

Executable Architectures using Cuckoo Search Optimization coupled with OPM and CPN-A module: A new Meta-Architecture Model for FILA-SoS

Siddhartha Agarwal¹, Renzhong Wang¹ and Cihan H. Dagli¹

Abstract

Understanding System of Systems (SoS) requires novel ways to apply systems engineering processes. Acknowledged SoS have recognized objectives, a designated manager and resources for the SoS. The goal of this research is to develop a proof of concept tool suite for Acknowledged SoS systems simulation. This suite is named flexible, intelligent and learning architectures for System of Systems (FILA-SoS). FILA-SoS assists the SoS manager in architecture generation, selection, and implementation working as an aid for decision making. Binary cuckoo search constrained optimization is used to generate meta-architectures which are evaluated by a fuzzy assessor for quality assurance. The architecture is then converted into an executable structure using Object Process Methodology (OPM) and Colored Petri Nets (CPN). A hybrid methodology comprising of OPM and CPN approach is implemented for simulating the acquisition environment. Initial application for a Search and Rescue (SAR) SoS, consisting of 25 individual systems with ten capabilities gave promising results.

Keywords: Acknowledged; Cuckoo Search; CPN; fuzzy; SAR; meta-architectures; negotiation; OPM ; SOS; binary; optimization; simulation

1 Introduction

Flexible models and scalable search approaches are needed to manage the design space of systems of systems (SoS) architecture, capture its emergent behavior, and assist in dealing with the rapidly changing machinery. Four types of SoS that appear in literature are directed, collaborative, virtual and acknowledged. Acknowledged SoS share properties with both collaborative and directed SoS [1]. Acknowledged SoS have documented objectives, a nominated manager, and resources for the SoS. The constituent systems preserve their autonomous ownership, objectives, funding, and development and sustainment approaches. All systems have to work in collaboration to achieve larger goal which they are incapable of achieving just by themselves. Simulation and modeling techniques for Acknowledged SoS are still in their infancy. Our objective in this paper has been to present a module within the larger model called as FILA-SoS that stands for flexible, intelligent and learning architectures for system of systems. FILA is an integrated tool suite in ANYLOGIC* and involves

¹ Engineering Management and Systems Engineering Department, Missouri University of Science and Technology, Rolla, Missouri - {sa265@mst.edu, rwkb4@mst.edu, dagli@mst.edu}

meta-architecture generation for acknowledged SoS using computational intelligence based on key performance attributes specified by the stakeholders [2]. The meta-architectures are a crude form of the final architectures. To realize them in actuality SoS coordinator has to negotiate his demands and resources individually with participating systems [3]. This major issue is resolved by introducing negotiation modules between individual systems and SoS manager based on domain specific information. For further details please refer to the system engineering research center report [4]. The set of issues being negotiated over are defined by the domain of the problem. This paper proposes simulation for ISR (intelligence, surveillance and reconnaissance) scenario. The scenario depicts an area pacification/low level guerilla war situation. The garrison has the systems detailed in section 4 for performing ISR and fire support to patrols and convoys and the area being controlled is about 100 nmi in radius. This SoS includes negotiation attributes as price (value for capability being acquired), performance (task execution capacity) and deadline (delivery date). Executable architectures are generated using a hybrid of Object Process Methodology (OPM) and Colored Petri Nets (CPN) [5]. These executable architectures are useful in providing the much-needed information to the SoS coordinator for assessing the architecture quality and help him in negotiating better. The intended contribution of this paper is as follows. First, a different method of meta-architecture generation based on cuckoo search algorithm combined with fuzzy inference engine is proposed for system architecting. Secondly the non-dominated solutions generated, all of which represent a different architecture, are then made executable through the use of OPM and CPN. Finally, our attempt is to present an integrated acknowledged SoS architecting model called FILA-SoS whose capabilities include extensive multi-level SoS meta architecture generation covering the entire design space, flexible and robust architecture assessment, and final architecture securement through simulated negotiations. This paper is motivated by current lack of understanding of system participation choice on the overall SoS capability. In addition, there is an imminent need for domain independent SoS architecture generation and assessment methodology.

The rest of the paper is organized as follows. Section 2 details related work in other SoS projects and FILA-SoS. Section 3 describes the FILA-SoS model used in this study by detailing different modules and some basic concepts. Section 4 describes our methodology for architecture generation and converting them to executable architectures. In section 5, we analyze the results of the proposed procedures. Finally, Section 6 outlines the conclusions and our plans for future study.

2 Literature review on current and past SoS projects

In this section a brief description of major SoS projects currently being pursued in a variety of domains are discussed. This section will help the reader get an overview of the scope of research being conducted. The descriptions do not necessarily follow any order in which the projected came into inception. DANSE SoS stands for Designing for Adaptation and Evolution in System of Systems [6]. DANSE project addresses the challenging technical, management, and political problems within

organizations. The main features include combining the strengths of several infrastructures and objects present because of advances in communications, sensors and actuating competencies. DANSE is among several projects in SoS funded by the European Commission as part of the Seventh Framework Program. The purpose of the DYMASOS (Dynamic Management of Physically Coupled Systems of Systems) project is to explore methods for the distributed management of large physically connected systems along with distributed autonomous management and global coordination [7]. COMPASS stands for Comprehensive Modelling for Advanced Systems of Systems and aims to develop collaborative research on model-based techniques for developing and maintaining SoS [8]. For example, a flexible and responsive SoS can be developed for emergency management, given the fact that individual systems were not intended for collaboration. T-AREA-SoS (Trans-Atlantic Research and Education Agenda on Systems of Systems) was developed through cooperation between EU-US Systems of Systems (SoS) research [9]. T-AREA-SoS aims to achieve European competitiveness and improve the societal impact through development and management of large complex systems. The CYPHERS project aims at developing an integrated cyber-physical roadmap and strategy for Europe [10]. Its ultimate goal is to combine and expand Europe's capability in embedded and mobile computing as well as in control of networked embedded systems. Some projects that are closely related to CYPHERS are Hycon2: highly-complex and networked control systems, EMSIG: embedded systems special interest group, artist design: European network of excellence on embedded systems design and CPSoS: cyber-physical systems of systems. AMADEOS aims critical systems certification for SoS [11]. Its abbreviation stands for Architecture for Multi-criticality Agile Dependable Evolutionary Open System of Systems. The AMADEOS project emphasizes on evolution, emergence, dependability and security, taking into consideration-embedded devices and the cloud as the projects execution platform. It has three significant objectives namely: to introduce a concept of global time that can be accessed and recognized by all elements of the SoS, ability to explain and formalize SoS evolvability and dynamicity, and handling emerging properties in SoS. The CPSOS is a support action, to be completed in 30 months, that aims at developing a roadmap on research and innovation in engineering and management of cyber-physical systems of systems [12]. CPSOS are cyber-physical systems which exhibit the features of systems of systems. The aim of CPSOS is to study and analyze computing and communication systems that interact with large complex physical systems. Local4Global- project stands for Systems of Systems that act locally for optimizing globally [13]. Its desired goal is to develop, comprehensively test and evaluate in real-life Traffic Systems of Systems (TSoS). In addition, the project needs to generate a generic, integrated and fully functional methodology for TSoS. The optimization method developed so far is demonstrated in two real scenarios: the climate control of a building and optimizing the traffic on a test site in the North of Munich. Another traffic prediction project involving SoS techniques is smarter traffic predictions in collaboration with IBM for the city of Cologne, Germany. COBWEB - Citizens OBServatory WEB - is another project that is funded under the European Union's Seventh Framework Programme (FP 7) for developing community-based environmental systems using innovative and novel earth observations applications [14]. It is a large collaboration of experts from 13 partners and 5 countries. The

projects major aim is to create a platform environment enabling citizens living under the biosphere reserves designated by UNESCO (United Nations Educational, Scientific and Cultural Organization) to collect environmental data using their mobile devices. GEOSS stands for global earth observation system of systems, aims to provide solutions for a number of problems around the world [15]. So far, it has been used in forecasting meningitis outbreaks, guarding biodiversity, and helping in improving climate observations in Africa and Central and South America. The environmental protection agency (EPA) in USA along with Group on Earth Observations (GEO) helps in contribution to the development of GEOSS. The ultimate goal of GEOSS is to provide decision makers with correct and prompt scientific information for advancement of social benefits. Integrated Mobile Security Kit (IMSK) aims at detecting critical situations [16]. It helps to provide quickly an effective deployment of information fused with intelligence on mobile platforms for enhanced security. Some examples of its application are mass events such as football games and terrorism attacks. Lastly, the ministry of economics and technology in Germany sponsors Shared e-Fleet project [17]. It aims at higher utilization of systems electric vehicles so that they can be used commonly and very efficiently.

2.1 Recent work on FILA-SoS

Some of the recent work that has been done in FILA-SoS includes creating meta-architecture consisting of a set of systems and their interconnections using binary particle swarm, genetic algorithms and modular type II fuzzy nets as fitness evaluator [18]. Executable systems architectures are created using CPN models and their simulations contain detailed quantitative information about the performance of a system, such as throughput and processing time. These results support the exploration and detection of structural and dynamic system properties [19]. A bilateral counter offer based negotiation mechanism between SoS manager and individual system is employed involving three attributes namely performance demands from each system, deadline to prepare for participation in SoS, and amount of funds to be received as remuneration from the manager. The system behaviors are anticipated to be basically of three types selfish, cooperative and opportunistic. ANYLOGIC [20] based integrated model is designed to incorporate negotiation strategies and generate multiple responses of the counter offer game between the two parties. A number of negotiation rounds with different system types and SoS coordinator are conducted [21]. This negotiation data is then analyzed through various unsupervised clustering approaches to delineate behaviors. An unsupervised clustering of the difference between offer and counteroffer of attributes by hierarchical and k-means techniques reveal four optimal numbers of clusters present in the data [22].

3 Flexible Intelligent & Learning Architecture for System of Systems-FILA-SoS

The authors propose an integrated model that provides a useful multidisciplinary tool for modeling and simulation of systems of systems in an acquisition environment. FILA-SoS examines, the different mathematical models working in cohesion, to determine SoS characteristics that can help increase the chance of success of complex SoS development efforts. In this disposition, this section presents the following concepts and related mathematical models. The initial inputs to the model include selecting a context and environment for the model. The meta-architecture generation ingredients also include a set consisting of participating systems and their interconnections. Figure 1 shows a process flow diagram of the modeling and simulation of SoS. Multiple negotiation rounds and assessment for the architecture after each negotiation round is termed as an epoch in the cycle. Whereas once the architecture selected after multiple epochs is called the final architecture ready for implementation. This culminates a wave of architecting methodology. The next wave would involve different set of stakeholders, new systems and new capabilities being added to the inputs for meta-architecture generating process. Some systems will be excluded and thereby reducing some capabilities of the SoS.

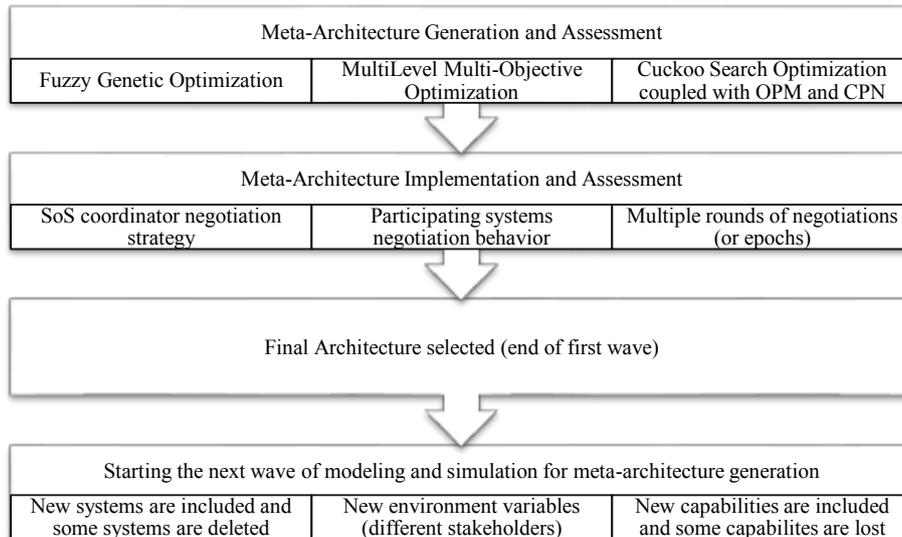


Fig. 1. Process flow of modeling and simulation technique adopted for FILA-SoS.

3.1 *FILA-SoS Model overview*

The SoS achieves the required capability goal by combining existing system capabilities and adding new capabilities. The integrated system model is comprised of

three main elements: SoS acquisition environment, SoS behavior, and Individual system behavior. The following sections present a brief overview of different modules contained within the FILA-SoS.

Meta-Architecture generation. The SoS meta-architecture includes:

- The systems selected.
- The interfaces selected amongst the systems.
- The capabilities possessed by the systems

The three meta-architecture generation models are: multi-level optimization model, fuzzy-genetic optimization model and hybrid particle swarm-fuzzy modular nets model. This paper proposes a new meta-architecture generation methodology based on hybrid binary cuckoo search [23, 24] constrained optimization algorithm and fuzzy logic for architecture generation.

Architecture assessment models. These models are based on multiple stakeholders' point of view and are domain specific. The fuzzy assessor is used to evaluate the fitness of architecture. Fitness function is an amalgamation of multiple key performance attributes (KPA) of architecture, such as affordability, performance, robustness, modularity, net-centricity. These KPAs are developed through guided discussions with stakeholders and subject matter experts (SMEs). The attributes will be domain adjusted and selectable, using guidance from SMEs. Fuzzy membership functions (derived from stakeholder views) describe the degrees of goodness in each attribute area. Fuzzy rules (also derived from stakeholder views) combine attributes into an overall fitness grade. The two types of assessors in FILA-SoS are:

- Fuzzy Type I assessor
- Fuzzy Type II modular nets

Acknowledged system of systems negotiation model. After generating meta-architectures, for their actual implementation there is a need to model negotiations between individual systems and the Acknowledged systems coordinator. The model assumes that there will be bilateral negotiation between individual systems and the SoS coordinator. The idea is to assist Acknowledged SoS coordinator in forming a joint capability. This module provides a contract and convinces the individual systems to join the SoS architecture. The SoS coordinator motivates individual systems to do their tasks well. The SoS negotiation model is based on incentive contracting mechanism inspired by game theory.

Negotiation Protocol. The steps in order of occurrence of the protocol are listed below:

1. Send connectivity request to individual systems
2. Individual systems generate counter-offer values based on its behavioral characteristic
 - Selfish
 - Opportunistic
 - Cooperative

3. SoS manager utilizes the incentive based negotiation model to find the optimal strategy for the SoS manager
 - Accept
 - Counter-offer

Individual system behavior models. In an interoperable world, reasoning about counterpart's behavior improves the predictive power of the opponent that is negotiating. The individual systems negotiation strategy is described by three behavioral traits namely selfish, cooperative and opportunistic. Three behaviors that are exhibited by the individual systems while they negotiate are selfish, opportunistic, and semi-cooperative models. A brief outline for each of the models is given below:

–Semi-cooperative fuzzy negotiation: This model of semi-cooperative behavior is based on an agent's preferences and the negotiation length. Each systems agent has two inherent behaviors of cooperativeness which in is referred to as Purposive behavior and behavior driven by unforeseen circumstances which is referred to as Contingent.

–Selfish Linear Optimization: The necessary condition for an individual system to collaborate with the SoS is to obtain nonnegative incremental profit (hence, termed selfish).

–Opportunistic Markov chain: Opportunistic model allows the system to behave selfishly as well as unselfishly (or selflessly) by tweaking certain tunable parameters.

SoS coordinator adaptive negotiation strategy. The authors propose the use of deep belief networks as an unsupervised clustering technique for grouping the negotiation responses from the systems.

Executable System Architecture Modeled by OPM and CPN. A hybrid methodology comprising of object process methodology (OPM) and colored petri nets (CPN) approach is applied for simulating the acquisition environment. Aiding the SoS manager in future decision making through adaptive learning of systems behavior.

4 Methodology of Generating Executable Architectures

The domain specific information required for the architect to proceed is mentioned as a list:

- Constituent system capabilities required are numbered as c_i : $i = \{1, 2, \dots, M\}$.
- Total number of systems that are present is represented by S_s : $s = \{1, 2, \dots, N\}$.
- A systems capability i is represented as $c_i^{S_s}$.
- A systems amount of area coverage is represented by A^{S_s} .
- The systems operating time is Ot^{S_s} .
- The systems frequency of information collection is I^{S_s} .
- The systems communication capacity is CC^{S_s} .
- The systems time taken to recover between operations is T^{S_s} .
- The system type is denoted by an alphabet: $\{A, B, C, D, E, F, G, H, I\}$

The following equations describe the formulation of multi-objective problem with variables and constraints.

$$f(\mathbf{x})^T = \{ f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ \dots \ f_p(\mathbf{x}) \} \quad (1)$$

$$\mathbf{A}(\mathbf{x})^T = \{ a_1(\mathbf{x}) \ a_2(\mathbf{x}) \ \dots \ a_q(\mathbf{x}) \} \quad (2)$$

$$\mathbf{x}^T = \{ x_1 \ x_2 \ \dots \ x_n \} \in \mathbf{X} \quad (3)$$

\mathbf{x} : vector of the variables; f : objective function(s); \mathbf{a} : inequality constraints;

The four objectives in this multi-objective problem are amount of area coverage, operating time, communication capacity, and time taken to recover between operations. These are also the key performance attributes of the SoS and are measured as follows: $A^{SoS} = \sum_{s=1}^N A^{S_s}$, $O^{SoS} = \sum_{s=1}^N Ot^{S_s}$, $CC^{SoS} = \sum_{s=1}^N CC^{S_s}$, $T^{SoS} = \sum_{s=1}^N T^{S_s}$. There are a total of eight linear constraints which make sure that at least one systems of each type is selected for forming the overall architecture. There are five systems of type A called shadow which are tactical unmanned aircraft system (UAS), two systems of type B called gray eagle which are UAS as well, type C define apache helicopters and there are two of them. Similarly, there are two system of type D representing command and control surveillance units, type E has two exploitation control systems, type F has one artillery based system, type G, H, I include four, four, and three systems each and they are comprised of voice/chat systems, line of sight capability and beyond line of sight capability systems.

The first part of the methodology screens the initial list of all systems to a smaller subset. This subset will be combined with OPM and CPNs to produce executable architectures.

A binary evolutionary algorithm called cuckoo search (CS) with multiple objectives is used to solve for obtaining the right set of systems. The objectives here are to maximize area of coverage, maximizing operating time, maximizing the communication capacity and finally minimizing the time between operations. The numbers of parameters necessary to be regulated in CS are much less than genetic algorithms (GA) and particle swarm optimization (PSO) making it more appropriate for an extensive class of the optimization processes. For multi-objective CS constraint optimization problems with K different objectives, the rubrics are stated below:

- Each cuckoo lays an egg at a time, and dumps them in a randomly chosen nest. Nth egg corresponds to the solution in the Nth nest.
- The best nests with high quality of eggs (solutions) will carry over to the next generations.
- The quality of the egg is judged by a fuzzy assessor which incorporates rules for all objectives relating to the overall solution quality.
- A penalty term is subtracted from the overall fitness value of the solution. This penalty term is composed of the sum of all the linear constraints. The concept is that the fitness function decreases as degree of the constraint violation increases.
- Each nest will be abandoned with a probability p and a new nest with K eggs will be built, according to the similarities/differences of the eggs.
- Dimension of each solution is equal to the number of variables in the problem
- A rule of thumb recommends to have an initial population possibly more than $2 \times \text{number of variables} \times \text{number of objectives}$.

4.1 Multi-objective optimization

The solutions are initially represented as a vector of random numbers and using a sigmoid function is converted to binary value. Each solution is assessed by a fuzzy assessor [20] which helps in reducing the complexity and computational time. Out of 16, some rules created to define the trade-offs between the many objectives are stated:

- If operating time is *low* and coverage is *small* and communication is *short* and time between ops is *excess* the SoS architecture quality is *worst*.
- If operating time is *high* and coverage is *more* and communication is *excellent* and time between ops is *average* the SoS architecture quality is *finest*.
- If operating time is *moderate* and coverage is *medium* and communication is *mediocre* and time between ops is *average* the SoS architecture quality is *ordinary*.
- If operating time is *low* and coverage is *small* and communication is *short* and time between ops is *less* the SoS architecture quality is *finest*.

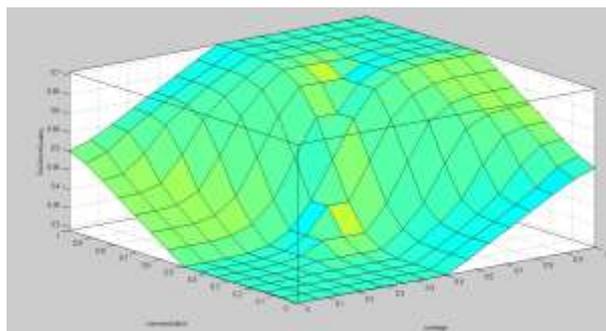


Fig. 2. A plot of the non-linear output surface of a given fuzzy inference system (fis) using the communication and coverage as inputs and the SoS Architecture quality as the output.

Table 1. A list of 6 solutions presented as binary vectors*

A	B	C	D	E	F	G	H	I
1 1 1 1 1	1 1	1 0	1 1 1 1	1	1 1 1 1 1	1 1 1 1 1	1 1 0 1 1 1 1	
0 1 1 1 1	1 1	1 0	0 1 1 1	0	1 1 1 1 0	1 1 1 1 1	1 1 1 1 1 1 1	
1 1 1 1 0	1 1	1 0	1 1 1 1	1	1 1 1 1 1	1 1 1 1 1	1 1 1 0 1 1 1	
1 1 1 1 1	1 1	1 0	1 1 0 1	1	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1 1 1	
1 1 1 1 0	1 1	1 1	1 1 1 1	1	1 1 1 1 1	1 1 1 1 1	1 1 0 1 1 1 1	
1 1 0 1 1	1 1	1 1	1 1 1 1	1	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1 1 1	

* "1" system present and "0" means system absent

The colors highlight the type of each system. The architecture quality of the solutions obtained differed from each other only in the second decimal. The highest quality obtained was 8.31.

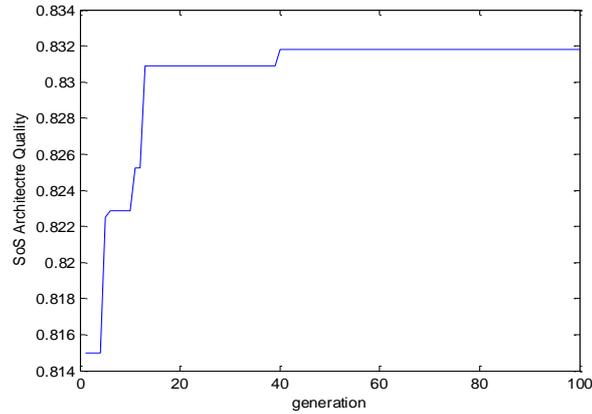


Fig 3 Graph of architecture quality changing with each generation of population

4.2 Executable Architectures

A more detailed assessment of the performance of the SoS to be built entails the consideration of the interactions between the participating systems in achieving the overall SoS capabilities. This include examining the work flow, the dependency between functions and capabilities, the information or resources flow between related activities, and parameters that define these interactions.

To facilitate such analysis, an executable architecture model is imperative. In this research, a modeling approach that combines the capabilities of OPM and CPN is proposed. Specifically, OPM is used to specify the formal system model as it can capture both the structure and behavior aspects of a system in a single model. CPN supplements OPM by providing simulation and behavior analysis capabilities. Consequently, a mapping between OPM and CPN is needed. Such mapping can follow the one developed in [25, 26]. OPM modeling supports both object-oriented and process-oriented paradigm. CPN supports state-transition-based execution semantics with discrete-event system simulation capability, which can be used to

conduct extensive behavior analyses and to derive many performance metrics. The incorporation of CPN also allows the developed system model to be doubled as an analysis model. A large collection of analysis methods and tools developed for CPN can be utilized for strong model analysis, verification, and validation. This modeling approach is explained here with ISR system development as an example.

Problem Definition. Suppose an ISR system to be developed is required to work under either of the following two scenarios in carrying out its designated mission: Scenario 1: Find concentration and movement of irregular troops that might be a warning of impending attack; and Scenario 2: Provide surveillance support to friend troops during an attack. The systems participating the SoS need to collaborate with each other to implement such scenarios, which can be described, in more details, by following work flow:

- 1) The C&C selects and dispatches required number of systems that carry EO, IR or SAR capabilities to the target terrain.
- 2) The deployed systems carrying EO, IR or SAR capabilities watch the designated terrain and collect the observations with the carried equipment.
- 3) The collected data are then transmitted over wide-band data links (provided by communication systems) to the exploitation centers in real time.
- 4) Upon receiving the data, the exploitation centers analyze the data and provide recommendations.
- 5) The recommendations are sent to the C&C through the communication systems.
- 6) C&C makes decisions based on the data received from the exploitation center. Such decisions may include adjusting the deployment of the EO, IR and SAR capabilities by sending command through communication system to those deployed systems.

Table 2 summarizes the related parameters of these systems the capabilities each of the system carries.

Systems	Shadow					Gray			apache		C&C		Exp.		Atl		Voice/Chat				LOS				BLOS	
	A					B			C		D		E		F		G				H				I	
ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
Operating time (hour)	9	9	9	9	9	25	25	2	2	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	
Time between Operations	5	5	5	5	5	24	24	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Coverage	85	85	85	85	85	150	150	200	200	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Capacity	-	-	-	-	-	-	-	-	-	10	10	10	10	40	20	20	20	20	15	15	15	15	10	10	10	
Capability																										
a EO	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
b IR	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
c Radar	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
d Command, Control	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
e Data Analysis	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0		
f Fusion	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0		
g Strike	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0		
h LOS	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1		
i BLOS	0	0	0	0	0	1	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0		

* "1" means possession of capability and "0" means absent of capability

System model: based on the problem definition presented in last section 4, an OPM model specifying the ISR system can be the one shown in Fig. 3a and Fig. 3b. Fig. 3a presents the overall system model whereas Fig. 3b presents an unfolding of an object, *ReconnaissanceSys*, to reveal its detailed characters. Similar unfolding for other objects in Fig. 3a is omitted in this paper. This OPM model is a class model, from which instance models can be created. The systems identified in Section 4.2.1 are grouped into five types: Reconnaissance, Communication, Exploitation center, Command & Control Systems, and Attack Systems. They are modeled as OPM objects (represented as rectangular boxes). Reconnaissance systems have two states, Operation and OffDuty, represented as smoothed rectangles encapsulated in the *ReconnaissanceSys* object. Two process (oval shaped) *GoOffDuty* and *GoOnDuty* can change a reconnaissance system from Operation state to OffDuty state or vice versa. An OPM object, *Data*, representing all types of data and information to be processed by the SoS, is created on Fig. 3a with 5 states (Raw, ReceivedByExpl, Recommendations, ReceivedByCC, and ActionPlan), each of which represents a stage of the information being processed during the operation of the SoS. Four processes (*Transmit1*, *Analyze*, *Transmit2*, and *DecisionMaking*) cause the change of states of a data object. The execution of each of these processes requires the support of one or more objects as indicated by instrument links of OPM (circle end connectors between objects and processes).

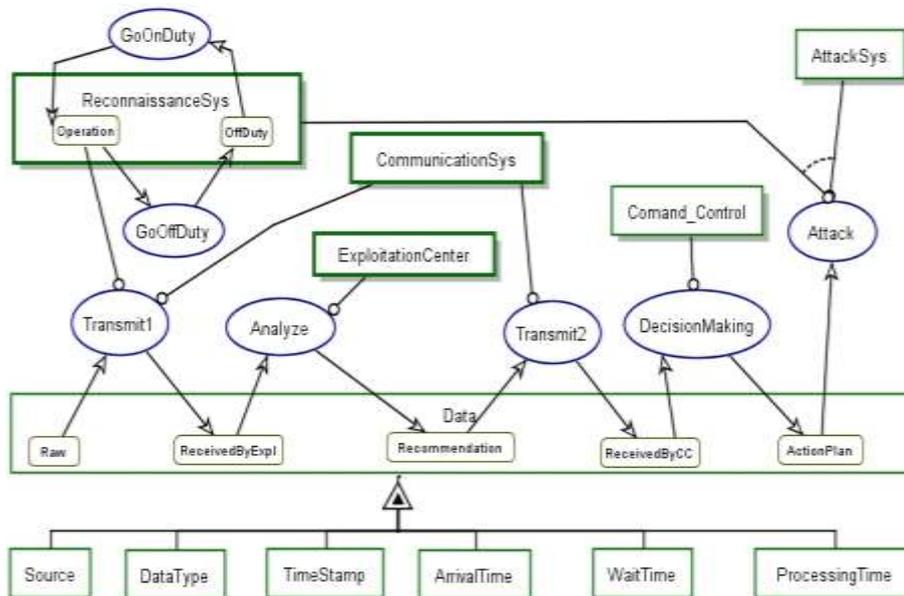


Fig. 4a Generative class model of the SoS represented using OPM

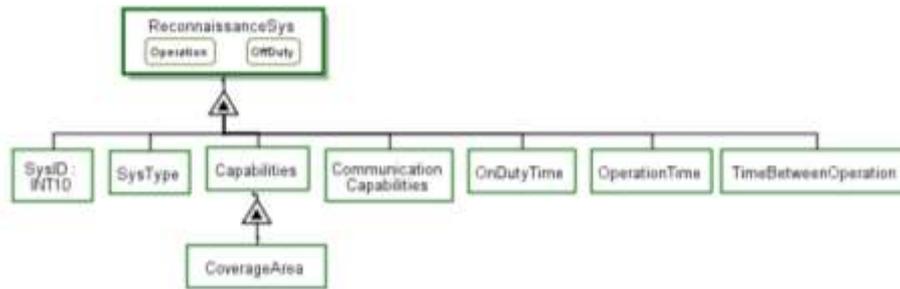


Fig. 4b Unfolding the Reconnaissance System

From such OPM model, A CPN model as the one shown in Fig. 4a and Fig. 4b can be developed by following the mapping rules [25, 26]. The basic idea that maps OPM to CPN is as follows: Map OPM processes to CPN transitions. Map OPM attribute objects (objects connected to their parent object using exhibition-characterization links) to CPN color sets. Such color set thus defines the set of class attributes for the OPM objects being connected by those attribute objects. Map non-attribute objects that have no states and object states of OPM to CPN places. Map the value(s) of an OPM object to CPN token(s). One or a set of tokens on a CPN place represents either the existence of an object or an object being at the state represented by that place. The former corresponds to the case that the place is mapped from an OPM object with no state and the token(s) on that place represents alternative objects. The latter corresponds to the case that the place is mapped from an OPM state. An object, as in object-oriented modeling, is defined by three parts, states, attributes and services (or methods, functions, or processes). OPM structural links that have no effect on the system dynamics are not mapped to CPN.

In the CPN model shown in Fig. 4a and Fig. 4b, tokens are used to represent the instance of a system in the SoS. Particularly, tokens at CPN place S_{dn} represent the available reconnaissance systems at off duty state, which is also the initial state of all such systems. Such tokens are defined by a list type of value (color set), which encode the attributes that define the corresponding objects. The first element in the list is the system ID (ranging from 1 to 25), the second element is system type (ranging from A to I), the third element is a list of capabilities carried (ranging from a to i), the fourth element is a list of communication capabilities (with value of i and j), and the fifth to seventh elements are the time attributes, i.e., on duty time (the simulation time at which a system goes on duty), operation time (mins), and break time (mins), respectively. Tokens at place $Comm$ represent the communication systems and they are defined with similar list values. The number of tokens with a particular system ID at that place represents the capacity of that communication system. Similarly tokens and their numbers at place Exp represent the exploitation centers and their capacities, respectively. The processing time required by Exploitation center and C&C is simulated by time delays added to the output arc of the $Process$ and $Decision$ transition, respectively. Such time delays are assumed to follow exponential distribution, the value of which is computed through a function (not shown in Fig. 4a or Fig. 4b). The CPN page $Ltran$ models is a substitution transition that represents

the data collection activity of the reconnaissance systems. The arrival of data items is assumed to follow an exponential distribution too.

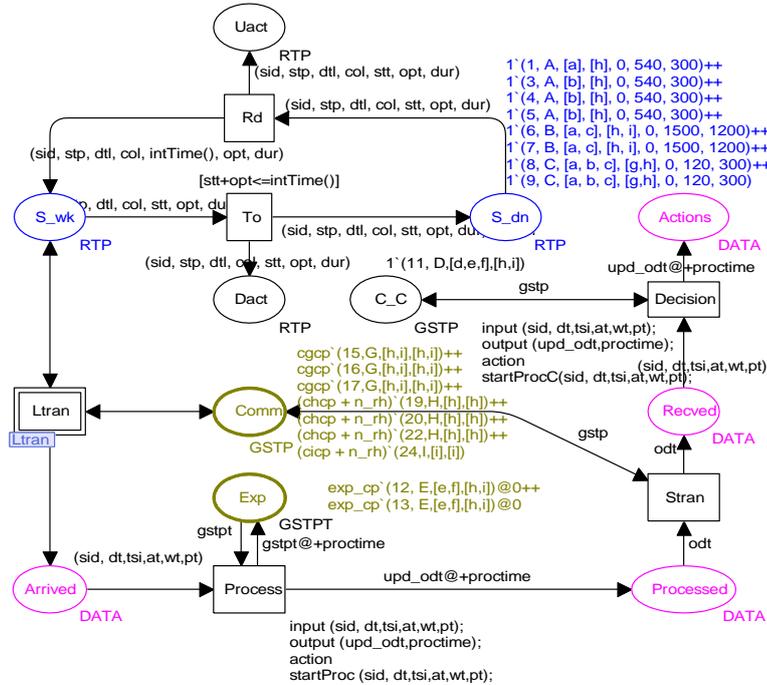


Fig. 5a CPN model converted from OPM

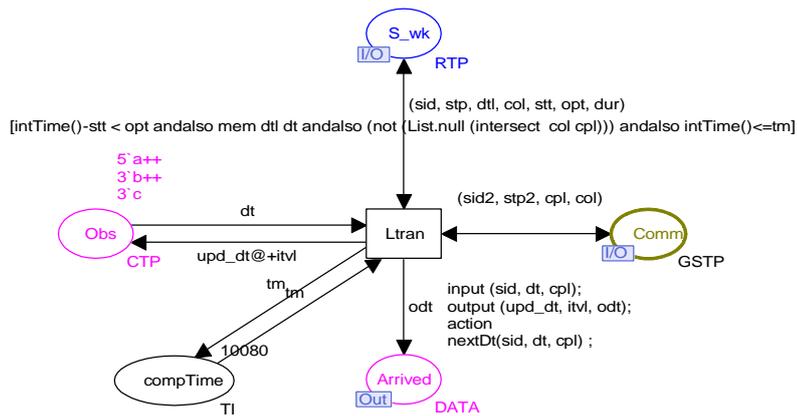


Fig. 5b CPN substitution transition representing data collection activity.

Performance assessment. The optimization objectives introduced in the beginning of this section can be evaluated through the information collected from running the CPN simulations. The total coverage area A^{SoS} of the SoS is calculated from summing up the coverage areas of the individual systems on duty. To get the information regarding what system (s) is on duty at a particular time, a CPN place, U_{act} , is added to the CPN model shown in Fig. 4a, as an output place of transition R_d . By collecting the tokens recorded at place U_{act} , it can be known what systems are on duty at a particular time and therefore, the total coverage area of the SoS as a function of time. Accordingly, the average coverage area and minimum coverage area during the entire operation time can also be obtained. The Operating time, time taken to recover between operations can be obtained directly from the on duty/ off duty recorded store at the tokens at place U_{act} . The communication capacity is the sum of individual communication system's communication capacities. In addition, the Data token has a list type of value, (sid, dt, tsi, at, wt, pt), where sid is the system type identifier, dt is the data type identifier (each data type binds to a particular capability, i.e., EO, IR, or SAR), tsi is the system simulation time, at is the initial arrival time of a data item, i.e., the time that the data is collected by a reconnaissance system, wt is the total waiting time, up to the current system time, that a data item spent, and pt is the total processing time, up to the current system time, that a data item spent. The difference between the current system simulation time tsi and the initial arrival time at is the latency (equals to wt + pt). Finally, the total simulation time is controlled by the token value at place compTime. It has an initial value of 10080, which represents the period of a week length.

5 Conclusion

The Acknowledged SoS systems can be used to model a broad variety of complex systems in the real world. The ANYLOGIC based integrated model developed in this research is designed to incorporate negotiation strategies and generate multiple responses of the counter offer game between the two parties [27]. The meta-architecture generated is negotiated for possible implementation by the acknowledged SoS. The Binary cuckoo search constrained optimization offers an efficient alternative for generating and selecting optimum meta-architectures. Any one of the architectures can be converted to executable models using the aforementioned techniques. The results thus obtained will help the SoS coordinator in selecting the architecture and negotiate effectively with the systems. The incorporation of OPM and CPN modeling allows the capturing of the interactions between participating systems and the behavior analysis based on such interactions. Accordingly, more detailed performance assessment can be carried out based on such models. Together, these technologies enrich the FILA with comprehensive decision support capabilities.

Acknowledgments. This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University

Affiliated Research Center managed by Stevens Institute of Technology. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

References

1. Dahmann, J., Rebovich, G., Lowry, R., Lane, J., & Baldwin, K. (2011, April). An implementers' view of systems engineering for systems of systems. In Systems Conference (SysCon), 2011 IEEE International (pp. 212-217). IEEE.
2. Cihan H. Dagli. (2014, Feb). Flexible and Intelligent Learning Architectures for SoS (FILA-SoS) SoS Analysis part 2. 5th Annual SERC Sponsor Research Review, Georgetown University, Washington, DC.
3. Siddhartha Agarwal, Lou E. Pape, Nil Kilicay-Ergin, Cihan H. Dagli, Multi-agent Based Architecture for Acknowledged System of Systems, Procedia Computer Science, Volume 28, 2014, Pages 1-10, ISSN 1877-0509.
4. Dagli, Cihan. (2013). An Advanced Computational Approach to System of Systems Analysis & Architecting Using Agent-Based Behavioral Model, SERC.
5. Wang, Renzhong, and Cihan H. Dagli. 2011. "Executable System Architecting Using Systems Modeling Language in Conjunction with Colored Petri Nets in a Model-Driven Systems Development Process." Systems Engineering 14 (4): 383–409.
6. Arnold, Alexandre, Boyer, Benoît and Legay, Axel. "Contracts and Behavioral Patterns for SoS: The EU IP DANSE approach." Paper presented at the meeting of the AiSoS, 2013.
7. R.Paulen, S. Engell: DYMASOS – Dynamic Management of Physically Coupled Systems of Systems, published on ERCIM News 97, April 2014, Special theme: Cyber-Physical Systems, February 25, 2014.
8. Joey W. Coleman, Anders Kaels Malmos, Peter Gorm Larsen, Jan Peleska, Ralph Hains, Zoe Andrews, Richard Payne, Simon Foster, Alvaro Miyazawa, Cristiano Bertolini, and André Didier. COMPASS Tool Vision for a System of Systems Collaborative Development Environment. In Proceedings of the 7th International Conference on System of System Engineering, IEEE SoSE 2012, IEEE, July 2012.
9. Siemieniuch, C., Sinclair, M., Lim, S. L., Henson, M. S., Jamshidi, M., & DeLaurentis, D. (2013). Project Title Trans-Atlantic Research and Education Agenda in Systems of Systems (T-AREA-SoS).
10. CPS20: CPS 20 years from now - visions and challenges, a CPSWeek workshop, in Berlin, Germany, April 14 2014, <http://www.cyphers.eu/>.
11. Leonardo Montecchi, Paolo Lollini and Andrea Bondavalli. "A DSL-Supported Workflow for the Automated Assembly of Large Performability Models". In Proceedings of the 10th

European Dependable Computing Conference (EDCC 2014). Newcastle upon Tyne, United Kingdom.

12. Sebastian Engell, Cyber-physical Systems of Systems –Definition and core research and development areas. Working Paper of the Support Action CPSoS, 2014.
http://www.cpsos.eu/wp-content/uploads/2014/05/CPSoS-Press-Release_Project-Launch.pdf
13. Local4Global: "SYSTEM-OF-SYSTEMS THAT ACT LOCALLY FOR OPTIMIZING GLOBALLY", Project Number: 61153, Project Start Date: 01/10/2013. <http://local4global-fp7.eu/>.
14. Hodges, C. 2014. COBWEB – Citizen Observatories Web: Ecology meets the crowd. In GIS and Remote Sensing: the End of Fieldwork? Remote sensing and its role in ecological assessment, Chartered Institute of Ecology and Environmental Management (CIEEM) Welsh Section Conference and AGM, Aberystwyth, 21st February 2014.
15. Uhler, P. F., Chen, R. S., Gabrynowicz, J. I., & Janssen, K. (2009). Toward Implementation of the Global Earth Observation System of Systems Data Sharing Principles. *J. Space L.*, 35, 201.
16. Laudy, C., Petersson, H., & Sandkuhl, K. (2010, July). Architecture of knowledge fusion within an Integrated Mobile Security Kit. In *Information Fusion (FUSION), 2010 13th Conference on* (pp. 1-8). IEEE.
17. <http://www.shared-e-fleet.com/> http://www.iao.fraunhofer.de/images/iao-news/iao-news_june_2013_en.pdf
18. Louis Pape, Siddhartha Agarwal, Kristin Giammarco, Cihan Dagli, Fuzzy Optimization of Acknowledged System of Systems Meta-architectures for Agent based Modeling of Development, *Procedia Computer Science*, Volume 28, 2014, Pages 404-411.
19. Renzhong Wang, Cihan H. Dagli: Computational System Architecture Development Using a Holistic Modeling Approach. *Complex Adaptive Systems 2012*: 13-20.
20. Agarwal, S., Pape, L., Ergin, N., Dagli, C. Multi-agent Based Architecture for Acknowledged System of Systems, *Procedia Computer Science*, Volume 28, 2014, Pages 1-10.
21. Acheson, P., Dagli, C., & Kilicay-Ergin, N. (2014). Fuzzy Decision Analysis in Negotiation between the System of Systems Agent and the System Agent in an Agent-Based Model. arXiv preprint arXiv:1402.0029.
22. Agarwal, S., Dagli, H.C, Adaptive Negotiating Strategies of Systems of Systems Coordinator using Deep Belief Networks, CESUN 2014. 4th International Engineering Systems Symposium.
23. Yang, X. S., & Deb, S. (2013). Multiobjective cuckoo search for design optimization. *Computers & Operations Research*, 40(6), 1616-1624.
24. Yang X. S. (2010). *Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley & Sons.

25. Wang, Renzhong. 2012. "Search-Based System Architecture Development Using a Holistic Modeling Approach." Ph.D dissertation, Missouri University of Science and technology.
26. Wang, R and C. H. Dagli, "Developing a holistic modeling approach for search-based system architecting," *Procedia Computer Science*, vol. 16, pp. 206–215, 2013.
27. Riddle, S., "Contract-based modelling and analysis technologies for Systems-of-Systems," *System of Systems Engineering (SoSE)*, 2012 7th International Conference on , vol., no., pp.469,470, 16-19 July 2012.