SYSTEMS ENGINEERING
Research Center

MISSOURI S&T
Smart Engineering Systems Lab
Engineering Management and Systems Engineering Department

Flexible and Intelligent Learning Architectures for SoS
FILA-SoS

By
Cihan H. Dagli
dagli@mst.edu
Principal Investigator DoD Systems Engineering Research Center-UARC
Founder and Director Systems Engineering Graduate Program
Missouri University of Science and Technology

NDIA 18th Annual Systems Engineering Conference October 26-29, 2015 Waterford . Springfield, VA

Project Team

- **Principal Investigator and Presenter:** Dr. Cihan Dagli, Missouri S&T
- Dr. Nil Ergin, Assistant Professor, Penn State University
- Dr. David Enke, Professor, Missouri S&T
- Dr. Abhijit Gosavi, Associate Professor, Missouri S&T
- Dr. Ruwen Qin, Assistant Professor, Missouri S&T
- Dr. Dincer Konur, Assistant Professor, Missouri S&T
- Dr. Renzhong Wang, Former Postdoctoral Fellow , Missouri S&T (Currently with Microsoft)
- Louis Pape, Siddhartha Agarwal, Former System Engineering PhD Students, Missouri S&T

Acknowledgement

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.

FILA-SoS Highlights

FILA-SoS and the Wave Process address four of the most challenging aspects of system-of-system architecting:

1.) Dealing with the uncertainty and variability of the capabilities and availability of potential component systems.

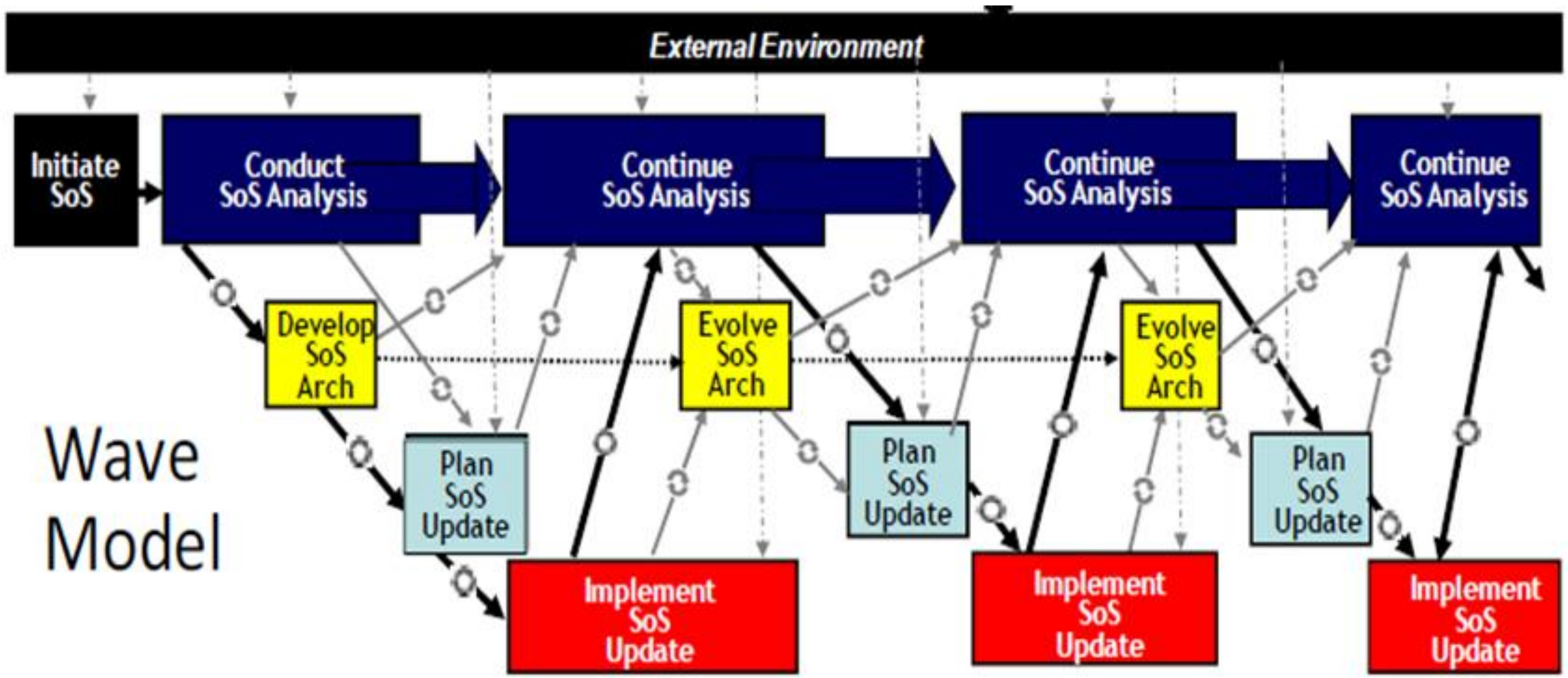
2.) Providing for the evolution of the system-of-system needs, resources and environment over time.

3.) Accounting for the differing approaches and motivations of the autonomous component system managers.

4.) Optimizing system-of-systems characteristics in an uncertain and dynamic environment with fixed budget and resources

FILA-SoS does so using straightforward system definitions methodology and an efficient analysis framework that supports the exploration and understanding of the key trade-offs and requirements by a wide range system-of-system stakeholders and decision makers in a short time.

The Wave Model of SoS



The Wave Model of SoS initiation, engineering, and evolution

Wave Process and FILA-SoS

Initialize SoS

- Wave process: Understand the SoS objectives and operational concept (CONOPS), gather information on core systems to support desired capabilities
- FILA-SoS: Enter Input values required to run the FILA-SoS which include the number of negotiation cycles, meta-architecture generation model selection type and individual system negotiation model types

Conduct SoS Analysis

- Wave process: Establish an initial SoS baseline architecture for SoS engineering based on SoS requirements space, performance measures, and relevant planning elements
- FILA-SoS: Execute the meta-architecture generation model which selects an initial SoS baseline architecture using the given input data

Develop/ Evolve SOS

- Wave process: Identify the necessary changes in contributing systems in terms of interfaces and functionality in order to implement the SoS architecture
- FILA-SoS: Send connectivity request to individual systems and start the negotiation between SoS and individual systems

Plan SoS Update /

- Wave process: Plan for the next SoS upgrade cycle based on the changes in external environment, SoS priorities, options and backlogs
- FILA-SoS: Determine which systems to include based on the negotiation outcomes and form a new SoS architecture

Implement SoS Architecture

- Wave process: Establish a new SoS baseline based on SoS level testing and system level implementation
- FILA-SoS: Evaluate the negotiated architecture quality and decide to renegotiate or move on to the next acquisition wave

SoS Behavior Object Process Model

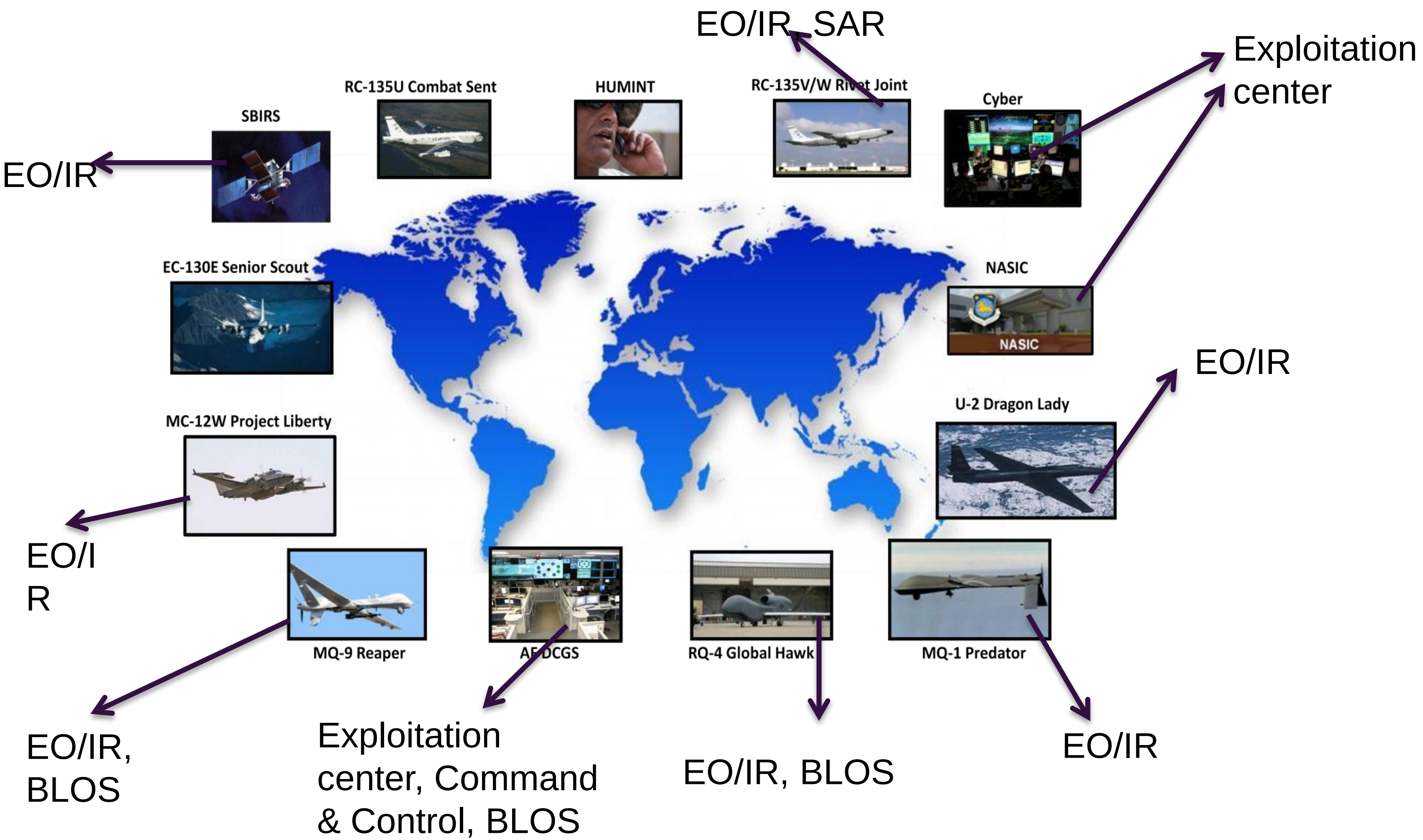
- Run SoS behavior model (Colored Petri Nets) for overall functionality and capability of the meta-architecture

FILA-SoS ISR Implementation

Initialize SoS

- Wave process: Understand the SoS objectives and operational concept (CONOPS), gather information on core systems to support desired capabilities
- FILA-SoS: Enter Input values required to run the FILA-SoS which include the number of negotiation cycles, meta-architecture generation model selection type and individual system negotiation model types
- The overall Capability: ISR & Targeting of Gulf War Scud TELs
- The sub capabilities (capabilities of contributing systems):
  - Electro-Optic/InfraRed (EO/IR) search capability
  - Side looking, synthetic aperture radar (SAR)
  - Command and control facilities
  - Exploitation centers (smaller ones in theater and a large one in CONUS)
  - Communication capabilities, both line of sight (LOS) limited to in-theater, and beyond line of sight (BLOS)

ISR Domain Model Detail

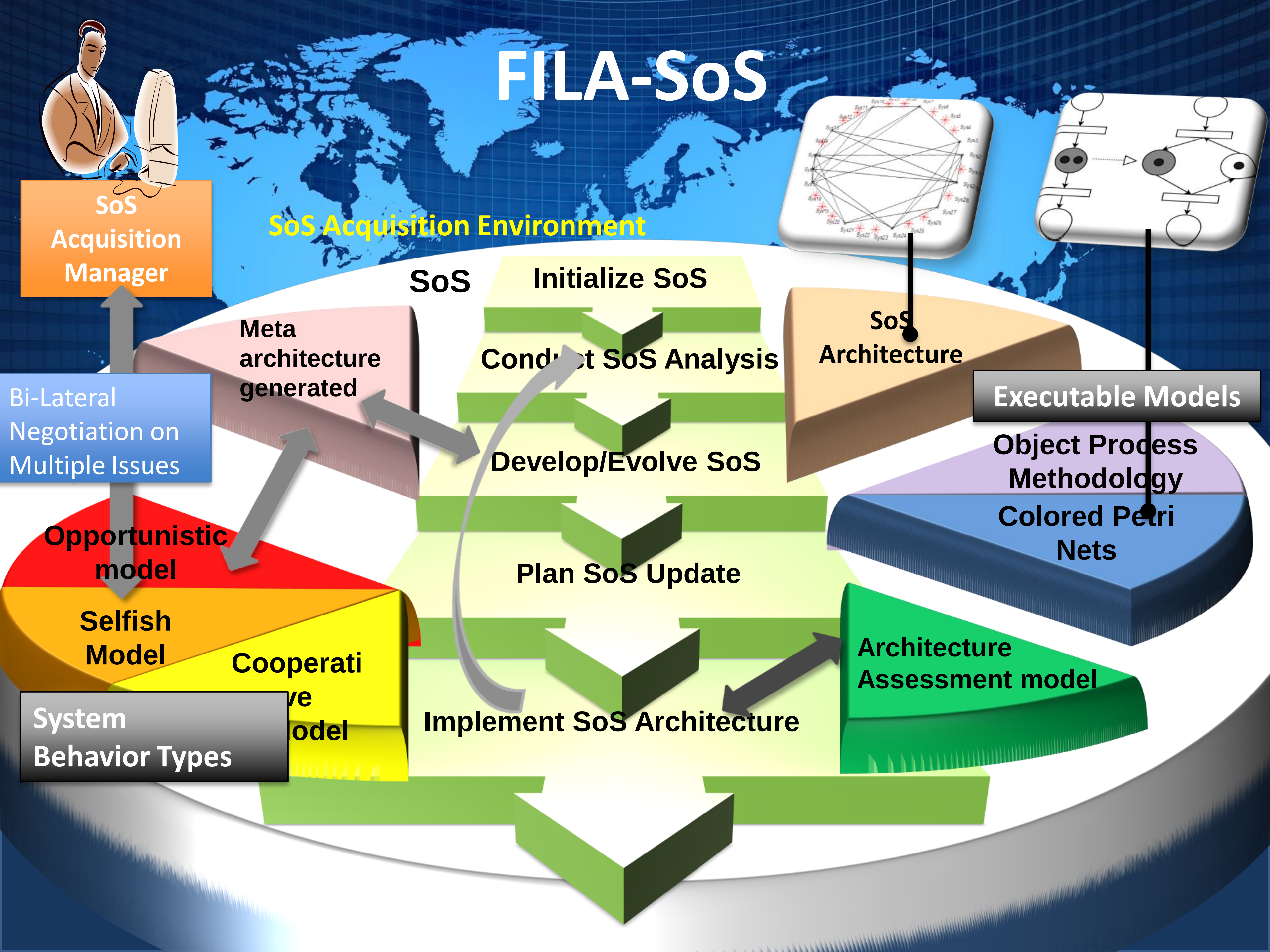


FILA-SoS ISR Implementation

__Conduct_SoS_Analysis:__ __Wave process:__ Establish an initial SoS baseline architecture for SoS engineering based on SoS requirements space, performance measures, and relevant planning elements. Execute the meta-architecture generation model which selects an initial SoS baseline architecture using the given input data. The key performance attributes are given below:

- **Performance**: Systems bring coverage sq/mi that are summed to provide SoS Performance
- **Affordability**: There are costs to develop interoperability, and costs to operate the systems. These both are summed to create the inverse of Affordability
- **Robustness**: We define the robustness as being greater if a smaller loss in performance results for the loss of any single cooperating system and its interfaces from the SoS architecture instance.
- **Flexibility:** Attribute describes the number of choices available to the SoS manager. The evaluation of Flexibility consists simply of counting the capabilities that have less than two sources.

System	Type Sub-System	Cap ability Number	Coverage sq mi/hr;	Develop $M/ epoch/ interface	Operate $K/hr per system	Time to Develop, Epochs	Number possible in SoS	System Number
Fighter	EO/IR	1	500	0.2	10	1	3	1-3
RPA	EO/IR	1	2000	2	2	1	4	4-7
U-2	EO/IR	1	50000	0	15	0	1	8
DSP	IR	1	100000*.01	1	1	1	1	9
Fighter	Radar	2	3000	0.7	10	1	3	10-12
JSTARS	Radar	2	10000	0.1	18	1	1	13
Theatre	Exploit	4	5000	2	10	1	2	14-15
CONUS	Exploit	4	25000	0.2	0	0	1	16
Control Station/ AOC	Cmd & Control	5	1	1	2	1	2	17-18
LOS Link	Comm	3	1	0.2	0	1	2	19-20
BLOS Link	Comm	3	1	0.5	3	1	2	21-22


FILA-SoS
SoS Acquisition Manager
SoS Acquisition Environment
SoS
Initialize SoS
Conduct SoS Analysis
Develop/Evolve SoS
Plan SoS Update
Implement SoS Architecture
Meta architecture generated
Bi-Lateral Negotiation on Multiple Issues
Opportunistic model
Selfish Model
Cooperati
ve
odel
System Behavior Types
SoS Architecture
Executable Models
Object Process Methodology
Colored Petri Nets
Architecture Assessment model

Independent Modules In FILA-SoS

1. Meta-Architecture Generation Fuzzy Genetic model—MATLAB code
2. Meta-Architecture Generation Multi-Level model—MATLAB code
3. Architecture Assessment--- MATLAB code
4. SoS Negotiation Model- JAVA code
5. System Negotiation Model: Selfish – MATLAB code
6. System Negotiation Model: Cooperative –MATLAB code
7. System Negotiation Model: Opportunistic-MATLAB code
8. Executable Model--- OPM & CPN
9. Overall Negotiation Framework – JAVA code

FILA-SoS: Flexible Intelligent Learning Architectures for SoS

FILA-SoS Capabilities

- Integrated model for modeling and simulating SoS systems with evolution for multiple waves.
- Models can be run independently and in conjunction with each other
- Two model types represent SoS behavior and various individual system behavior
- Study of negotiation dynamics between SoS and individual systems

FILA-SoS "What-if" Analysis; Model Modularity

- Variables such as SoS funding and capability priority can be changed as the acquisition progresses though wave cycles
- Simulation of any architecture through colored petri nets.
- Simulate rules of engagement & behavior settings: all systems are selfish, all systems are opportunistic, all systems are cooperative or a combination

FILA-SoS Value

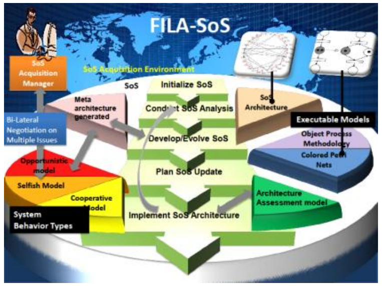
- Aiding the SoS manager in future decision making
- Understand emergent behavior of systems in the acquisition environment and impact on SoS architecture quality
- Study the dynamic behavior of different type of systems (selfish, opportunistic, cooperative)
- Identify intra and interdependencies among SoS elements and the acquisition environment

Potential Application

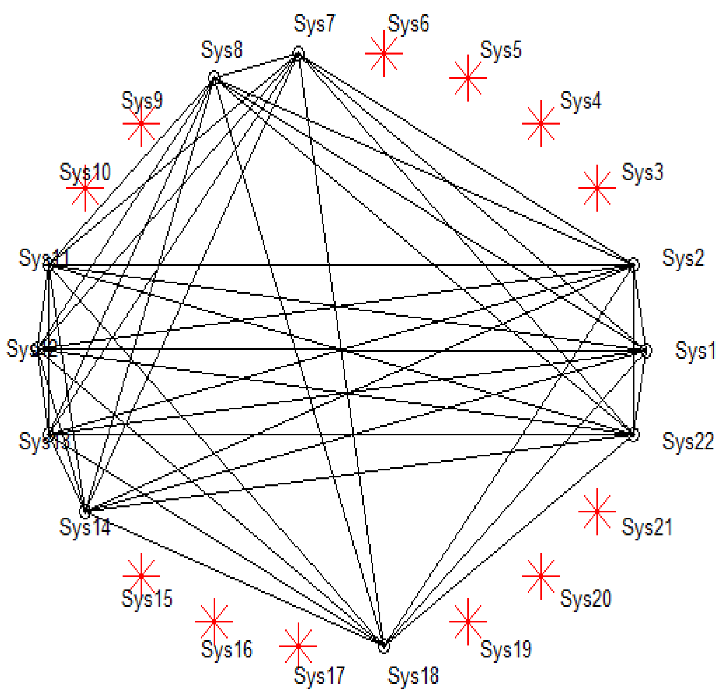
- Can be use to model of wide variety of complex systems models such as logistics, cyber-physical systems etc.
- Test-bed for decision makers to evaluate operational guidelines and principles for managing various acquisition environment scenarios
- Applicable to SoS that evolve as it has the multiple wave simulation capability

Possible Future Capabilities

- Extending the model to include multiple interface alternatives among systems
- Incorporation of risk models into environmental scenarios



Searching for SoS Meta-Architecture
(Multi-level Optimization)



Architecture I

Systems Selected
Fighters (EO/IR) (1-2)
RPA (7)
U-2 (8)
Fighters (Radars-11,12)
JSTARS (13)
THEATRE (14)
Control Station/ AOC (18)
BLOS Link(19,20)

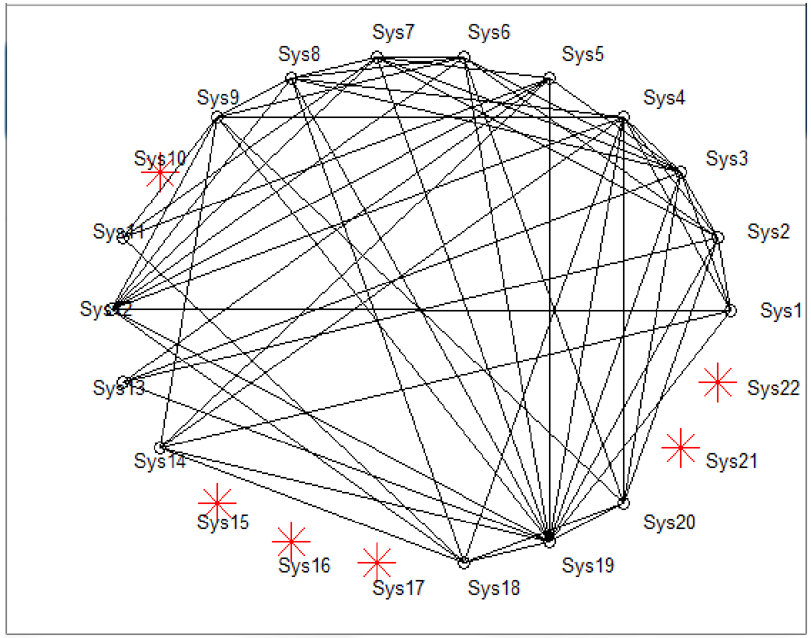
Interfaces Selected	
Fighters (EO/IR)	Fighters (Radar)
Fighters (EO/IR)	U-2
Fighters (EO/IR)	JSTARS
U-2	JSTARS
THEATRE	U-2
Control Station/ AOC	BLOS Link
…	…

Fighters (EO/IR)	Fighters (EO/IR)	…	…	BLOS	$I_{\text{Fighters with JSTARS}}$	$I_{\text{Fighters with U-2}}$	$I_{\text{JSTARS with U-2}}$
Systems					Interfaces		

X_1	X_2	X_i	…	X_m	$X_{1 \text{ with } 2}$	$X_{1 \text{ with } 3}$	$X_{1 \text{ with } m}$	$X_{2 \text{ with } 3}$	…	$X_{i \text{ with } j}$	…	$X_{(m-1) \text{ with } m}$
Systems					Interfaces							

Searching for SoS Meta-Architecture

(Fuzzy genetic Optimization)



Architecture II

Systems Selected
Fighters (EO/IR) (1,2,3)
RPA (4,5,6,7)
U-2 (8)
DSP(9)
Fighters (Radars-11,12)
JSTARS (13)
THEATRE (14)
Control Station/ AOC (18)
BLOS Link(22)

Interfaces Selected	
Fighters (EO/IR)	Fighters (Radar)
Fighters (EO/IR)	U-2
Fighters (EO/IR)	JSTARS
U-2	JSTARS
THEATRE	U-2
Control Station/ AOC	BLOS Link
…	…

Fighters (EO/IR)	Fighters (EO/IR)	…	…	BLOS	$I_{\text{Fighters with JSTARS}}$	$I_{\text{Fighters with U-2}}$	$I_{\text{JSTARS with U-2}}$
Systems					Interfaces		

X_1	X_2	X_i	…	X_m	$X_{1\text{ with }2}$	$X_{1\text{ with }3}$	$X_{1\text{ with }m}$	$X_{2\text{ with }3}$	…	$X_{i\text{ with }j}$	…	$X_{(m-1)\text{ with }m}$
Systems					Interfaces							

SYSTEMS ENGINEERING Research Center

SoS Architecture Assessment Model

MISSOURI S&T Smart Engineering Systems Lab
Engineering Management and Systems Engineering Department

Model Capabilities

- Capture non-linearity in key performance attribute tradeoffs
- Can accommodate any number of attributes for a selected SoS capability
- Capture multiple stakeholder's understanding of key performance attributes
- Algorithms to determine the value of the attributes
- Evaluate the quality of a given architecture based on the value of the attributes

What-if Analysis; Model Modularity

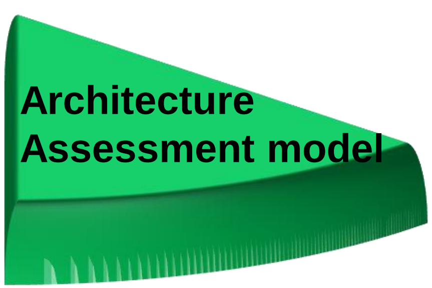
- Attribute definitions and algorithms are easily changed based on domain
- Model can be adjusted for different domains and stakeholder's
- New attributes can be added and old ones discarded
- Relative priorities of the attributes can also be accommodated by prioritizing assessment rules

Model Value

- By exploring the architecture 'space' with the "What-if" analysis, the stakeholders can develop a better understanding of how component systems can fit and work together
- Provide more realistic assessment than utility functions

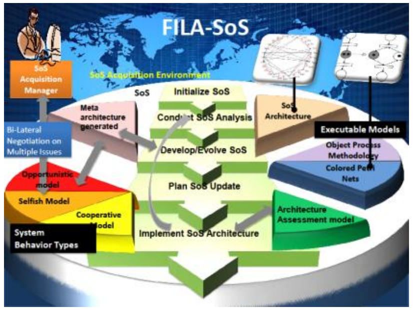
Potential Application

- Discovering, exploring, and adjusting stakeholder's firmly held (and sometimes mistaken) beliefs
- Finding new ways for systems to work together
- Finding more cost effective SoS arrangements
- Aid in negotiations with component systems to build an SoS



Possible Future Capabilities

- Improved visualization of the impact of many variables in SoS architecture and design
- Automated adjustment of model parameters



SYSTEMS ENGINEERING Research Center

Fuzzy-Genetic Optimization Model

MISSOURI S&T Smart Engineering Systems Lab
Engineering Management and Systems Engineering Department

Model Capabilities

- Evolutionary multi-objective optimization model for SoS architecting with many key performance attributes (KPA)
- Involves dynamic assessment of domain inputs
- Returns the best architecture consisting of systems and their interfaces

What-if Analysis; Model Modularity

- What happens if selected attributes such as performance, cost, and deadline of the systems change
- Analyze the range of different KPA's over the set of architectures
- What happens if number of systems having net centric capability reduces

Model Value

- Adds to the existing meta-architecture generation techniques
- Takes into account the net-centricity of the architecture
- Returns a set of SoS to initiate negotiation
- Fuzzy Assessor for several competing objectives

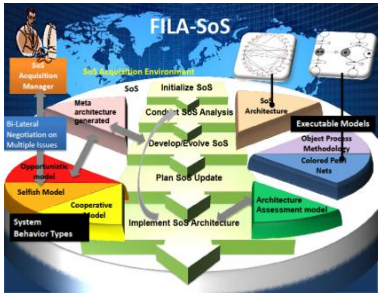
Potential Application

- Capable of finding architectures for multiple waves
- Model can be applied in any SoS domain such as logistics, network-centric systems, cyber-physical systems and supply chain

Meta architecture generated

Possible Future Capabilities

- Add multiple interfaces among a set of systems e.g. energy flow, information, and mechanical
- Estimation of Component and Interface Complexity
- Modify the model for a specific potential application



Multi-Level Optimization Model

Model Capabilities

- Generic mathematical model for SoS architecting with multiple attributes such as:
  - Min cost, Max performance, Min deadline
- Efficient evolutionary algorithm for solutions
- Returns a set of SoS alternatives

What-if Analysis; Model Modularity

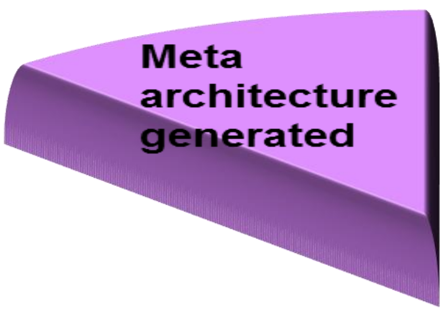
- What happens if selected attributes such as performance, cost, and deadline of the systems change
- What happens if some of the systems are not available or they cannot provide some of the capabilities they could provide
- What happens if systems can provide additional capabilities

Model Value

- Models practical settings of SoS architecting
  - Fund allocation for improvement
- Considers different objectives for SoS architecting
- Returns a set of SoS to initiate negotiation

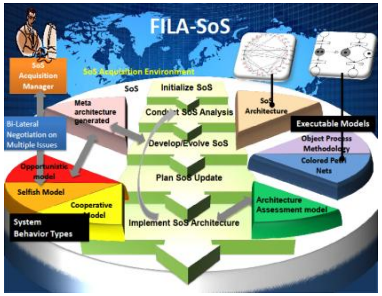
Potential Application

- Initiating the negotiation process
- Model can be applied in any SoS domain such as logistics, network-centric systems, cyber-physical systems and supply chain



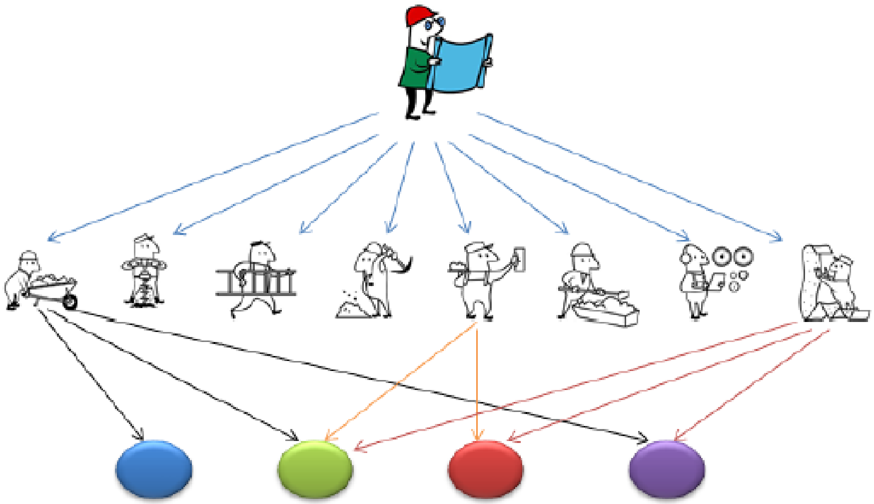
Possible Future Capabilities

- Modeling negotiation within SoS architecting
- Modeling competition among the systems
- Modeling flexibility of the systems and how to incentivize systems to become flexible
- Modify the model for a specific potential application



A System of Systems with Individual System Contracts

A Stackelberg Game



A Stackelberg Game:

- Leader: SoS Architect
  - Select the systems and allocate funds to the selected systems
- Followers: Selected systems
  - Improve the performance of the capabilities

FILA-SoS ISR Implementation

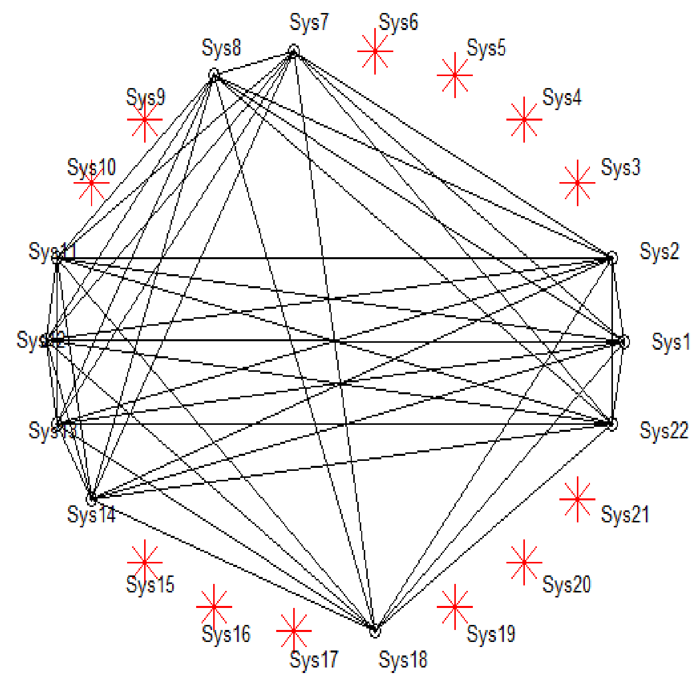
•Develop/ Evolve SOS

- Wave process: Identify the necessary changes in contributing systems in terms of interfaces and functionality in order to implement the SoS architecture
- FILA-SoS: Send connectivity request to individual systems and start the negotiation between SoS and individual systems

FILA-SoS ISR Implementation

Select Different System Behaviors

Systems Selected	Behavior
Fighters (EO/IR) (1-2)	Opportunistic, Opportunistic
RPA (7)	Opportunistic
U-2 (8)	Cooperative
Fighters (Radars-11,12)	Cooperative, Selfish
JSTARS (13)	Opportunistic
THEATRE (14)	Cooperative
Control Station/ AOC (18)	Selfish
BLOS Link(22)	Selfish



Architecture I

- ✓ **Assessment for the meta-architecture =3.69**
- ✓ **Key Performance Parameters values**
  - Performance=2.65 (More Acceptable)
  - Affordability=3.72
  - Flexibility=3
  - Robustness=3.76

SoS Negotiation Models

Model Capabilities

- Game theoretic negotiation model that will maximize the welfare for parties involved in the negotiation
- SoS utility function that takes into account local objectives for the individual systems as well as global SoS objective
- Incentive contract design to persuade uncooperative systems to join the SoS development

What-if Analysis; Model Modularity

- Analysis of incentive mechanisms under different behavioral settings including
  - When selfish behavior dominates the acquisition environment
  - When opportunistic behavior dominates
  - When cooperative behavior dominates
- Analysis of incentive mechanisms when there is uncertainty in individual system performance outcomes

Model Value

- Analysis of how incentives can be used to improve lack of collaboration in SoS acquisition which is a leading problem in SoS acquisition effectiveness.
- Analysis of how incentives can be used to ensure effective SoS mission performance

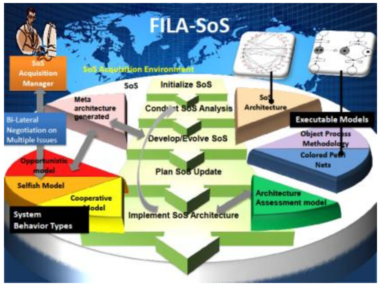
Potential Application

- Tool for evaluating operational guidelines and principles for incentive contract design for SoS acquisition under various acquisition environment scenarios

Bi-Lateral Negotiation on Multiple Issues

Possible Future Capabilities

- Study of risk taking preferences of individual systems and SoS manager and its impact on incentive contract design
- Incentive contract design for individual system groups that interact with each other



SYSTEMS ENGINEERING Research Center

Individual System Behavior Models: Non-Cooperative model (Selfish behavior)

Missouri S&T Smart Engineering Systems Lab
Engineering Management and Systems Engineering Department

Model Capabilities

- A negotiation protocol that clearly defines how negotiations are initiated, continued, and terminated.
- A decision framework of contract negotiation for individual systems.
  - A model of individual system's participation capability and negotiation behavior.
  - A generator of negotiation alternatives in the presence of multiple conflicts.
  - Three optimization models that help search alternatives with a minimum impact of conflicts.
  - An conflicts evaluation model that estimates negotiation outcomes for each alternative.

What-if Analysis; Model Modularity

- What if an individual system is more/less capable than the SoS expects?
- What if an individual system is more/less cooperative than the SoS expects?
- What if an individual system is a strategic negotiator?
- Whether an individual system can be impacted by negotiation strategies of the SoS, such as monetary incentive, time pressure, and others; and how?

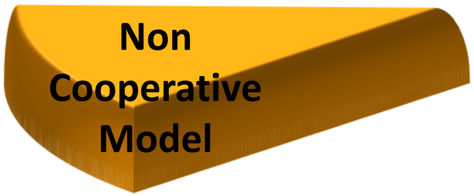
Model Value

- A negotiation model for individual systems in the setting of SoS acquisition, which can be used by the SoS manager to assess and train the SoS acquisition abilities/strategies.
- Realistic, challenging responses to the SoS manager's request for participation, which the SoS manager can use for developing an understanding of individual systems that have self-interests and are strategic negotiators, and also developing strategies for handling them.

Potential Application

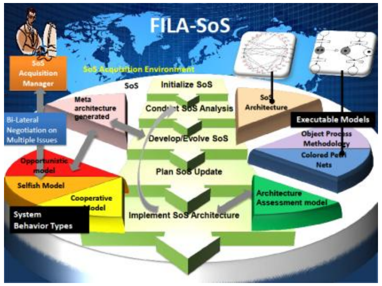
The negotiation model can be used by

- When changes in a system is difficult to make
- Individual suppliers/service providers in the negotiation of supply-procurement contracts.
- Individual persons in the development of dynamically reconfigurable teams.



Possible Future Capabilities

- An intelligent algorithm that can determine an optimal alternative in a fast manner.
- A learning mechanism with which the individual systems model can effectively calibrate the guess of SoS's utility functions.
- A broader band of negotiation strategies that can handle a wider range of negotiation scenarios.



Individual System Behavior Models: Semi-Cooperative (opportunistic behavior)

Model Capabilities

- Capability of being flexible or opportunistic: i.e., selfish or unselfish
- Markov-chain based model designed for handling uncertainty in negotiation modeling in an SoS

What-if Analysis; Model Modularity

- Testing scenarios for given performance criteria and given number of interacting systems in an SoS
- Ability to determine budget and schedule for any given negotiation model for an SoS

Model Value

- Useful for testing opportunistic behavior prevalent in industry partners exhibiting risk-prone behavior
- Ability to model very selfish to very selfless behavior on a continuum using a numerical user-adjustable scale

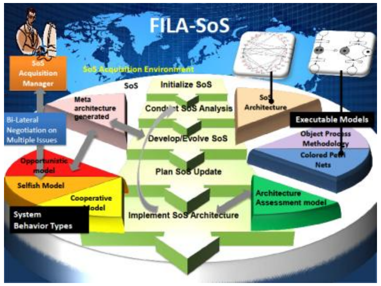
Potential Application

- Modeling behavior of defense firms competing to obtain contract
- Modeling of project durations for any system within the SoS
- Changes in a system are reasonable to make

Semi-Cooperative Model

Possible Future Capabilities

- Testing for risk-prone behavior of systems within an SoS
- Implementation of Machine Learning models for SoS controller



SYSTEMS ENGINEERING Research Center

Individual System Behavior Models: Cooperative

MISSOURI S&T Smart Engineering Systems Lab
Engineering Management and Systems Engineering Department

Model Capabilities

- The negotiation protocol is computationally scalable to a large number of issues and system types
- Presents a semi-cooperative behavior
- Can negotiate multiple issues simultaneously
- Illustrates the cognitive and financial aspects of human negotiations
- Bilateral negotiation mechanism

What-if Analysis; Model Modularity

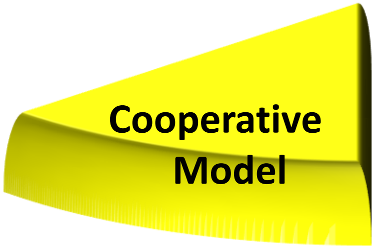
- The ability to work with other negotiation models
- Can work as an independent module
- The preferences and the strategy considerations of the systems are private, i.e., they are not known to the other systems or manager
- Can perform simulations for various scenarios
- Knowledge discovery and agent learning tools

Model Value

- Provide solutions in complex automated negotiation scenarios
- Model predictions can be used about similar situations which are previously not modeled
- Identify counterintuitive results or causal relationships

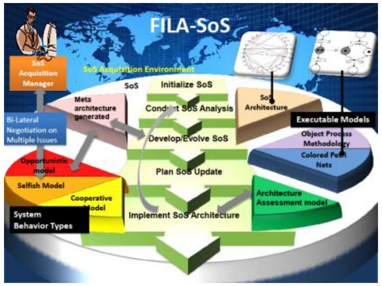
Potential Application

- Includes logistics, supply chain, cyber-physical systems, e-commerce, decision-making support etc.
- Changes in a system are easy to make



Possible Future Capabilities

- Solve the problem using a multi-criteria group decision making problem to handle multiple offers from the SoS manager
- Employ computing with words for decision making



Execute the negotiated Architecture through Object Process Methodology and Colored Petri Nets

SYSTEMS ENGINEERING Research Center

Missouri S&T Smart Engineering Systems Lab
Engineering Management and Systems Engineering Department

Model Capabilities

- Model the interactions between components of a system or subsystems in SOS
- Capture the dynamic aspect of the SoS and Simulate the behavior of the SoS
- Access various behavior related performance of the SoS
- Access different constitutions or configurations of the SoS

What-if Analysis; Model Modularity

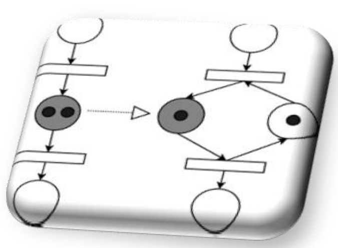
- Can be used in accessing the impact of changes in system parameters, constitution, and configuration to the overall functionality and capability of the SoS
- Can assess the system performance under various operational scenarios
- Good support of hierarchical modeling and can be used independently

Model Value

- Examine whether and how well the constituent systems can collaborate with each other in delivering the desired capabilities when the SoS is in operation
- Detailed, quantitative performance analysis

Potential Application

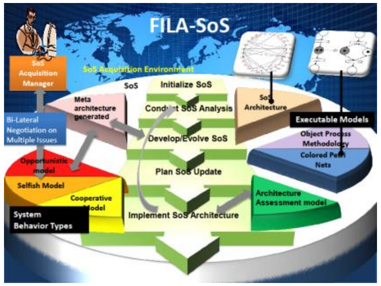
- Useful in situation where interactions between constituent systems, or system components are critical to fulfillment of the overall functionality and capability of the SoS
- Access the emergent behavior of the SoS



Executable
Colored Petri Nets

Possible Future Capabilities

- Automate the model construction, alternative generation and performance analysis process.
- Examine all possible operational states of the SoS



SECOND WAVE

- **Plan SoS Update /**

- Wave process: Plan for the next SoS upgrade cycle based on the changes in external environment, SoS priorities, options and backlogs
- FILA-SoS: Determine which systems to include based on the negotiation outcomes and form a new SoS architecture

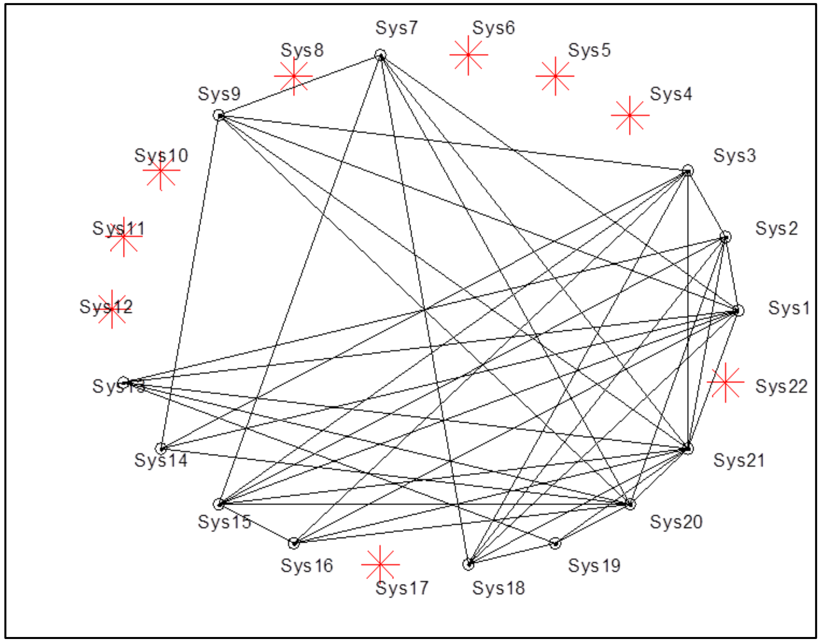
Evolution of SoS Architecture Through Multiple Waves

Sample Scenario for ISR-Wave 1 results

System	Type Sub-System	Cap ability Number	Coverage sq mi/hr;	Develop $M/ epoch/ interface	Operate $K/hr per system	Time to Develop, Epochs	System Number
Fighter	EO/IR	1	500	0.2	10	1	1
Trainer	EO/IR	1	2000	2	2	1	2-3
UAV	EO/IR	1	50000	0	15	0	4-8
DSP	IR	1	8000	0.1	1	1	9
Fighter	Radar	2	3000	0.7	10	1	10-12
JSTARS	Radar	2	10000	0.1	18	1	13
Theatre	Exploit	3	5000	2	10	1	14-15
CONUS	Exploit	3	25000	0.2	0	0	16
Control Station/ AOC	Cmd & Control	4	10000	1	2	1	17-18
LOS Link	Comm	5	10000	0.2	0	1	19-20
BLOS Link	Comm	5	5000	0.5	3	1	21-22

Evolution of SoS Architecture

- Meta-Architecture Wave 1



Assessment for the meta-architecture =3.47
Key Attribute values:

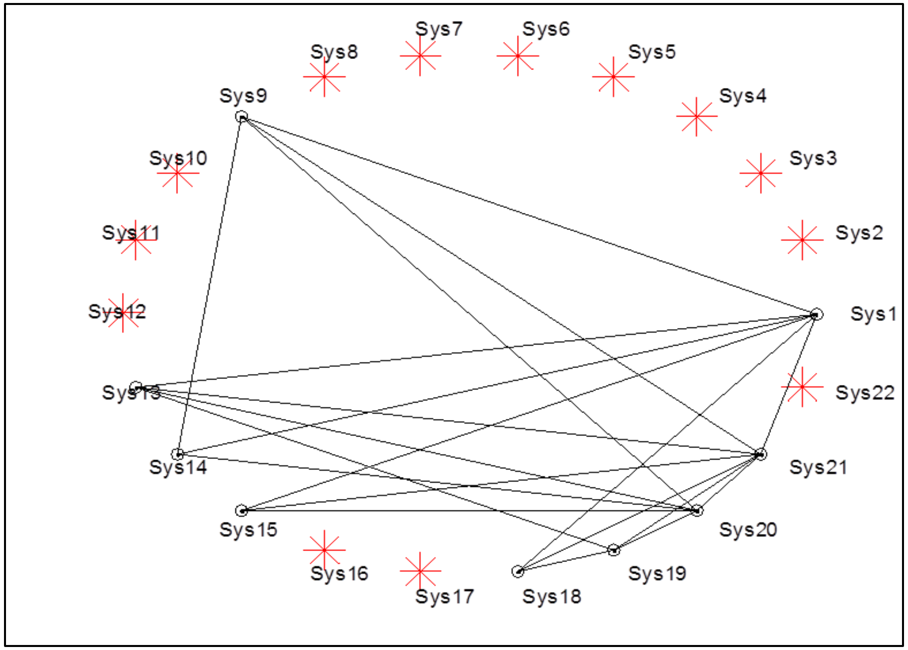
- Performance=2.16
- Flexibility=4
- Affordability=3.5
- Robustness=3.91

Systems selected in the Meta-Architecture

Systems Selected
Fighters (EO/IR) (1-3)
RPA (7)
U-2 (9)
JSTARS (13)
THEATRE (14,15)
CONUS (16)
Control Station/ AOC (18)
LOS link (19,20)
BLOS Link(21)

Evolution of SoS Architecture

- Negotiated Architecture Wave 1



Assessment for the meta-architecture =2.5
Key Attribute values:

- Performance=1.5
- Flexibility=2
- Affordability=3.72
- Robustness=2

Systems selected after negotiation S_i

Sys no	Behavior
9	selfish
13,14,15	cooperative
18	opportunistic
19	selfish
21	selfish
22	cooperative

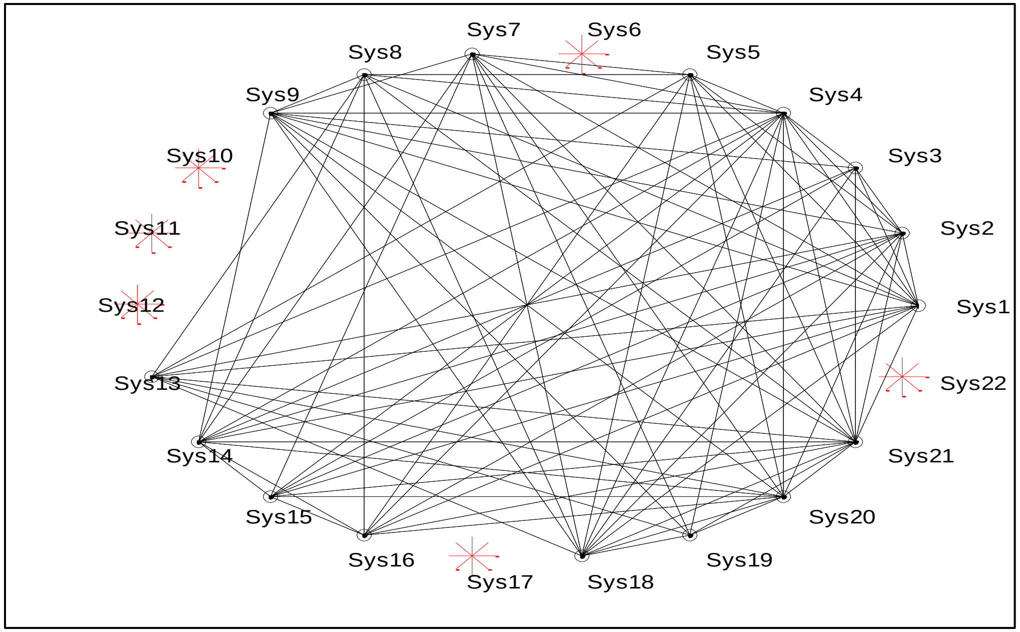
Evolution of SoS through to the Wave 2

- Input Domain File

System	Type Sub-System	Cap ability Number	Coverage sq mi/hr;	Develop $M/ epoch/ interface	Operate n///hr$K/hr per system	Time to Develop, Epochs	System Number
Fighter	EO/IR	1	500	0.2	10	1	1
Trainer	EO/IR	1	12000	0.1	8	1	2-3
UAV	EO/IR	1	8000	0.5	2.5	1	4-8
DSP	IR	1	8000	0.1	1	1	9
Blimp	Radar	2	20000	0.5	12	1	10-12
JSTARS	Radar	2	10000	0.1	18	1	13
Theatre	Exploit	3	5000	2	10	1	14-15
MOBExp	Exploit	3	15000	0.1	0.2	0	16
MOBC2	Exploit	4	12000	1	2	0	17
Control	Cmd &	4	10000	1	2	1	18
LOS Link	Comm	5	10000	0.2	0	1	19-20
BLOS Link	Comm	5	5000	0.5	3	0	21
Mil-Sat	Comm	5	15000	1	5	1	22

Evolution of SoS Architecture

- Meta-Architecture Wave 2



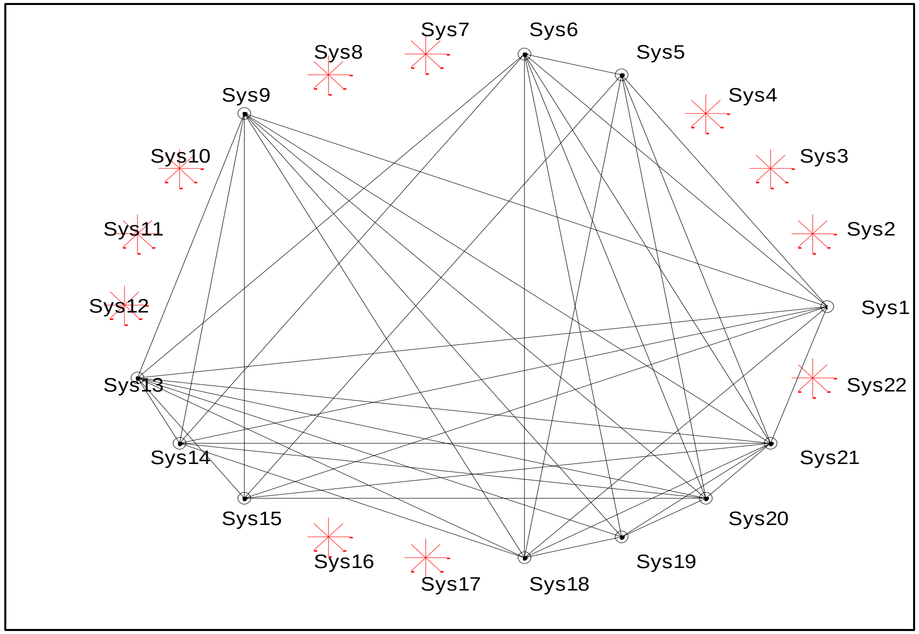
Assessment for the meta-architecture =3.61
Key Attribute values:

- Performance=2.28
- Flexibility=4
- Affordability=3.09
- Robustness=3.77

Systems Selected
Fighters (EO/IR) (1)
Trainer (2-3)
UAV (4,5,7,8)
DSP(9)
JSTARS (13)
THEATRE (14,15)
MOBExp(16)
Control Station/ AOC (18)
LOS link (19,20)
BLOS Link(21)

Evolution of SoS Architecture

- Negotiated Architecture Wave 2



Assessment for the meta-architecture =2.9
Key Attribute values:

- Performance=1.94
- Flexibility=2.7
- Affordability=3.6
- Robustness=3

Systems selected after negotiation S_i

Sys no	Behavior
1	selfish
5,6	opportunistic
9	selfish
13,14,15	cooperative
18	opportunistic
19,20,21	selfish

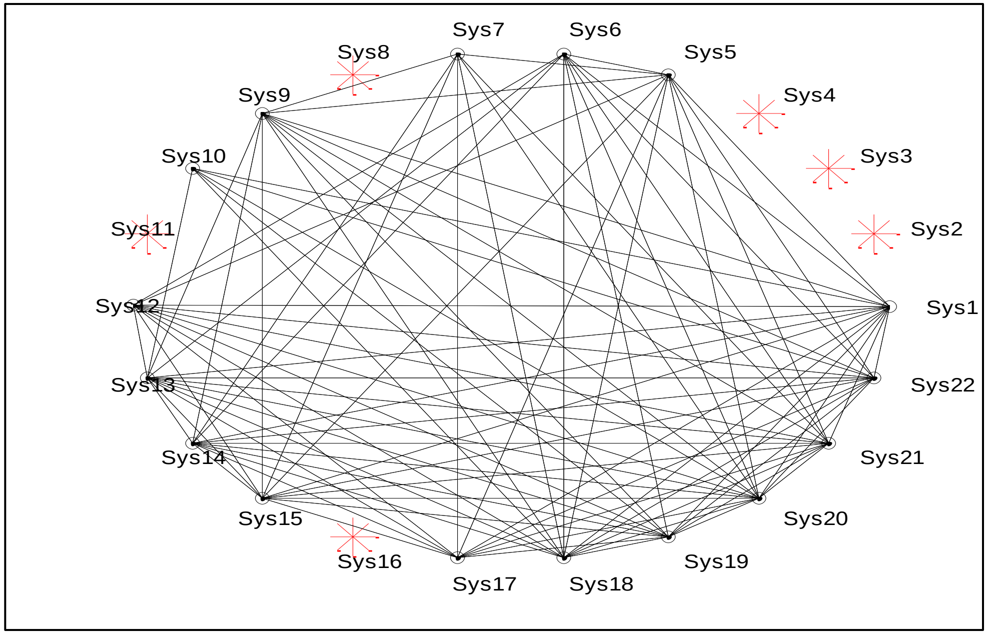
Evolution of SoS through to the Wave 3

Input Domain File

System	Type Sub-System	Cap ability Number	Coverage sq mi/hr;	Develop $M/ epoch/ interface	Operate n///hr$K/hr per system	Time to Develop, Epochs	System Number
Fighter	EO/IR	1	500	0.2	10	1	1
Trainer	EO/IR	1	4000	0.5	15	1	2-3
UAV -A	EO/IR	1	3000	1	12		
UAV-B	EO/IR	1	8000	0.5	2.5	1	5-6
UAV-C	EO/IR	1	5000	0	10	1	7-8
DSP	IR	1	8000	0.1	1	1	9
Blimp	Radar	2	30000	1.9	5	1	10-12
JSTARS	Radar	2	10000	0.1	12	1	13
Theatre	Exploit	3	5000	2	10	1	14-15
MOBExp	Exploit	3	4000	0.15	1	0	16
MOBC2	Exploit	4	4000	0.1	0.2	0	17
Control	Exploit	4	12000	1	2	1	18
LOS Link	Cmd &	5	10000	0.2	0	1	19-20
BLOS Link	Comm	5	5000	0.5	3	1	21
Mil-Sat	Comm	5	6000	1	1	0	22

Evolution of SoS Architecture

- Meta-Architecture Wave 3



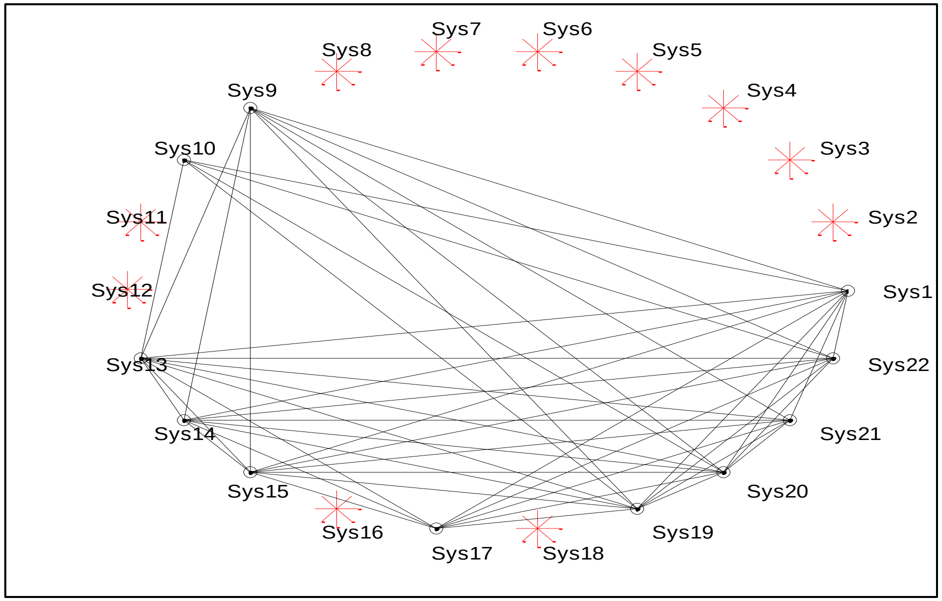
Assessment for the meta-architecture =3.59
Key Attribute values:

- Performance=2.23
- Flexibility=4
- Affordability=2.69
- Robustness=4

Systems Selected
Fighters (EO/IR) (1)
UAV-B (5,6)
UAV –C (7)
DSP(9)
Blimp (10)
JSTARS (13)
THEATRE (14,15)
MOBC2(17)
Control Station/ AOC (18)
LOS link (19,20)
BLOS Link(21)
Mil-Sat (22)

Evolution of SoS Architecture

- Negotiated Architecture Wave 3



Assessment for the meta-architecture =2.5
Key Attribute values:

- Performance=1.78
- Flexibility=2
- Affordability=3.27
- Robustness=1.29

Systems selected after negotiation S_i

Sys no	Behavior
1	selfish
9	selfish
10	opportunistic
13,14,15	cooperative
17	cooperative
19,20,21	selfish
22	cooperative

Publications

- Wang, R., Agarwal,S., & Dagli, C. (2014). Executable System of Systems Architecture Using OPM in Conjunction with Colored Petri Net: A Module for Flexible Intelligent & Learning Architectures for System of Systems, In *Europe Middle East & Africa Systems Engineering Conference (EMEASEC)*.

- Ergin, N. K.,(2014), Improving Collaboration in Search and Rescue System of Systems, *Procedia Computer Science, Volume 36*, Pages 13-20.

- Agarwal, S., & Dagli, C. H. (2013). Augmented Cognition in Human–System Interaction through Coupled Action of Body Sensor Network and Agent Based Modeling. *Procedia Computer Science*, *16*, 20-28.

- Acheson, P., Dagli, C., & Kilicay-Ergin, N. (2013). Model Based Systems Engineering for System of Systems Using Agent-based Modeling. *Procedia Computer Science*, 16, 11-19.

- Agarwal, S., Pape, L. E., & Dagli, C. H. (2014). A Hybrid Genetic Algorithm and Particle Swarm Optimization with Type-2 Fuzzy Sets for Generating Systems of Systems Architectures. *Procedia Computer Science, 36,* 57-64.

- Agarwal, S., Pape, L. E., Kilicay-Ergin, N., & Dagli, C. H. (2014). Multi-agent Based Architecture for Acknowledged System of Systems. *Procedia Computer Science, 28*, 1-10.

Publications

- Agarwal, S., Saferpour, H. R., & Dagli, C. H. (2014). Adaptive Learning Model for Predicting Negotiation Behaviors through Hybrid K-means Clustering, Linear Vector Quantization and 2-Tuple Fuzzy Linguistic Model. *Procedia Computer Science, 36,* 285-292.

- Agarwal,S., Wang, R., & Dagli, C., (2015) FILA-SoS, Executable Architectures using Cuckoo Search Optimization coupled with OPM and CPN-A module: A new Meta-Architecture Model for FILA-SoS, France, Complex Systems Design & Management (CSD&M) editor, Boulanger, Frédéric, Krob, Daniel, Morel, Gérard, Roussel, Jean-Claude, P 175-192 . Springer International Publishing.

- Pape, L., Agarwal, S., Giammarco, K., & Dagli, C. (2014). Fuzzy Optimization of Acknowledged System of Systems Meta-architectures for Agent based Modeling of Development. *Procedia Computer Science*, *28*, 404-411.

- Pape, L., & Dagli, C. (2013). Assessing robustness in systems of systems meta-architectures. *Procedia Computer Science*, *20*, 262-269.

- Pape, L., Giammarco, K., Colombi, J., Dagli, C., Kilicay-Ergin, N., & Rebovich, G. (2013). A fuzzy evaluation method for system of systems meta-architectures. *Procedia Computer Science*, *16*, 245-254.

- Acheson, P., Dagli, C., & Kilicay-Ergin, N. (2013). Model Based Systems Engineering for System of Systems Using Agent-based Modeling. *Procedia Computer Science*, *16*, 11-19.

Publications

- 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*.

- Kilicay-Ergin, N. H., Acheson, P., Colombi, J. M., & Dagli, C. H. (2012). Modeling system of systems acquisition. In *SoSE* (pp. 514-518).

- Acheson, P., Pape, L., Dagli, C., Kilicay-Ergin, N., Columbi, J., & Haris, K. (2012). Understanding System of Systems Development Using an Agent-Based Wave Model. *Procedia Computer Science*, *12*, 21-30.

- Konur, D., & Dagli, C. (2014). Military system of systems architecting with individual system contracts. *Optimization Letters*, 1-19.

- Agarwal et al., 2015 Flexible and Intelligent Learning Architectures for SoS (FILA-SoS): Architectural evolution in Systems-of-Systems, 2015 Conference on Systems Engineering Research.

- Kilicay-Ergin, Nil, and Cihan Dagli. "Incentive-Based Negotiation Model for System of Systems Acquisition." Systems Engineering 18.3 (2015): 310-321.

- Wang, R., & Dagli, C., Search Based Systems Architecture Development Using Holistic Approach (Accepted to IEEE Systems Journal with minor revisions)