".

Pattern perspective is the relationship between components, thus network is a good terminology. Structure is a set of features that evolves during life. These features together will make the agent concept. Therefore, Capra cognitive framework is redefined in four perspectives: network, agent, process, and meaning (Fig. 1).

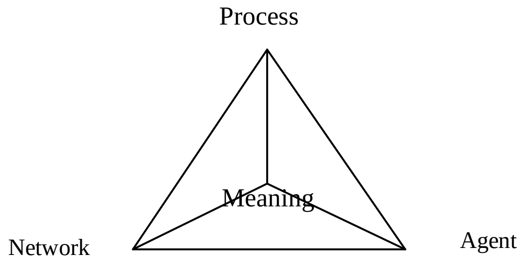


**Fig. 1:** Redefinition of Capra 's Conceptual Framework for Complex System Modeling

## 4. The Proposed Methodology

Generally, methodologies such as Gaia, MaSE, and Tropos are suitable for developing agent based business applications in order to produce high quality software. These applications provide services for users in a real environment.

For designing and developing complex systems simulator, we need a more specific and appropriate methodology because the main purpose of simulator software is complex systems modeling. With this software, the behavior of complex systems is investigated under different conditions or capabilities. So, simulation software (exploratory application) is different from business applications.

One common approach for C2 complex systems modeling is multi agent approach. In this approach, main components of a C2 complex system are mapped to agents. Multi agent systems model and sometimes predict the behavior of C2 complex systems.

In redefinition of Capra cognitive framework (Fig. 1), agent, network, and process were determined as main perspectives for understanding and modeling complex systems. Mapping the meaning on to these three perspectives gives us the main capabilities of C2 complex system.

Our methodology focuses on exploratory application software in C2 complex systems. The proposed methodology has three main phases including analysis, design, and development (Fig. 2). If necessary, these three phases can be repeated many times spirally and the modeling and simulating will improve in each iteration.

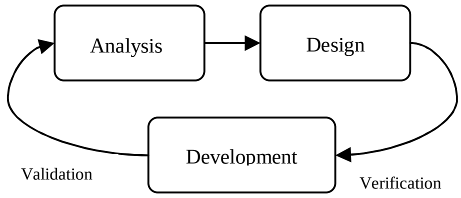


**Fig. 2**: Three main phases of proposed methodology (Spiral)

In the analysis phase of the proposed methodology, C2 complex system is defined and investigated from different aspects. It is usually necessary to acquire specialized knowledge from experts in that field. For this purpose, we can use questionnaires, interviews, or special tables. In defining a problem, three main questions must be answered:

1- What are the main components of a C2 complex system? Main components are autonomous and have the ability to affect and be affected. For example, in the complex system modeling of ant colony, ants are the main components.

2. What kinds of relationships are there between the main components? In other words, how the autonomous components use sensors and actuators to form relationships? Determining the relationships and network types is very important in C2 complex systems since it leads to information sharing and situation awareness. In an ant colony, for instance, communications and relationships are formed by estigmergy and pheromone.

3. What process is going to the run in a C2 complex system that uses main components and their relationships for evolution? The most important part of this process is decision making for survival. For example, in ant colony system, ants are more likely to choose the path with more pheromones. This type of decision making leads to relatively optimal decisions.

In the design phase, the concepts defined in the analysis phase are modeled and designed based on three perspectives including agent, network, and process. But the main question is which perspective is better to start? Regarding that the process perspective combines and intertwines network and agent perspectives to create meaning, therefore it is suggested that the first step in this phase be "process" design and modeling. It should be noted that the main cycle of simulator is based on process. This cycle determines main stages of simulation.

After designing the process, agent is modeled by main properties. In order to have modeling and simulation closer to reality, main properties of agent should be used for modeling. For example, in ant colony simulation, main properties for ant agent can be pheromone production, pheromone detection, and mobility.

What makes agent powerful in multi agent systems is the relation between agents. It means that agents have relation and communication with each other for achieving a given goal. Relationships and interactions between agents depend on the type of agent. In simple reactive agents such as ants, communication is achieved by stigmergy and for cognitive agents such as human by conversation. These relations form networks of agents. So, the third step of design phase is modeling the relation among agents. These relations can be a set of constraints such as hierarchy in organizations, or a set of rules. For example, if agent A is within the vision range of agent B, then agent B sees agent A. This vision relation is a link from agent B to agent A.
Three steps of design phase in our proposed methodology are summarized as follow (Fig. 3):

First step: process design and modeling
Second step: agent design using main properties
Third step: determining and modeling necessary networks for describing the relation and communication between agents.

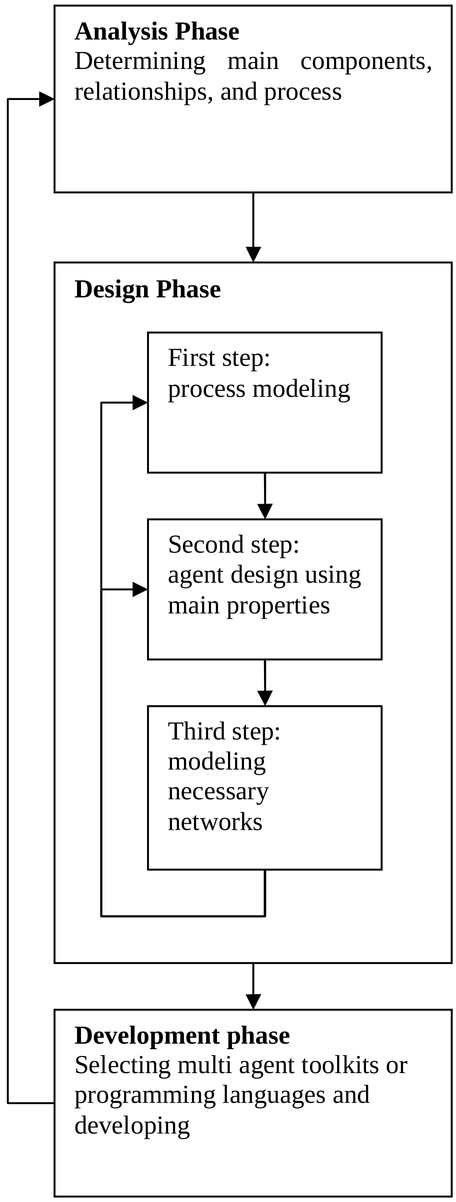


**Fig. 3:** Main phases of the proposed methodology with three steps of design phase

Of course, after performing these three main steps for the first time in the design phase respectively, each step can go back to one or two steps before for revision. For example, when certain networks are determined, some properties of agents might be added, revised, or redefined. In the end, a model is designed based on agent, network, and process perspectives.
After the design phase was repeated enough, development phase begins. In the development phase, we decide whether the existing multi agent toolkits, such as SWARM and NetLogo, are suitable for developing. If not, what

programming languages such as Java, C, Matlab are appropriate for developing simulator?

Since the proposed methodology emphasizes the main concepts and components and relationships between them, it is based on ontology. It means that this methodology determines what concepts, and components should be used for C2 complex systems modeling. In other words, this methodology expresses the meaning perspective (the forth perspective in Fig. 1) as ontology in agent, network, and process perspectives. This ontology determines general principles for simulator architecture. It also offers guidelines for the design and implementation of actual systems. For example, what principles and structures should be used for network design? What features are more important for agent design? What kinds of processes are suitable for modeling a given complex system? In brief, this methodology explores the capabilities. With this explanation, the proposed methodology emphasizes accuracy.

## 5. Developing a C2 Simulator Using the Proposed Methodology

The emergence of NCW theory is due to the Information era. NCW or Network Centric Warfare refers to the organization of services and how to fight in the information age [48]. NCW has made structural and process changes in C2, in a way that C2 is a decentralized and distributed system of command agents. C2 is a key concept in military environments and many researchers assume it as a complex system. C2 complex system has been modeled with different methods [18, 20, and 49].

Some researchers have paid attention to general aspects of C2 and some other to its specific ones. For example, Mofat and Lauren have assumed C2 as fractal process. Thus, all processes of C2 at different levels of command and control have the same and recursive structure [19, 50]. On the other hand, since C2 is a universal concept and covers the entire battlefield, most battlefield modelings are about C2 complex system. Today, many researchers use Complex Adaptive Systems theories (CAS) for modeling the battlefield or C2 complex system. In this approach, battlefield is a CAS that adapts and self-organizes with its environment. ISSAC [42], EINSTein [43], MANA [44], WISDOM-I, WISDOM-II [45] and ACOMSIM [46] are ABDs that have been designed and developed based on this approach. In most of these simulation environments, there has been some attention to agent, network, and process perspectives for modeling either implicitly or explicitly. WISDOM-II is an explicit example. However, for designing and developing these simulators, a special-purpose methodology has not been used.

We have used our proposed methodology for designing and developing the C2 complex system simulator. In the analysis phase, available ABDs were analyzed. Also, specialists in the related field were consulted for better understanding of the subject. In this phase, combatant (people) was detected as the main component of the battlefield. Communications and interactions among combatants are formed by vision, communication, and hierarchical networks. Also, friendship and enmity among people are considered as a relation. Another relationship between people in the battlefield is engagement. In tactic level, OODA is a general process for decision making. In OODA process, each combatant observes the environment (Observe) and from the observation knowledge is made (Orient). The decision is made based on knowledge (Decide), and finally this decision is executed (Act). It is noticeable that OODA is one of the common processes for C2 specification.

In the design phase, the defined concepts are modeled based on agent, network and process perspectives. As a process perspective in redefined Capra cognitive framework (Fig. 1), MOODA [51] was used for developing C2 simulator. MOODA is the extended OODA and models decision making in this simulator. Observe, orientation, decision, and action are four main stages of MOODA process that are included in four modules. These modules make simulator engine. MOODA process has strengthened the learning concept with internal and external feedback loops (Fig. 4).

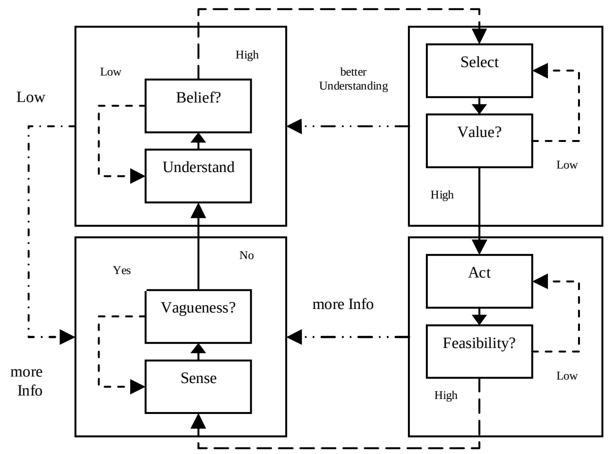


**Fig. 4:** The MOODA loop

Agents that model combatants in battlefield simulation can be represented by vectors of properties. These vectors include personalities and characteristics properties (Fig. 5). Tendency to friends and attacking the enemies are examples of agents' personality properties. The characteristics properties can be radius of vision, firing range, communication range, and rank of agents.

| Personalities properties | Characteristics Properties |
|---|---|

**Fig. 5:** Personalities and characteristics properties of Agents

Agents interact with each other in a battlefield simulator and graph can be suitable to model these interactions. Since in the analysis phase, vision, communication, hierarchy, friendship, enmity, and engagement have been determined as main relations between agents, therefore six graphs model them in the design phase (Fig. 6).

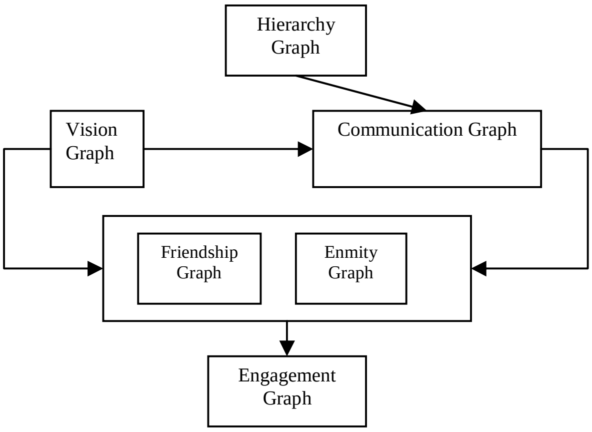


**Fig. 6:** Six graphs for modeling necessary networks

After designing and modeling battlefield in agent, network, and process perspectives, simulator development phase begins. In this phase, MATLAB programming was found suitable, because the relations between agents have been modeled with graphs and graphs in turn can be represented by adjacent matrixes and agents by vectors. It should be noted that the main data structure in MATLAB programming is Matrix.

## 6. Conclusion

Today, complex systems modeling and simulation such as C2 is one of the challenges for researchers in different fields. This modeling often leads to simulator software. Like any other software, C2 simulator developing needs a methodology. As software simulators are different from business applications, we need a special methodology for their design and development. In this paper, a methodology was proposed based on Capra cognitive framework for C2 complex systems modeling and simulation. This methodology suggests what concepts, and components should be used for C2 modeling and what kind of relationship is there between them. That is, the proposed methodology is based on ontology.

The main emphasis of this methodology is on design phase that models C2 complex systems based on agent, network, and process perspectives. It results in an appropriate architectural design for C2 simulator and therefore high quality simulator software would be made. Using the proposed methodology, we have designed and developed C2 complex system simulator and explored agents' decision making for engagement. In addition, this methodology can be used for determining the main concepts and components of modeling other complex systems such as ant colony and stock market.

## References

1- Niazi, M. A., & Hussain, A., “A Novel Agent-Based Simulation Framework for Sensing in Complex Adaptive Environments”, IEEE Sensors Journal, 11, 2, 404–412, 2010.
2- Ilachinski, A. Artificial War: Multiagent-Based Simulation of Combat, Singapore, World Scientific Publishing Company, 2004.
3- Janssen, M. A., & Martens, W. J. M. "Modeling Malaria as a Complex Adaptive System", Artificial Life, 3(3), 213-236, 1997.
4- Mauboussin, M. J., "Revisiting Market Efficiency: The Stock Market as a Complex Adaptive System", Journal of Applied Corporate Finance, 14(4), 47–55, 2002.
5- Yang, A. & Shan, Y., Intelligent Complex Adaptive Systems, IGI Publishing, 2008.
6- Miller, J. H., & Page, S. E., Complex Adaptive Systems: An Introduction to Computational Models of Social Life, Princeton University Press, Princeton, 2007.
7- Mirzaie, K., & Fesharaki, M. N “A Thought Structure for Complex Systems Modeling Based on Modern Cognitive Perspectives”, International Journal of Computer Science Issues, 8, 3, 1, 182-187, 2011.
8- Rzevski, G. “Modelling Large Complex Systems Using Multi-agent Technology”, 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2012, Kyoto, Japan,, 434 – 437, August 8-10, 2012.
9- Capra, F., The Hidden Connections: Integrating The Biological, Cognitive, And Social Dimensions Of Life Into A Science Of Sustainability, Doubleday, 2002.
10- Gilbert, N., & Troitzsch, K. G., Simulation for the Social Scientist, Open University Press, McGraw-Hill Education, Second Edition, 2005.
11- Cannata, N., & Corradini, F., Merelli, E., Omicini, A., & Ricci, A.,”An Agent-oriented Conceptual Framework for Biological Systems Simulation”, Transaction on Computation System Biology, 3, 105-122, 2005.
12- Clymer, J., Simulation Based Engineering of Complex Systems, Wiley-Interscience, 2009

13- Aumann, C. A., "A methodology for developing simulation models of complex systems", Ecological Modelling, Volume 202, Issues 3-4, pp. 385-396, 2007.
14- Alberts, D. S., & Hayes, R. E., Understanding Command and Control, CCRP publication, 2006.
15- Lawson, J. S., "Command and control As a Process", IEEE Control Systems Magazine, 5-12,1981.
16- Lee, T., & Ghosh, S. " A Novel Approach to Asynchronous, Decentralized Decision-Making in Military Command and Control", IEEE Computational Science and Engineering, 3, 4, 69-79, 1996.
17- Li, X. et al., "Cognitive Agents for Modeling Military Command Entities", Proc. 5th IEEE Int. Conf. on Cognitive Informatics (ICCI'06), pp. 176-184, 2006.
18- Mason, C. R., & Moffat, J., An Agent Architecture for Implementing Command and Control in Military Simulations, IEEE Computer Society Washington, DC, USA, 721–729, 2001.
19- Moffat, J., & Witty, S., "Experimental Validation of Meta-Models for Intelligent Agent in Conflict", Information and Security, An International Journal, vol. 12, No. 1, pp. 95-103, 2003.
20- Stanton, N. A., Baber, C. & Harris, D., Modelling Command and Control: Event Analysis of Systemic Teamwork, Ashgate Publishing Company, 2008.
21- Russell, S., & Norvig, P., Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2009.
22- Weiss, G., Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence, MIT Press, Cambridge, MA, 3rd edition, 2001.
23- Bernon, C. et al., "An Overview of Current Trends in European AOSE Research", Informatica 29, pp. 379-390, 2005.
24- Stone P., & Veloso, M. "Multiagent Systems: A Survey from a Machine Learning Perspective", Autonomous Robotics, Volume 8, Number 3, pp. 345-383, 2000.
25- Wooldridge, M., Jennings, N. R., & Kinny, D., "The Gaia Methodology for Agent-Oriented Analysis and Design", Autonomous Agents and Multi-Agent Systems, Vol. 3, No. 3, pp. 285-312, 2000.
26- DeLoach, S. A., Wood M. F., & Sparkman, C. H., "Multiagent systems engineering", International Journal of Software Engineering and Knowledge Engineering, Vol. 11, No.3, pp. 231-258, 2001.
27- Evans, R., et al., "MESSAGE: Methodology for Engineering Systems of Software Agents", Technical Report P907, EURESCOM, 2001.
28 - Lin, H., Architectural Design of Multi-Agent Systems: Technologies and Techniques, IGI Publishing, 2008
29- Bresciani, P. et al., "TROPOS: An Agent Oriented Software Development Methodology", Autonomous Agents and Multi-Agent Systems, Vol. 2, No. 3, pp. 203–236, 2004.
30- Odell, J., van D. Parunak, H., Bauer, B. (1999). "Extending UML for agents", Available online at: http://www.jamesodell.com/ExtendingUML.pdf
31- Trencansky, I., & Cervenka, R., "Agent Modeling Languages (AML): A Comprehensive Approach to Modeling MAS", Informatica 29, pp. 391-400, 2005.
32- Bellifemine, F., Caire, G., & Greenwood, D., Developing Multi-Agent Systems with JADE, John Wiley and Sons Ltd, 2007.
33- Modi, P. J., et al., "Towards a Reference Model for Agent Based Systems", Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, pp. 1475 – 1482, Hakodate, Japan, 2006
34- Bordini, R. H.,& Hubner, J. F., Wooldridge, M., Programming Multi-Agent Systems in AgentSpeak using Jason, John Wiley and Sons Ltd, 2007
35- Bordini, R. H., et al. "A Survey of Programming Languages and Platforms for Multi-Agent Systems", Informatica 30, pp. 33–44, 2006.
36- Minar, N., et al. "The Swarm Simulation System: a Toolkit for Building Multi-Agent Simulation", Working paper 96-06-042, Santa Fe Institute, 1996.
37- North, M. J., Collier, N. T., & Vos, J. R., "Experiences creating three implementations of the repast agent modeling toolkit", ACM Transactions on Modeling and Computer Simulation, 16(1), pp. 1-25, 2006.
38- Tisue, S., & Wilensky, U., NetLogo: A simple environment for modeling complexity. Paper presented at the International Conference on Complex Systems, Boston, May 16 – 21, 2004.
39- Allan, R.,"Survey of Agent Based Modelling and Simulation Tools", Computational Science and Engineering Department, STFC Daresbury Laboratory, Warrington WA4 4AD, 2009.
40- Easton, A., & Barlow, M.,"CROCADILE: An Agent-based Distillation System Incorporating Aspects of Constructive Simulation", 2002. Available online at: www.siaa.asn.au/get/2395361059.pdf
41- Sprague, K., & Dobias, P., "Behaviour in Simulated Combat: Adaptation and Response to Complex Systems Factors", Defence R&D Canada, Centre for Operational Research and Analysis, DRDC CORA TM 2008-044, November 2008.
42- Ilachinski, A., "Irreducible Semi-Autonomous Adaptive Combat (ISSAC): An artificial life approach to land combat", Research Memorandum, CRM 97-61, Center for Naval Analyses, Alexandria, 1997.
43- Ilachinski, A., "Enhanced ISAAC Neural Simulation Toolkit (EINSTein): An Artificial-Life Laboratory for Exploring Self-Organized Emergence in Land Combat (U) ", Beta-Test User's Guide, CIM 610.10, Center for Naval Analyses,1999.

44- Lauren, M. K., & Stephen, R. T., "MANA: Map Aware Non-uniform Automata", A New Zealand approach to scenario modeling, Journal of Battlefield Technology 5(1), pp. 27-31, 2002.
45- Yang, A. "A Networked Multi-Agent Combat Model: Emergence Explained", PhD Thesis, University of New South Wales, Australian Defiance Force Academy, 2006.
46- Cil, I., & Mala, M.,A Multi-agent Architecture for Modelling and Simulation of Small Military Unit Combat in Asymmetric Warfare, Expert Systems with Applications, 37, 1331-1343, 2010.
47- Banirostam, T. & Fesharaki, M. N., "A New Approach for Biological Complex Adaptive System Modeling and Simulation", Life Science Journal, 9(3) ,2257-2263, 2012.
48- Alberts, D. S., Garstka, J. J., Stein, F.P., Network Centric Warfare: Developing and Leveraging Information Superiority, 2nd Edition (Revised), CCRP publication, 2002.
49- Moffat, J., Witty, S., "Experimental Validation of Meta-Models for Intelligent Agent in Conflict", Information and Security, An International Journal, vol. 12, No. 1, pp. 95-103, 2003.
50- Lauren, M. K., Smith, J. M., Moffat, J. & Perry, N. D., "Using the Fractal Attrition Equation to Construct a Meta-model of the MANA Cellular Automat Combat Model", The Technical Cooperation Program (TTCP), November 2005.
51- Breton, R., Rousseau, R., "The M-OODA: Modelling of the OODA loop as a modular functional system", Defence R&D Canada – Valcartier, Technical Memorandum, DRDC Valcartier TM 2008-130, 2008.

10/30/2012