SYSTEMS ENGINEERING
Research Center

Flexible and Intelligent Learning Architectures for SoS (FILA-SoS)

Volume 12 – Architecture Evolution Strategy for FILA-SoS Version 2.0

Technical Report SERC-2015-TR-021-4

February 28, 2015

Principal Investigators

Dr. Cihan H. Dagli, Missouri University of Science and Technology

Research Team

Co-PI: Dr. David Enke, Missouri S&T

Co-PI: Dr. Nil Ergin, Penn State University

Co-PI: Dr. Dincer Konur, Missouri S&T

Co-PI: Dr. Ruwen Qin, Missouri S&T

Co-PI: Dr. Abhijit Gosavi, Missouri S&T

Dr. Renzhong Wang

Missouri S&T Graduate Students

Louis Pape II, Siddhartha Agarwal and Ram Deepak Gottapu

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 (Task Order 033, RT 48). SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology

Any opinions, findings and 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.

NO WARRANTY
THIS STEVENS INSTITUTE OF TECHNOLOGY AND SYSTEMS ENGINEERING RESEARCH CENTER MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. STEVENS INSTITUTE OF TECHNOLOGY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. STEVENS INSTITUTE OF TECHNOLOGY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Multi-faceted systems of the future will entail complex logic and reasoning with many levels of reasoning in intricate arrangement. The organization of these systems involves a web of connections and demonstrates self-driven adaptability. They are designed for autonomy and may exhibit emergent behavior that can be visualized. Our quest continues to handle complexities, design and operate these systems. The challenge in Complex Adaptive Systems design is to design an organized complexity that will allow a system to achieve its goals. This report attempts to push the boundaries of research in complexity, by identifying challenges and opportunities. Complex adaptive system-of-systems (CASoS) approach is developed to handle this huge uncertainty in socio-technical systems.

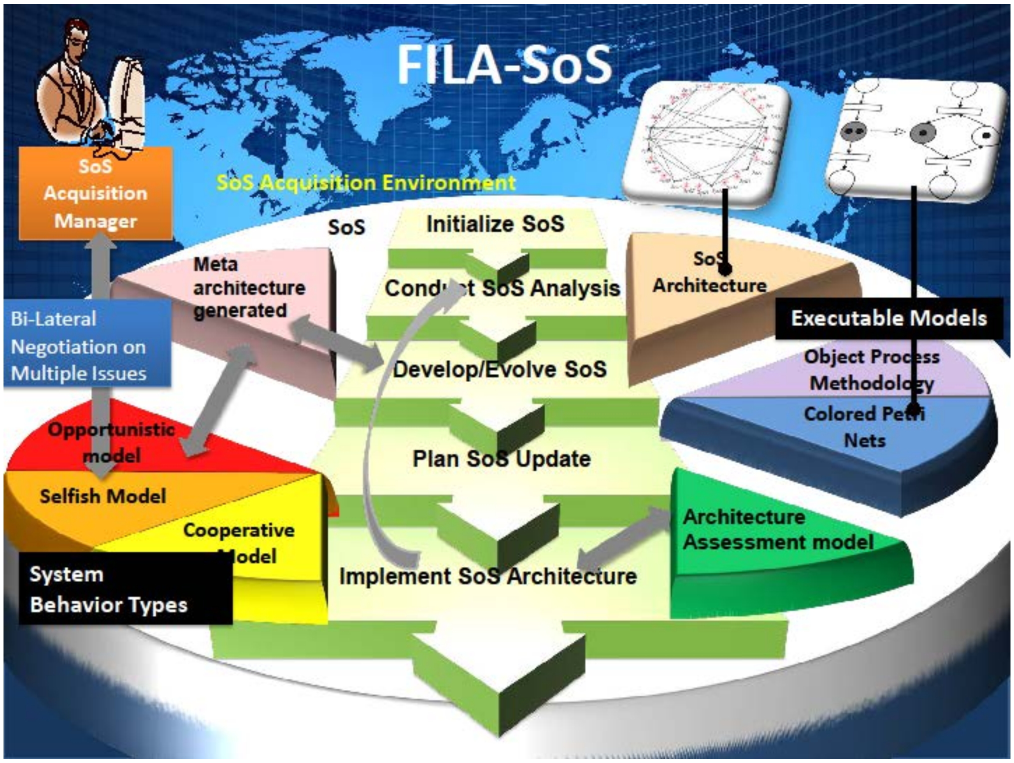
Although classically (Dahmann, Rebovich, Lowry, Lane, & Baldwin, 2011) four categories of SoS are described in literature namely; Directed, Collaborated, Acknowledged and Virtual. However, there exist infinitely many SoS on the edges of these categories thus making it a continuum. Many SoS with different configurations can fill this gap. These four types of SoS vary based on their degree of managerial control over the participating systems and their structural complexity. The spectrum of SoS ranges from Directed SoS that represents complicated systems to Virtual SoS that are complex systems.

Acknowledged SoS lie in between this spectrum. This particular SoS is the focal point of our research endeavor. Acknowledged SoS and Directed SoS share some similarities such as both have (Dahman & Baldwin, 2011) SoS objectives, management, funding and authority. Nevertheless, unlike Directed SoS, Acknowledged SoS systems are not subordinated to SoS. However, Acknowledged SoS systems retain their own management, funding and authority in parallel with the SoS. Collaborative SoS are similar to Acknowledged SoS systems in the fact that systems voluntarily work together to address shared or common interest.

Flexible and Intelligent Learning Architectures for SoS (FILA-SoS) integrated model is developed in this research task provides a decision making aid for SoS manager based on the wave model. The model developed called the 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. 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

Some of the highlights of FILA-SoS are listed in terms of its capabilities, value added to systems engineering, ability to perform "What-if Analysis", modularity of integrated models, its potential applications in the real world and future additions to the current version.



FILA-SoS has a number of unique capabilities such as integrated model for modeling and simulating SoS systems with evolution for multiple waves. It also has modularity in the structure where the models can be run independently and in conjunction with each other. Besides there are a couple of different models for both architecture generation and SoS behavior and various individual system behavior negotiation models between SoS and individual systems. In terms of value added FILA-SoS aids the SoS manager in future decision making. It also helps in understanding the emergent behavior of systems in the acquisition environment and impact on SoS architecture quality. FILA-SoS serves as an artifact to study the dynamic behavior of different type of systems (non-cooperative, semi-cooperative, cooperative). It enables us to identify intra and interdependencies among SoS elements and the acquisition environment. FILA-SoS can provide a "What-if" Analysis depending on variables such as SoS funding and capability priority that can be changed as the acquisition progresses through wave cycles. It has the ability to simulate any architecture through colored petri nets. In addition, it can simulate rules of engagement & behavior settings: all systems are non-cooperative, all systems are semi-cooperative, and all systems are cooperative or a combination. Some of the potential applications include modeling a wide variety of complex systems models such as logistics, and cyber-physical systems. It also acts as a test-bed for decision makers to evaluate operational guidelines and principles for managing various acquisition environment scenarios. Future Capabilities that are currently in progress are extending the model to include multiple interface alternatives among systems and incorporation of risk models into environmental scenarios.

Integrated Model Structure for FILA-SoS Version 1.0 is described. It provides a short description of all independent models that make up the FILA-SoS integrated model and reports the workings of the model with three notional System-of-Systems namely; Toy Problem for aircraft carrier performance assessment, ISR (intelligence surveillance and reconnaissance) and SAR (search and rescue).

The project reports span 17 volumes. Each report describes the various aspects of the FILA-SOS integrated model:

Volume 1: Integrated Model Structure

Volume 1 is the Integrated Model Structure report for FILA-SoS Version 1.0. It provides a short description of all independent models that make up the FILA-SoS integrated model. Integrated FILA-SoS developed is tested in three notional System-of-Systems namely; Toy Problem for Aircraft Carrier Performance Assessment, ISR (intelligence surveillance and reconnaissance) and SAR (search and rescue). FILA-SoS integrated model is currently being validated with a real life data from a medium sized SoS. The results of this validation are given in volume 17.

Volume 2: Meta-Architecture Generation Multi-Level Model

Volume 2 describes Meta-Architecture Generation Multi-Level Model. The multi-level meta-architecture generation model considers constructing an SoS architecture such that each capability is provided by at least one system in the SoS and the systems in the SoS are able to communicate with each other. Secondly, it has multiple objectives for generating a set of SoS architectures namely; maximum total performance, minimum total costs and minimum deadline. Finally, the model establishes initial contracts with systems to improve performances.

Volume 3: Fuzzy-Genetic Optimization Model

Volume 3 illustrates the second meta-architecture generation model known as the Fuzzy-Genetic optimization model. This model is based on evolutionary multi-objective optimization for SoS architecting using genetic algorithms and four key performance attributes (KPA) as the objective functions. It also has a type-1 fuzzy assessor for dynamic assessment of domain inputs and that forms the fitness function for the genetic algorithm. It returns the best architecture (meta-architecture) consisting of systems and their interfaces. It is a generalized method with application to multiple domains such as Gulf War Intelligence/Surveillance/Reconnaissance Case, Aircraft Carrier Performance Assessment Case and Alaskan Maritime Search and Rescue Case.

Volume 4: Architecture Assessment Model

Volume 4 describes an Architecture Assessment Mode that can capture the non-linearity in key performance attribute (KPA) tradeoffs, is able to accommodate any number of attributes for a selected SoS capability, and incorporate multiple stakeholder's understanding of KPA's. Assessment is based on a given meta-architecture alternative. This is done using type-1 fuzzy sets and fuzzy inference engine. The model provides numerical values for meta-architecture quality.

Volume 5: Cooperative System Negotiation Model

Volume 5 specifically describes the Cooperative System Negotiation Model. The systems following this model behave cooperatively while negotiating with the SoS manager. The model of cooperative behavior is based on agent preferences and the negotiation length. Each system agent has two inherent behaviors of cooperativeness: Purposive (normal behavior) and Contingent (behavior driven by unforeseen circumstances). The approach models the tradeoff between the two behaviors for the systems. A fuzzy weighted average approach is used to arrive at the final proposed value.

Volume 6: Non-Cooperative System Negotiation Model
Volume 6 goes on to describe the Non-Cooperative System Negotiation Model in which systems behave in their self-interest while negotiating with the SoS coordinator. A mathematical model of individual system's participation capability and self-interest negotiation behavior is created. This methodology is an optimization-based generator of alternatives for strategically negotiating multiple items with multiple criteria. Besides, a conflict evaluation function that estimates prospective outcome for identified alternative is proposed.

Volume 7: Semi-Cooperative System Negotiation Model
Volume 7 describes the third and last system negotiation model, which illustrates the Semi-Cooperative System Negotiation Model. It exhibits the capability of being flexible or opportunistic: i.e., extremely cooperative or uncooperative based on different parameter values settings. A Markov-chain based model designed for handling uncertainty in negotiation modeling in an SoS. A model based on Markov chains is used for estimating the outputs. The work assigned by the SoS to the system is assumed to be a ``project'' that takes a random amount of time and a random amount of resources (funding) to complete.

Volume 8: Incentive based Negotiation Model for System of Systems
Volume 8 explains the SoS negotiation model also called the Incentive Based Negotiation Model for System of Systems. This model is based on two key assumptions that are to design a contract to convince the individual systems to join the SoS development and motivate individual systems to do their tasks well. Game theory and incentive based contracts are used in the negotiation model that will maximize the welfare for parties involved in the negotiation. SoS utility function takes into account local objectives for the individual systems as well as global SoS objective whereas the incentive contract design persuades uncooperative systems to join the SoS development.

Volume 9: Model for Building Executable Architecture
Volume 9 illustrates the process of building Executable Architectures for SoS. The operations of the SoS is a dynamic process with participating system interacting with each other and exchange various kinds of resources, which can be abstract information or physical objects. This is done through a hybrid structure of OPM (Object process methodology) and CPN (Colored petri nets) modeling languages. The OPM model is intuitive and easy to understand. However, it does not support simulation, which is required for accessing the behavior related performance. This is achieved by mapping OPM to CPN, which is an executable simulation language. The proposed method can model the interactions between components of a system or subsystems in SoS. In addition, it can capture the dynamic aspect of the SoS and simulate the behavior of the SoS. Finally, it can access various behavior related performance of the SoS and access different

constitutions or configurations of the SoS which cannot be incorporated into the meta-architecture generation models of Volume 2 & 3.

Volume 10: Integrated Model Software Architecture and Demonstration FILA-SoS Version 1.0
Volume 10 elucidates the Integrated Model Software Architecture and Demonstration based on the models described above. Volume 11 and thereon the reports are aimed at the upcoming newer version 2.0 of FILA-SoS.

Volume 11: Integrated Model Structure FILA-SoS Version 2.0
Volume 11 provides Integrated Model Structure for FILA-SoS Version 2.0 that could be implemented in a new software environment.

Volume 12: Complex Adaptive System-of-System Architecture Evolution Strategy Model for FILA-SoS Version 2.0
Volume 12 provides a model to answer the first research question "What is the impact of different constituent system perspectives regarding participating in the SoS on the overall mission effectiveness of the SoS?" It is named the Complex Adaptive System-of-System Architecture Evolution Strategy Model and is incorporated in FILA-SoS Version 2.0. This volume describes a computational intelligence based strategy involving meta-architecture generation through evolutionary algorithms, meta-architecture assessment through type-2 fuzzy nets and finally its implementation through an adaptive negotiation strategy.

Volume 13: On the Flexibility of Systems in System of Systems Architecting: A new Meta-Architecture Generation Model for FILA-SoS Version 2.0
Volume 13 is termed the Flexibility of Systems in System of Systems Architecting: A new Meta-Architecture Generation Model for FILA-SoS Version 2.0. The research question is answered through an alternative technique to meta-architecture generation besides the one described in Volume 2.

Volume 14: Assessing the Impact on SoS Architecture Different Level of Cooperativeness: A new Model for FILA-SoS Version 2.0
Volume 14 proposes a new method for Assessing the Impact on SoS Architecture Different Level of Cooperativeness. Second research question is answered through a model that allows different levels of cooperativeness of individual systems.

Volume 15: Incentivizing Systems to Participate in SoS and Assess the Impacts of Incentives: A new Model for FILA-SoS Version 2.0
Volume 15 is an extension of previous systems negotiation models based on incentivizing and is aptly called Incentivizing Systems to Participate in SoS and Assess the Impacts of Incentives: A new Model for FILA-SoS Version 2.0. It also provides an approach to answer the third research question "How should decision-makers incentivize systems to participate in SoS, and better understand the impact of these incentives during SoS development and effectiveness?". This model is based on the fact that providing incentives only depending on the outcome may not be enough to attract the attention of the constituent systems to participate in SoS mission. Therefore, this model extends the approach as described in Volume 8 while considering the

uncertainty in the acquisition environment. The incentive contract is designed based on the objectives of the SoS and the individual systems. Individual system's objective is to secure highest incentives with minimal effort while the SoS manager's goal is to convince individual systems to join the SoS development while maximizing its own utility.

Volume 16: Integrated Model Software Architecture for FILA-SoS Version 2.0
Volume 16 gives an overview of the integrated model architecture in version 2.0 of the software. It includes all old and new models previously mentioned.

Volume 17: FILA-SoS Version 1.0 Validation with Real Data
Volume 17 describes the validation of the FILA-SoS Version 1.0 with a real life data provided by MITRE Corporation by from a moderately sized SoS.

INTRODUCTION

MOTIVATION FOR RESEARCH

In the real world, systems are complex, non-deterministic, evolving, and have human centric capabilities. The connections of all complex systems are non-linear, globally distributed, and evolve both in space and in time. Because of non-linear properties, system connections create an emergent behavior. It is imperative to develop an approach to deal with such complex large-scale systems. The approach and goal is not to try and control the system, but design the system such that it controls and adapts itself to the environment quickly, robustly, and dynamically. These complex entities include both socioeconomic and physical systems, which undergo dynamic and rapid changes. Some of the examples include transportation, health, energy, cyber physical systems, economic institutions and communication infrastructures.

In addition, the idea of “System-of-Systems” is an emerging and important multidisciplinary area. An SoS is defined as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities greater than the sum of the capabilities of the constituent parts. Either of the systems alone cannot independently achieve the overall goal. System-of- Systems (SoS) consists of multiple complex adaptive systems that behave autonomously but cooperatively (Dahman, Lane, Rebovich, & Baldwin, 2008). The continuous interaction between them and the interdependencies produces emergent properties that cannot be fully accounted for by the “normal” systems engineering practices and tools. System of Systems Engineering (SoSE), an emerging discipline in systems engineering is attempting to form an original methodology for SoS problems (Luzeaux, 2013).

Since SoS grow in complexity and scale with the passage of time it requires architectures that will be necessary for understanding and governance and for proper management and control. Systems architecting can be defined as specifying the structure and behavior of an envisioned system. Classical system architecting deals with static systems whereas the processes of System of Systems (SoS) architecting has to be first done at a meta-level. The architecture achieved at a meta-level is known as the meta-architecture. The meta-architecture sets the tone of the architectural focus (Malan & Bredemeyer, 2001). It narrows the scope of the fairly large domain space and boundary. Although the architecture is still not fixed but meta-architecture provides multiple alternatives for the final architecture. Thus architecting can be referred to as filtering the meta-architectures to finally arrive at the architecture. The SoS architecting involves multiple systems architectures to be integrated to produce an overall large scale system meta-architecture for a specifically designated mission (Dagli & Ergin, 2008). SoS achieves the required goal by introducing collaboration between existing system capabilities that are required in creating a larger capability based on the meta-architecture selected for SoS. The level of the degree of influence on individual systems architecture through the guidance of SoS manager in implementing SoS meta-architecture can be classified as directed, acknowledged, collaborative and virtual. Acknowledged SoS have documented objectives, an elected manager and defined resources for the SoS. Nonetheless, the constituent systems retain their independent ownership, objectives, capital, development, and sustainment approaches. Acknowledged SoS shares some similarities with directed SoS and collaborative SoS. There are four types of SoS that are described below:

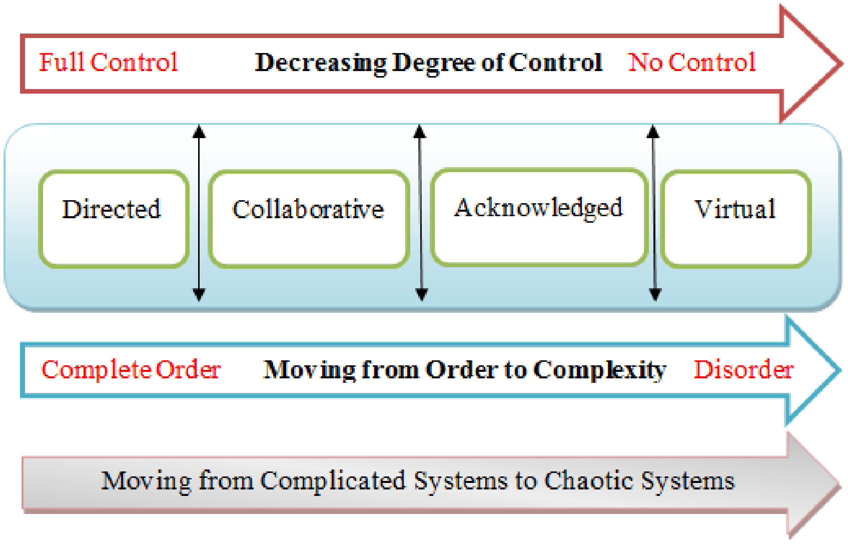

Full Control
Decreasing Degree of Control
No Control
Directed
Collaborative
Acknowledged
Virtual
Complete Order
Moving from Order to Complexity
Disorder
Moving from Complicated Systems to Chaotic Systems

Figure 1 Schematic Drawing of Four Classical Types of SoS Based on Degree of Control and Degree of Complexity

Virtual

- Virtual SoS lack a central management authority and a centrally agreed upon purpose for the system-of-systems.
- Large-scale behavior emerges—and may be desirable—but this type of SoS must rely upon relatively invisible mechanisms to maintain it.

Collaborative

- In collaborative SoS the component systems interact more or less voluntarily to fulfill agreed upon central purposes.

Acknowledged (*FILA-SoS integrated model is based on Acknowledged SoS*)

- Acknowledged SoS have recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, and development and sustainment approaches.
- Changes in the systems are based on collaboration between the SoS and the system.

Directed

- Directed SoS's are those in which the integrated system-of-systems is built and managed to fulfill specific purposes.
- It is centrally managed during long-term operation to continue to fulfill those purposes as well as any new ones the system owners might wish to address.
- The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose.

This research is based on Acknowledged SoS. The major objectives of the reasearch are:

- To develop a simulation for Acknowledged SoS architecture selection and evolution.
- To have a structured, repeatable approach for planning and modeling.
- To study and evaluate the impact of individual system behavior on SoS capability and architecture evolution process.

The dynamic planning for a SoS is a challenging endeavor. Department of Defense (DoD) programs constantly face challenges to incorporate new systems and upgrade existing systems over a period of time under threats, constrained budget, and uncertainty. It is therefore necessary for the DoD to be able to look at the future scenarios and critically assess the impact of technology and stakeholder changes. The DoD currently is looking for options that signify affordable acquisition selections and lessen the cycle time for early acquisition and new technology addition. FILA-SoS provides a decision aid in answering some of the questions.

This volume gives an overview of a novel methodology known as the Flexible Intelligent & Learning Architectures in System-of-Systems (FILA-SoS). Some the challenges that are prevalent in SoS architecting and how FILA-SoS attempts to address them is explained in the next section.

SYSTEM OF SYSTEM CHALLENGES

All these recent developments are helping us to understand Complex Adaptive Systems. They are at the edge of chaos as they maintain dynamic stability through constant self-adjustment and evolution. Chaos and order are two complementary states of our world. A dynamic balance exists between these two states.

Order and structure are vital to life. Order ensures consistency and predictability and makes the creation of systems possible. However, too much order leads to rigidity and suppresses creativity. Chaos constantly changes the environment creating disorder and instability but can also lead to emergent behavior and allows novelty and creativity. Thus, sufficient order is necessary for a system to maintain an ongoing identity, along with enough chaos to ensure growth and development. The challenge in Complex Adaptive Systems design is to design an organized complexity that will allow a system to achieve its goals. SoS is a complex systems by its nature due to the following characteristics that are component systems are operationally independent elements and also managerially independent of each other. This means that component systems preserve existing operations independent of the SoS. SoS has an evolutionary development and due to the large scale complex structure shows an emergent behavior. Emergence means the SoS performs functions that do not reside in any one component system.

2012 INCOSE SoS working group survey identified seven ‘pain points’ raising a set of questions for systems engineering of SoS which are listed in Table 1 (Dahman, 2012).

Table 1 System of Systems and Enterprise Architecture Activity

Pain Points	Question
Lack of SoS Authorities & Funding	What are effective collaboration patterns in systems of systems?
Leadership	What are the roles and characteristics of effective SoS leadership?
Constituent Systems	What are effective approaches to integrating constituent systems into a SoS?
Capabilities & Requirements	How can SE address SoS capabilities and requirements?
Autonomy, Interdependencies & Emergence	How can SE provide methods and tools for addressing the complexities of SoS interdependencies and emergent behaviors?
Testing, Validation & Learning	How can SE approach the challenges of SoS testing, including incremental validation and continuous learning in SoS?
SoS Principles	What are the key SoS thinking principles, skills and supporting examples?

The importance and impact on systems engineering of each pain point is illustrated below:

- **Lack of SoS Authorities & Funding and Leadership** pose several and severe governance and management issues for SoS. This conditions has a large impact on the ability to implement systems engineering (SE) in the classical sense to SoS. In addition, this problem affects the modeling & simulation activities.
- **Constituent Systems** play a very important role in the SoS. As explained earlier usually they have different interests and ambitions to achieve, which may or may not be aligned with the SoS.. Similarly models, simulations and data for these systems will naturally have to be attuned to the specific needs of the systems, and may not lend themselves easily to supporting SoS analysis or engineering
- **Autonomy, Interdependencies & Emergence** is ramifications of the varied behaviors and interdependencies of the constituent systems making it complex adaptive systems. Emergence comes naturally in such a state, which is often unpredictable. While modeling & simulation can aid in representing and measuring these complexities, it is often hard to achieve real life emergence. This is due to limited understanding of the issues that can bring up serious consequences during validation.
- **Capability of the SoS** and the individual systems capability needs may be high level and need definition in order to align them with the requirements of the SoS mission. The SoS mission is supported by constituent systems, which may not be able (or willing) to address them.
- **Testing, Validation & Learning** becomes difficult since the constituent systems continuously keep evolving, adapting, as does the SoS environment which includes stakeholders, governments, etc. Therefore creating a practical test-bed for simulating the large dynamic SoS is a challenge in itself. Again modeling & simulation can solve part of the problem such as enhancing live test and addressing risk in SoS when testing is not feasible; however, this requires a crystal clear representation of the SoS which can be difficult as discussed in earlier points.

- **SoS Principles** are still being understood and implemented. Therefore, the rate of success is yet to be addressed formally. This poses some pressure on the progress of SoS engineering. Similarly, there is an absence of a well-established agreeable space of SoS principles to drive development and knowledge. This constricts the effective use of potentially powerful tools.

The DoD 5000.2 is currently used as the acquisition process for complex systems. Schwartz (2010) described this process as an extremely complex systemic process that cannot always constantly produce systems with expected either cost or performance potentials. The acquisition in DoD is an SoS problem that involves architecting, placement, evolution, sustainment, and discarding of systems obtained from a supplier or producer. Numerous attempts undertaken to modify and reform the acquisition process have found this problem difficult to tackle because the models have failed to keep pace with actual operational scenarios. Dombkins (1996) offered a novel approach to model complex projects as waves. He suggested that there exists a major difference in managing and modeling traditional projects versus complex projects. He further illustrated his idea through a wave planning model that exhibits a linear trend on a time scale; on a spatial scale, it tries to capture the non-linearity and recursiveness of the processes. In general, the wave model is a developmental approach that is similar to periodic waves. A period, or multiple periods, can span a strategic planning time. The instances within the periods represent the process updates. A recently proposed idea (Dahman, Lane, Rebovich, & Baldwin, 2008) that SoS architecture development for the DoD acquisition process can be anticipated to follow a wave model process. According to Dahman DoD 5000.2 may not be applicable to the SoS acquisition process. Acheson (2013) proposed that Acknowledged SoS be modeled with an Object-Oriented Systems Approach (OOSA). Acheson also proposes that for the development of SoS, the objects should be expressed in the form of a agent based model.

The environment and the systems are continuously changing. Let there be an initial environment model, which represents the SoS acquisition environment. As the SoS acquisition progresses through, these variables are updated by the SoS Acquisition Manager to reflect current acquisition environment. Thus, the new environment model at a new time has different demands. To fulfill the demands of the mission a methodology is needed to assess the overall performance of the SoS in this dynamic situation. The motivation of evolution are the changes in the SoS environment (Chattopadhyay, Ross, & Rhodes, 2008). The environmental changes consist of:

- SoS Stakeholder Preferences for key performance attributes
- Interoperability conditions between new and legacy systems
- Additional mission responsibilities to be accommodated
- Evolution of individual systems within the SoS

Evaluation of architectures is another SoS challenge area as it lends itself to a fuzzy approach because the criteria are frequently non-quantitative, or subjective (Pape & Dagli, 2013), or based on difficult to define or even unpredictable future conditions, such as “robustness.” Individual attributes may not have a clearly defined, mathematically precise, linear functional form from worst to best. The goodness of one attribute may or may not offset the badness of another attribute. Several moderately good attributes coupled with one very poor attribute may be better than an architecture with all marginally good attributes, or vice-versa. A fuzzy approach allows many of these considerations to be handled using a reasonably simple set of rules, as well as having the ability to include non-linear characteristics in the fitness measure. The simple rule set allows small adjustments to be made to the model to see how seemingly small changes affect the outcome. The methodology outlined in this research and technical report falls under a multi-level plug-and-play type of modeling approach to address various aspects of SoS acquisition environment: SoS architecture evaluation, SoS architecture evolution, and SoS acquisition process dynamics including behavioral aspects of constituent systems.

HOW DOES FILA-SOS ADDRESS SOS PAIN POINTS

The first pain point is Lack of SoS Authorities & Funding which begs a question "What are effective collaboration patterns in systems of systems?"

Since there is lack of SoS Authority but more so persuasion involved in the workings of a SoS, systems are allowed to negotiate with the SoS manager. Deadline for preparation, funding and performance required to complete the mission are some of the issues that form the negotiation protocol. Besides different combination of behavior types assigned to the systems can help us gauge the best effective collaboration patterns in systems of systems after the end of negotiations.

The leadership issues pose the question, "What are the roles and characteristics of effective SoS leadership?" This is addressed by incorporating views from multiple stakeholders while assessing the architecture's quality. In addition, we maintain that the characteristics are similar to what an Acknowledged SoS manager would have while distributing funds and resources among systems for a joint operation. The SoS manager also has the opportunity to form his decision based on most likely future scenarios, thus imparting him an edge as compared to other models. This will improve the process of acquisition in terms of overall effectiveness, less cycle time and integrating legacy systems. Overall, the role of the leadership is presented a guide than someone who would foist his authority.

The third pain point question, "What are effective approaches to integrating constituent systems into a SoS? is addressed below. A balance has to be maintained during acquisition between amount of resources used and the degree of control exercised by the SoS manager on the constituent systems. The meta-architecture generation is posed as a multi-objective optimization problem to address this pain point. The constituent systems and the interfaces between them are selected while optimizing the resources such as operations cost, interfacing cost, performance levels etc. The optimization approach also evaluates the solutions based on views of multiple stakeholders integrated together using a fuzzy inference engine.

How can SE address capabilities and requirements? is the fourth pain point and is answered in this paragraph. Organizations that acquire large-scale systems have transformed their attitude to acquisition. Hence, these organizations now want solutions to provide a set of capabilities, not a single specific system to meet an exact set of specifications. During the selection process of systems it is ensured that, a single capability is provided by more than one system. The idea is to choose at least one systems having unique capability to form the overall capability of the SoS.

The fifth pain point on autonomies, emergence and interdependencies is one of the most important objectives of this research. This objective can be described as “How can SE provide methods and tools for addressing the complexities of SoS interdependencies and emergent behaviors?”. Each system has an autonomous behavior maintained through pre-assigned negotiation behaviors, differ operations cost, interfacing cost and performance levels while providing the same required capability. The interfacing among systems is encouraged to have net-centric architecture. The systems communicate to each other through several communication systems. This ensures proper communication channels. Together the behavior and net-centricity make it complex systems thus bringing out the emergence needed to address the mission.

FILA-SoS is an excellent integrated model for addressing the complexities of SoS interdependencies and emergent behaviors as explained in the above paragraphs.

As for the sixth pain point on testing, validation and learning goes, FILA-SoS has been tested on three notional examples so far the ISR, Search and Rescue (SAR) and the Toy problem for Aircraft Carrier Performance Assessment. For ISR (refer to Figure 2) a guiding physical example is taken from history. During the 1991 Gulf War, Iraqi forces used mobile SCUD missile launchers called Transporter Erector Launchers (TELS) to strike at Israel and Coalition forces with ballistic missiles. Existing intelligence, surveillance, and reconnaissance (ISR) assets were inadequate to find the TELs during their vulnerable setup and knock down time. The “uninhabited and flat” terrain of the western desert was in fact neither of those things, with numerous Bedouin goat herders and their families, significant traffic, and thousands of wadis with culverts and bridges to conceal the TELs and obscure their movement.

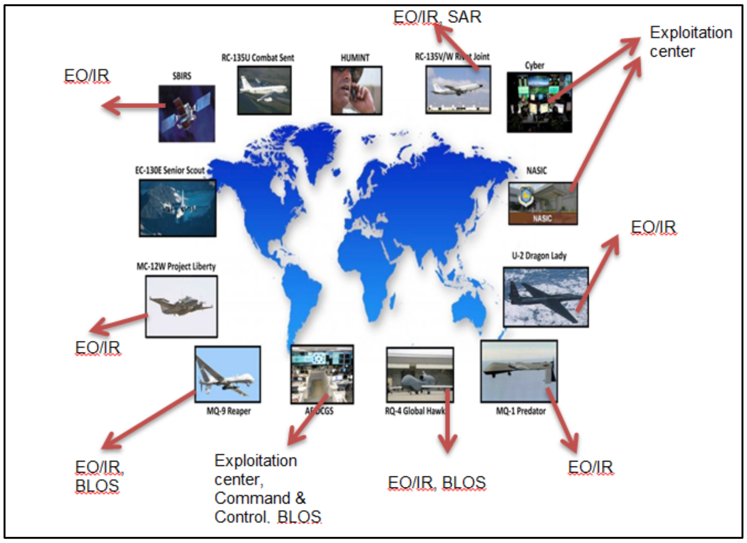


Figure 2 ISR System-of-Systems for Testing FILA-SoS

A Coast Guard Search and Rescue (SAR) (Figure 3) SoS engineering and development problem is selected for serving the Alaskan coast. Detailed information about this case study can be found in Dagli et al (2013). There is increasing use of the Bering Sea and the Arctic by commercial fisheries, oil exploration and science, which increases the likelihood of occurrence of possible SAR scenarios.

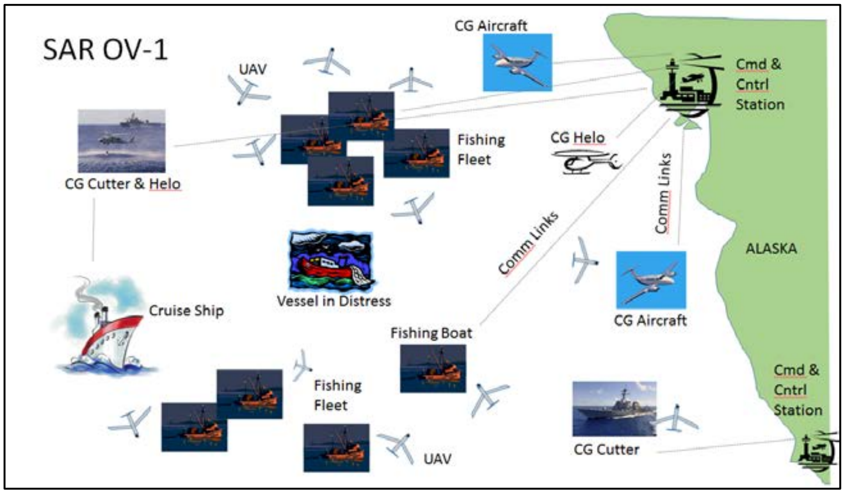


Figure 3 SAR System-of-Systems for Testing FILA-SoS

The toy problem for assessing the performance of the aircraft carrier involves multiple systems such as satellites, uav's and ground station that support the aircraft carrier to fulfill the mission (refer to Figure 4). The results have been obtained for multiple waves of the evolution process for all the examples.

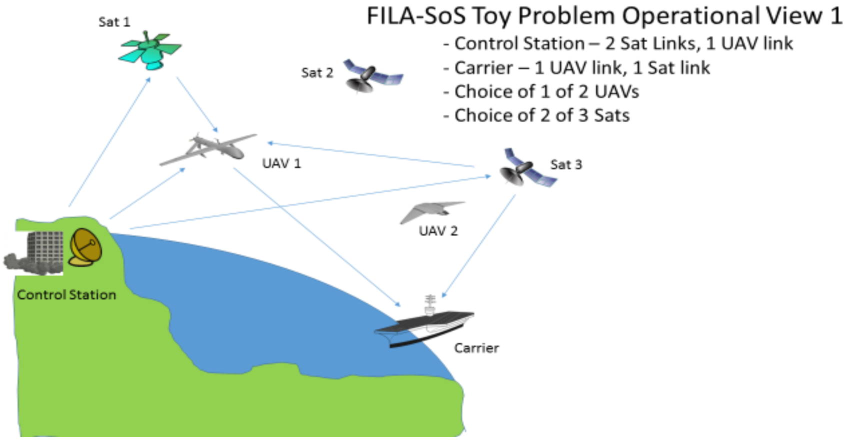

FILA-SoS Toy Problem Operational View 1
- Control Station – 2 Sat Links, 1 UAV link
- Carrier – 1 UAV link, 1 Sat link
- Choice of 1 of 2 UAVs
- Choice of 2 of 3 Sats
Sat 1
Sat 2
Sat 3
UAV 1
UAV 2
Control Station
Carrier

Figure 4 Aircraft Carrier Performance Assessment for Testing FILA-SoS

These example discussed above clearly show the domain independence of FILA-SoS.

FILA-SoS is a novel method of making sequential decisions over a period for SoS development. The goal is to apply the integrated model to dynamically evolve SoS architecture and optimize SoS architecture, design and validate through simulation tools. The integrated model structure can be applied to various application areas including development of dynamic water treatment SoS architecture, development of dynamic Air Traffic Management SoS, and development of autonomous ground transport SoS. FILA-SoS has a number of abilities that make it unique such as:

- Aiding the SoS manager in future decision making
- To assist in understanding the emergent behavior of systems in the acquisition environment and impact on SoS architecture quality
- To facilitate the learning of dynamic behavior of different type of systems (cooperative, semi-cooperative , non-cooperative)
- Identifying intra and interdependencies among SoS elements and the acquisition environment
- Modeling and application to a wide variety of complex systems models such as logistics, cyber-physical systems and similar systems
- Acting as a Test-bed for decision makers to evaluate operational guidelines and principles for managing various acquisition environment scenarios
- Appropriate to model SoS that evolve over a period of time under uncertainties by multiple wave simulation capability.

Overview of the FILA-SoS integrated model

In this section an overview of FILA-SoS is described. The model developed called the FILA-SoS is 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. FILA-SoS and the Wave Process address four of the most challenging aspects of system-of-system architecting:

- Dealing with the uncertainty and variability of the capabilities and availability of potential component systems.
- Providing for the evolution of the system-of-system needs, resources and environment over time.
- Accounting for the differing approaches and motivations of the autonomous component system managers.
- Optimizing system-of-systems characteristics in an uncertain and dynamic environment with fixed budget and resources

Definition of Variables for SoS

This list comprises of the notation for variables used to solve the Acknowledged SoS architectural evolution problem:

C: Overall capability (the overall goal to be achieved by combining sub-capabilities)
c_j: *j* ∈ ***J***, ***J***= *{1, 2,…, M}*:
Constituent system capabilities required
s_i: *i* ∈ ***I***, ***I***= *{1, 2,…, N}*:
Total number of systems present in the SoS problem
Let $\boldsymbol{A}$ be a N x $M - matrix\ of\ a_{ij}\ where$
$a_{ij} = 1\ if$ capability j is possessed by system i
$a_{ij} = 0\ otherwise$
P_i: Performance of system i for delivering all capabilities $\sum_j a_{ij}$
F_i: Funding of system i for delivering all capabilities $\sum_j a_{ij}$
D_i: Deadline to participate in this round of mission development for system i
IF_{ik} Interface between systems $i\ and\ k\ s.t.\ s \neq\ k$, *k* ∈ *I*
IC_i: The cost for development of interface for system i
OC_i: The cost of operations for system i
KP_r : *r* ∈ ***R***, *R*= *{1, 2,…, Z}*:
The key performance attributes of the SoS
FA: Funding allocated to SoS Manager
p= {1, 2,…, P}:
Number of negotiation attributes for bilateral negotiation
t_{max}: Total round of negotiations possible

t : Current round of negotiation (epochs)
t_{max}: Total round of negotiations possible
$V_{pi}^{SoS}(t)$: The value of the attribute p for SoS manager at time t for system i
$V_{pi}^{S}(t)$: The value of the attribute p for system i owner at time t
TQ: Threshold architecture quality

The model involves a list of stakeholders such as the Acknowledged SoS manager, system owners/managers, SoS environment etc.

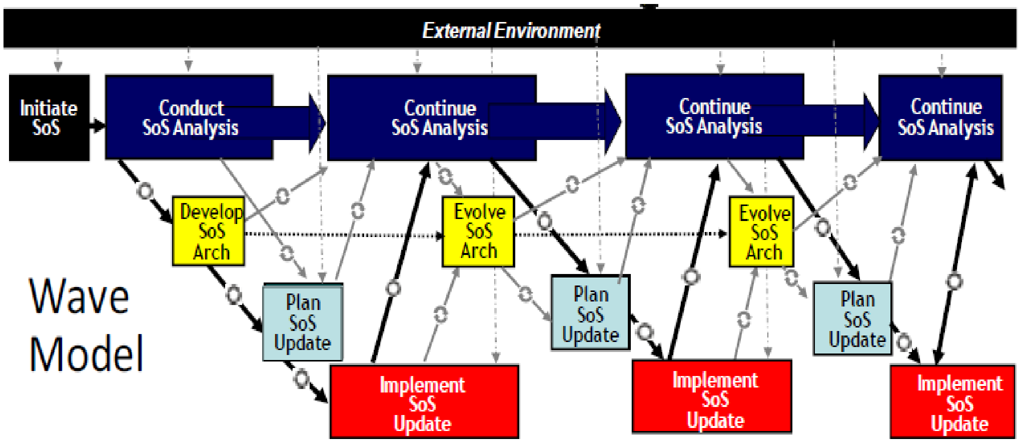


Figure 5 The Wave Model of SoS initiation, Engineering, and Evolution

FILA-SoS follows the Dahmann's proposed SoS Wave Model process for architecture development of the DoD acquisition process as depicted in Figure 5. FILA-SoS addresses the most important challenges of SoS architecting in regards to dealing with the uncertainty and variability of the capabilities and availability of potential component systems. The methodology also provides for the evolution of the system-of-system needs, resources and environment over time while accounting for the differing approaches and motivations of the autonomous component system managers. FILA-SoS assumes to have an uncertain and dynamic environment with fixed budget and resources for architecting SoS. The overall idea being to select a set of systems and interfaces based on the needs of the architecture in a full cycle called the wave. Within the wave, there may be many negotiation rounds, which are referred to as epochs. After each wave, the systems selected during negotiation in the previous wave remain as part of the meta-architecture whilst new systems are given a chance to replace those left out as a result.

Processes involved in the wave model and their analog in FILA-SoS can be explained through the first stage of Initializing the SoS. In terms of initializing, wave process requires to understand the SoS objectives and operational concept (CONOPS), gather information on core systems to support desired capabilities. This starts with the overarching capability C desired by Acknowledged SoS manager and defining the c_j or sub-capabilities required to produce capability C and FA, funding allocated to SoS Manager. These also form the input to the FILA-SoS for the participating systems s_i. FILA-SoS requires t_{max} the number of negotiation cycles, selection of the meta-architecture modelling procedure and system negotiation models assigned to participating systems.

The second stage is called the Conduct_SoS_Analysis. For the Wave process, it represents starting an initial SoS baseline architecture for SoS engineering based on SoS requirements space, performance measures, and relevant planning elements. For FILA-SoS the baseline architecture is called as the meta-architecture. Meta-architecture is basically picking up the systems s_i and their respective capabilities a_{ij}. Meta-architecture modelling requires the values for KP_t , the key performance attributes of the SoS, P_i (Performance of system i) , F_i (Funding of system i), and D_i deadline to participate in this round of mission development for system i which is assumed to be the total for all capabilities possessed by system i. The cost for development of a single interface for system i, IC_i and OC_i the cost of operations for system i is also needed at this stage of the model. The next step is the Develop/ Evolve SoS. In this case in terms of the Wave process essential changes in contributing systems in terms of interfaces and functionality in order to implement the SoS architecture are identified. Within FILA-SoS this signals the command to send connectivity request to individual systems and starting the negotiation between SoS and individual systems. This stage requires the number of negotiation attributes P for a bilateral negotiation between Acknowledged SoS manager and each systems i selected in the meta-architecture and t_{max} which denotes the total round of negotiations possible.

The next phase is Plan SoS Update in Wave process. In this, phase the architect plans for the next SoS upgrade cycle based on the changes in external environment, SoS priorities, options and backlogs. There is an external stimulus from the environment, which affects the SoS architecture. To reflect that in FILA-SoS determines which systems to include based on the negotiation outcomes and form a new SoS architecture. Finally, the last stage in Wave process is Implement SoS Architecture which establishes a new SoS baseline based on SoS level testing and system level implementation. In the FILA-SoS the negotiated architecture quality is evaluated based on KP_r, key performance attributes of the SoS. If the architecture quality is not up to a predefined quality or TQ the threshold architecture quality the Acknowledged SoS manager and systems i selected in the meta-architecture go for renegotiations. Finally the process moves on to the next acquisition wave. The evolution of SoS should take into account availability of legacy systems and the new systems willing to join, adapting to changes in mission and requirement, and sustainability of the overall operation. FILA-SoS also has the proficiency to convert the meta-architecture into an executable architecture using the Object Process Model (OPM) and Colored Petri Nets (CPN) for overall functionality and capability of the meta-architecture. 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.

Some of the highlights of FILA-SoS are described in terms of its capabilities, value added to systems engineering, ability to perform “What-if Analysis”, modularity of integrated models, its potential applications in the real world and future additions to the current version. The most important capability of FILA-SoS is it being an integrated model for modeling and simulating SoS systems with evolution for multiple waves. Secondly, all models within FILA-SoS can be run independently and in conjunction with each other. Thirdly, there are two model types that represent SoS behavior and various individual system behaviors. Finally, it has the capacity to study negotiation dynamics between SoS and individual systems.

The value added by FILA-SoS to systems engineering is it aids the SoS manager in future decision making, can help in understanding the emergent behavior of systems in the acquisition environment and its impact on SoS architecture quality. Besides, it has three independent systems behavior models, which are referred to as cooperative, semi-cooperative and non-cooperative. These behavior models are used to Study the dynamic behavior of different type of systems while they are negotiating with SoS manager. In addition, FILA-SoS assists in identifying intra and interdependencies among SoS elements and the acquisition environment.

FILA-SoS also can facilitate a "What-if" Analysis using variables such as SoS funding and capability priority that can be changed as the acquisition progresses though wave cycles. The parameter setting for all negotiation models can be changed and rules of engagement can be simulated for different combinations of systems behaviors.

Potential Application of FILA-SoS include complex systems models such as logistics, cyber-physical systems. In addition, it can act as test-bed for decision makers to evaluate operational guidelines and principles for managing various acquisition environment scenarios. While the future capabilities that we would like to be included are extending the model to include multiple interface alternatives among systems and incorporation of risk models into environmental scenarios.

INDEPENDENT MODULES OF FILA-SOS

The FILA-SoS has a number of independent modules that are integrated together for meta-architecture generation, architecture assessment, meta-architecture executable model, and meta-architecture implementation through negotiation. An overall view is presented in Figure 6.

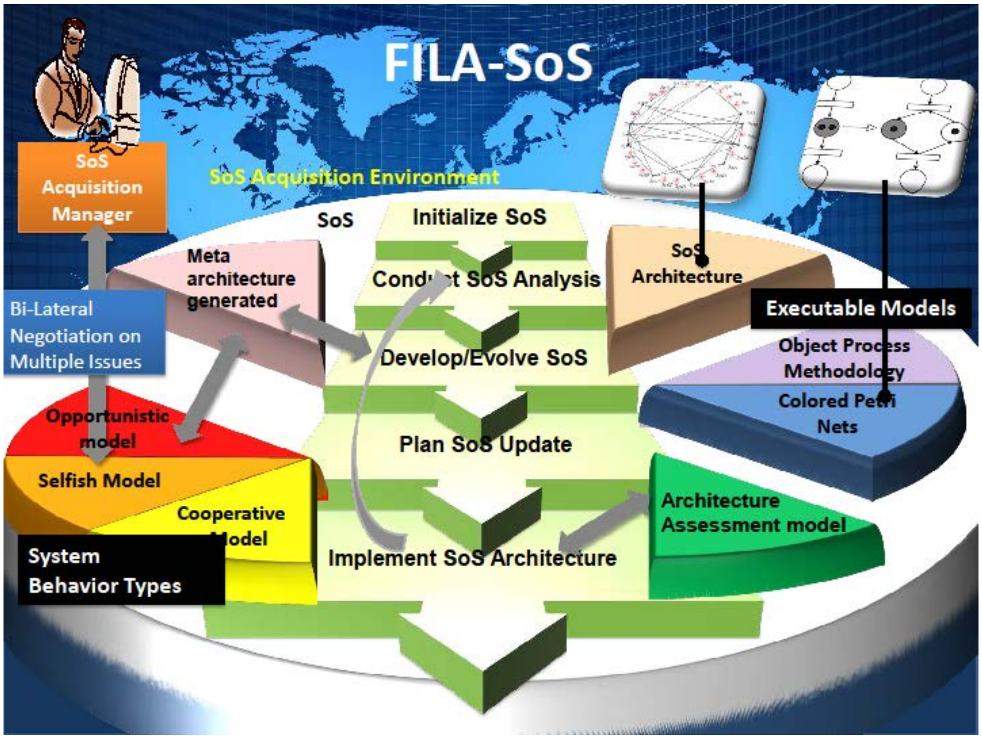


Figure 6 Integrated modules within FILA- SoS

All the independent models are listed below for reference:

- Meta-Architecture Generation Model
- Architecture Assessment Model
- SoS Negotiation Model
- System Negotiation Model: Non-Cooperative
- System Negotiation Model: Cooperative
- System Negotiation Model: Semi-Cooperative
- Executable Architecting Model: OPM & CPN
- Overall Negotiation Framework

The first meta-architecture generation method is fuzzy-genetic optimization model (Pape, Agarwal, Giammarco & Dagli, 2014). This model is based on evolutionary multi-objective optimization for SoS architecting with many key performance attributes (KPA). It also has a type-1 fuzzy assessor for dynamic assessment of domain inputs and that forms the fitness function for the genetic algorithm. It returns the best architecture (meta-architecture) consisting of systems and their interfaces. It is a generalized method with application to multiple domains such as Gulf War Intelligence/Surveillance/Reconnaissance Case and Alaskan Maritime Search and Rescue Case.

The second meta-architecture generation model is based on multi-level optimization (Konur & Dagli, 2014). In this model, architecting is done in two rounds: the first being the initiating the SoS by selecting the systems to be included in the SoS and then improving the SoS's performance by allocating funds to participating systems. The model is generic based on multiple attributes such as maximum performance, minimum cost and minimum deadline. It based on a Stackelberg game theoretical approach between the SoS architect and the individual systems.

The particle swarm optimization (Agarwal, Pape, & Dagli, 2014) technique for meta-architecture generation is similar to fuzzy-genetic model. Except for the fact that evolutionary optimization technique in this case is based on swarm intelligence. In addition, there are some new key performance attributes used to calculate the architectures quality. Cuckoo search optimization (Agarwal, Wang, & Dagli, 2014) based meta-architecture is again anew biologically inspired method of optimization. It has been shown that it in certain cases it performs better than PSO.

The first architecture assessment method is based on type-1 fuzzy logic systems (FLS) (Pape et al., 2013). The Key Performance Parameters (KPP) chosen are performance, affordability, flexibility, and robustness. It can capture the viewpoints of multiple stakeholders'. It can also accommodate any number of KPPs.

Another architecture assessment method is based on type-2 fuzzy modular nets (Agarwal, Pape & Dagli, 2014). The attributes used for evaluation were Performance, Affordability, Developmental Modularity, Net-Centricity and Operational Robustness. Type-1 fuzzy sets are able to model the ambiguity in the input and output variables. However, type-1 fuzzy sets are insufficient in characterizing the uncertainty present in the data. Type-2 fuzzy sets proposed by Zadeh (1975) can model uncertainty and minimize its effects in FLS (Mendel & John, 2002).

It is not possible to implement such meta-architecture without persuading the systems to participate, hence to address the issue a negotiation model is proposed based on game theory (Ergin, 2104). It is an incentive based negotiation model to increase participation of individual systems into Search and Rescue SoS. The model provides a strategy for SoS management to determine the appropriate amount of incentives necessary to persuade individual systems while achieving its own goal. The incentive contract is designed based on the objectives of the SoS and the individual systems. Individual system's objective is to secure highest incentives with minimal effort while the SoS manager's goal is to convince individual systems to join the SoS development while maximizing its own utility. Determining the incentives for individual systems can be formulated as a multi-constraint problem where SoS manager selects a reward for the individual system such that the reward will maximize SoS manager's expected utility while satisfying the constraints of the individual systems.

Another negotiation model based on clustering and neural networks is developed (Agarwal, Saferpour & Dagli, 2014). This model involves adapting the negotiation policy based on individual systems behavior that is not known to the SoS manager. The behavior is predicted by clustering the difference of multi-issue offers. Later the clustered data is trained using supervised learning techniques for future prediction.

Individual systems providing required capabilities can use three kinds of negotiation models based on their negotiation strategies non-cooperative Linear Optimization model, cooperative fuzzy negotiation model, and Semi-cooperative Markov chain model (Dagli et al., 2013).

Executable architectures are generated using a hybrid of Object Process Methodology (OPM) and Colored Petri Nets (CPN) (Agarwal, Wang, & Dagli, 2014), (Wang, Agarwal, & Dagli, 2014), and (Wang & Dagli, 2011). To facilitate analysis of interactions between the participating systems in achieving the overall SoS capabilities, 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. 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.

SoS ACQUISITION PROCESS

The DoD 5000.2 is currently used as the acquisition process for complex systems. Schwartz (2010) described this process as an extremely complex systemic process that cannot always constantly produce systems with expected either cost or performance potentials. The acquisition in DoD is an SoS problem that involves architecting, placement, evolution, sustainment, and discarding of systems obtained from a supplier or producer. Numerous attempts undertaken to modify and reform the acquisition process have found this problem difficult to tackle because the models have failed to keep pace with actual operational scenarios. Dombkins (1996) offered a novel approach to model complex projects as waves. He suggested that there exists a major difference in managing and modeling traditional projects versus complex projects. He further illustrated his idea through a wave-planning model that exhibits a linear trend on a time scale; on a spatial scale, it tries to capture the non-linearity and recursiveness of the processes. In general the wave model is a developmental approach that is similar to periodic waves. A period, or multiple periods, can span a strategic planning time. The instances within the periods represent the process updates. A recently proposed idea (Dahman, Lane, Rebovich, & Baldwin, 2008) that SoS architecture development for the DoD acquisition process can be anticipated to follow a wave model process. According to Dahman DoD 5000.2 may not be applicable to the SoS acquisition process. Acheson (2013) proposed that Acknowledged SoS be modeled with an Object-Oriented Systems Approach (OOSA). Acheson also proposes that for the development of SoS, the objects should be expressed in the form of a agent based model.

The environment and the systems are continuously changing. Let there be an initial environment model which represents the SoS acquisition environment. As the SoS acquisition progresses through, these variables are updated by the SoS Acquisition Manager to reflect current acquisition environment. Thus, the new environment model at a new time has different demands. To fulfill the demands of the mission a methodology is needed to assess the overall performance of the SoS in this dynamic situation. The motivation of evolution is changes in the SoS environment (Chattopadhyay, Ross, & Rhodes, 2008). The environmental changes consist of:

- SoS Stakeholder Preferences for key performance attributes
- Interoperability conditions between new and legacy systems
- Additional mission responsibilities to be accommodated
- Evolution of individual systems within the SoS
- Capabilities of individual systems

The methodology for architectural evolution in SoS should be such that it addresses all the changes in the environment stated above.

ASSESSING SYSTEMS ARCHITECTURE

In principle, systems engineering may be thought of as a decision-making activity. The architecting process involves the hierarchical reduction of ambiguity where a set of alternatives is evaluated so that the most suitable alternatives are selected. SoS design problems are based on multi- objective functions for binary variables. The design is judged based on a number of key performance parameters that together form a highly non-linear hyper surface. These techniques were employed in this study. The multi-objective approach combines multiple objectives into the following single objective:

Max $\mathbf{f}_k\ (\mathbf{x})^T\ \forall\ k$

$\mathbf{g}_i\ (\mathbf{x})^T <= b_i\quad \forall\ i$

$\mathbf{x}^T = \{\ x_1\ \ x_2\ \ \dots\ \ x_n\ \} \in \mathbf{X}$

x: vector of the variables; **f**: objective function(s); **g**: inequality constraints;

A solution to the multi-objective problem includes compromise that is acceptable to the decision maker with respect to all of the objectives pursued (Schutze, Lara, & Coello Coello, 2011).

HANDLING MANY OBJECTIVES

Multi-objective optimization algorithms are well-known and fully developed for situations with two or three objectives. Coello (1999) gives a list of references on evolutionary multiobjective optimization. Some popular and established ways to solve such problems are weighted approach (Marler & Arora, 2010), goal programming (Deb, 1999), Pareto dominance (Horn, Nafpliotis, & Goldberg, 1994), ε–Pareto Dominance Optimization is applied to workflow grid scheduling (Garg & Singh, 2011), and ranking of objectives (Garza-Fabre, Pulido, & Coello, 2009). Many objective optimization refers to conditions which more have than three objectives. Solving many objective optimization problems with the above listed methods can be difficult because nearly all solutions in a population grow into non-dominated, with increasing number of objectives. Secondly, the number of solutions required for approximation increases exponentially with the increase in dimensionality of the objective space (Schutze, Lara, & Coello Coello, 2011). As the number of objectives goes beyond five or more, the number of non-dominated solutions in a randomly generated population is more than 90% (He & Yen, 2014). The effectiveness of the recombination operators usually used in evolutionary algorithms is reduced (Deb & Jain, 2014). Besides it is hard to visualize solutions in higher dimensional spaces, weakening in search ability of Pareto dominance based algorithms and a very high computational cost (Ishibuchi, Tsukamoto, & Nojima, 2008). Stochastic heuristic techniques such as evolutionary algorithms are often used to generate solutions and fuzzy logic may be used for assessing the fitness of these solutions (Agarwal, Pape, & Dagli, 2014). These techniques were employed in this study.

Some methods to deal with many objective problems include; using reference-point-based nondominated sorting approach (Deb & Jain, 2014), Pareto corner search evolutionary algorithm and dimensionality reduction (Singh, Issacs, & Ray, 2011), objective reduction using linear and nonlinear algorithms (Saxena, Duro,Tiwari, Deb, & Zhang, 2013), designing a grid based evolutionary algorithm (Yang, Li, Liu & Zheng, 2013), fuzzy-based Pareto optimality (He & Yen, 2014), Borg multi-objective evolutionary algorithm (MOEA) proposes to combine all techniques such as ε-dominance, convergence speed measuring process called progress, random initialization, and auto-adaptive multi-operator recombination (Hadka & Reed, 2013), multiobjective optimization problem can be decomposed into a smaller number of scalar optimization subproblems and then optimize them concurrently (Zhang, & Li, 2007), many researchers are using hypervolume indicator as a quality measure of the Pareto fronts (Bader & Zitzler, 2011) and besides there exist other performance metrics to compare Pareto fronts obtained by evolutionary algorithms (Yen & He, 2014).

The key performance attributes serve as our objectives in optimization of resources required by each system participating in the SoS. Since there are many key performances attributes, hence they are incorporated in the optimization algorithm through fuzzy associative memory.

AUTOMATED NEGOTIATIONS

The importance of studying negotiation is realizable in electronic commerce, and artificial intelligence. Negotiations have two major components viz the number of parties who are negotiating and the issues on which they are negotiating. Each party negotiates in its own interest to reach at least the same or a better outcome than the previous offer made to it (An, 2011). Cooperative negotiation has found uses in maintaining real time load of a mobile cellular network (Bigham & Du, 2003, July), modeling complex physiological phenomena (Gatti, & Amigoni, 2004, July) and resolving air traffic conflicts efficiently (Wollkind, Valasek, & Ioerger, 2004, August). A negotiation can occur between two individuals, or one individual negotiating with several individuals, and finally many individuals negotiating with many other individuals. These negotiations are called bilateral (Lin, Kraus, Wilkenfeld, & Barry, 2006), one-to-many (Rahwan, Kowalczyk, & Pham, 2002) and many-to-many (Nguyen, & Jennings, 2006) respectively.

A detailed classification of automated negotiations can be accessed from Buttner (2006). Automated negotiation is an integral part of systems across all domains (Jennings et.al, 2001). Automated negotiation can be defined as an iterative process of settling on an issue or multiple issues between the negotiating parties (Fatima, Wooldridge, & Jennings, 2002).

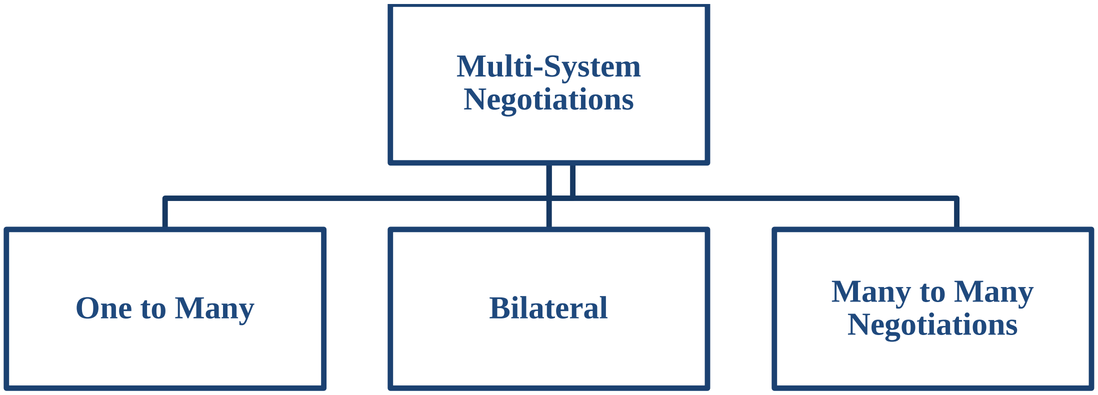


Figure 7 Automated Negotiations Protocol Categories

According to (Zheng et al., 2013; Guttman & Maes, 1998) negotiation in multi-agents is a decision process for resolving multiple issues, which may or may not be mutually exclusive. Most of the current research is focused on assigning utility functions encompassing all issues or a function for each issue and then combining the utilities to estimate the overall benefit of an offer (Ito,T., et al., 2009). This assumption is usually with the utilities making the decision a linear problem, which is usually, not the case. The utility functions can be classified into linear and nonlinear. Agents that utilize linear utility functions can aggregate the utilities of the issue-values by weighted linear summation. However, such an approach is considered naïve for modeling real world scenarios as aggregations are unrealistic. Multi-attribute utility theory (MAUT) (Dyer, 2005) believes that each outcome issue or attribute is independent. MAUT proposes to have a separate utility function for each of the issues.

Besides the systems can exhibit diverse behaviors which cannot be estimated as functions and it is hard to predict their ranking of preference for a particular issue (Marsá-Maestre et.al., 2014). Game theory postulates negotiation as a non-zero sum game along multi-dimensional issues (Binmore & Vulkan, 1999). Multiple issue negotiations can be broadly categorized as separate negotiations where each issue is dealt individually by the negotiators, in simultaneous negotiations all issues are taken up together, where in sequential negotiations, a set sequence is assigned to the total issues and each issues is then taken up in that order (Fatima, Wooldridge,& Jennings, 2006).

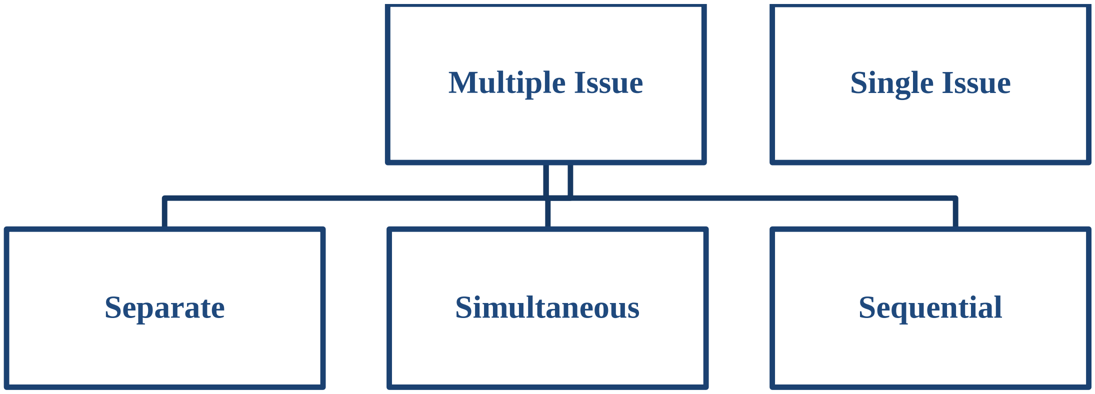


Figure 8 Categories of Attributes in Automated Negotiations

Agents are classified based on information possessed at the time of negotiation into complete or partial information states. If the agent has the complete information of the environment, which includes the opponent agent's, negotiation strategy, the external factors that affect the negotiation and the effect of the agent's strategy on the opponent it is said that that agent is in a complete information state. Otherwise, if any information is unclear or missing the agent is assumed to be in a partial information state. Information in multi agent systems are comprised of utility functions that the opponent agents use to evaluate various attributes, the reasoning models of opponent agents, and the constraints of opponent agents.

The better approach would be to calculate the opponent's behavior based on its previous offer, and then adapt the response accordingly (Chen & Weiss, 2013). Different adaptive strategies have been proposed earlier such as the ABiNeS: An Adaptive Bilateral Negotiating Strategy over Multiple Items for effectively handling different types of opponents (Hao & Leung, 2012). Other methods include game theoretic analysis (Jordan, Kiekintveld, & Wellman, 2007), use of genetic algorithms, differential evolution (Bi, & Xiao, 2012), bayesian networks (Hindriks, & Tykhonov, 2008), neural networks (Carbonneau, Kersten, & Vahidov, 2008) and fuzzy logic (Luo, et al., 2003). The sections below present the general model for Acknowledged SoS architecting.

To negotiate strategically, SoS manager needs to learn to choose efficient strategies for bargaining with other participating systems. Adaptation is a pragmatic approach towards the design of SoS coordinator agents to negotiate with systems when there is no previous knowledge of their behavior. The ability to predict the behavior of the other party requires anticipation based on previous offers. In literature, this is known as the preference sketch, which includes predicting the assumed behavior and the preference scale for all issues for the counter agent. The preference sketch of the individual systems will help SoS manager adapt to the current strategy used by the opponent. The behavioral aspect of systems is tackled through an adaptive SoS negotiation strategy.

This report aims to provide three independent models to be incorporated in version 2.0 that include, an alternative for meta-architecture generation based on swarm intelligence, a new architecture assessment technique based on type-II fuzzy logic systems, and bilateral negotiation mechanism for one SoS manager and many individual systems based on clustering and machine learning techniques. Together the three models can help in designing an overall evolution strategy for complex adaptive SoS (CASoS). The following sections include three major stages of architecture evolution:

- Meta-Architecture formulation and generation
- Meta-Architecture assessment and selection
- Meta-Architecture implementation through negotiation

META-ARCHITECTURE FORMULATION AND GENERATION

Optimization algorithms can be categorized as gradient based and non-gradient based methods. Some of the non-gradient based methods include evolutionary algorithms (Horn, Nafpliotis, & Goldberg, 1994), swarm optimization (Engelbrecht, 2006), grid search (Bergstra & Bengio, 2012) and nonlinear simplex such as Nelder-Mead (Nelder & Mead, 1965). Often gradient based methods encounter a problem when the decision variables are integer which is called a duality gap (Bertsekas, 1999). The duality gap is the difference between the primal and dual solutions which is always greater than or equal to 0. Using non-gradient based methods the duality gap reduces when solving the problem. Evolutionary algorithm based techniques have proved to be useful for solving such problems. Meta-architecture is a set of systems and interfaces selected to form a SoS based the KPAs of the problem domain. The problem of selection is posed a many-objective optimization problem. The objectives are the KPAs and the decision variables are the set of systems and interfaces. Usually in a more than one objective optimization problem there is no single optimum but a set of non-dominated solutions solving such problems with more than three objectives turns it into a many-objective optimization problem. This problem is analyzed as a Pareto-Box problem (Köppen, Vicente-Garcia, & Nickolay, 2005).

THE PARETO-BOX PROBLEM

A general approach for creating a Pareto solution can be expressed as follows:

- Let's assume there are z objective functions to be optimized.
- The decision variables are expressed as a decision vector $\vec{x} = (x_1, x_{2,} \dots, x_n)$ in the decision space X.
- A function $f: X \to Y$ evaluates a specific solution expressed as a point in objective space Y.
- Assume the objective space to be a subset of the real numbers. That is $Y \subseteq R$.
- In a single-objective optimization problem, a solution vector $x^1 \in X$ is better than $x^2 \in X$ if $f(x^1) > f(x^2)$.
- In case of a vector-valued evaluation function, the vector $g: X \to Y$ and $Y \subseteq R^k$ where $g > 1$, to compare two solutions x^1 and x^2, the Pareto dominance is applied.
- An objective vector u, where $u = g(x^1) = [f_1(x^1), f_2(x^1), \dots, f_z(x^1)]$ dominates another vector v, where $v = g(x^2) = [f_1(x^2), f_2(x^2), \dots, f_k(x^2)]$ is expressed as $u \succ v$ if and only if $\forall i \in \{1, .., z\}, u_i \geq v_i, \wedge \exists i \in \{1, .., z\}: u_i > v_i$. This is in a maximization problem. In a

minimization problem the signs of all the objective functions can be reversed and solved as a maximization problem.

- Accordingly a solution x^1 dominates x^2 $(x^1 \succ x^2)$ if $g(x^1) \succ g(x^2)$.
- The optimal solution in decision space can be expressed as $x^* \subseteq X$. Its image in objective space is $g^* \subseteq Z$.

The Pareto set X_E contains all optimal solutions also denoted efficient solutions. The Pareto front also denoted non-dominated frontier is the image of the Pareto set in objective space. The *Pareto Box problem* is explained further.

Given are x uniformly randomly selected y-dimensional points in the y-dimensional unit hypercube. If $e_x(y)$ denotes the expectation value for the size of the Pareto set of x randomly selected points in the $y - dimensional$ unit hypercube. Then, the following definitions hold (Köppen, Vicente-Garcia, & Nickolay, 2005):

Theorem 1. Given are x randomly selected points in the y-dimensional hypercube. For the expectation value of the size of the Pareto set of these x points we have the recursive relation:

$$e_{x-1}(y) = e_x(y) + \frac{1}{x} e_x(y-1) \ (x, y \geq 2) \qquad (3.1)$$

which implies,

$$e_1(y) = 1 \qquad (3.2)$$
$$e_x(1) = 1 \qquad (3.3)$$

Theorem 2. The expectation value for the size of the Pareto set of $x \geq 1$ randomly selected points in the $y \geq 1$-dimensional hypercube is

$$e_x(y) = \sum_{v=1}^{x} \frac{-1^{v+1}}{v^{y-1}} \binom{x}{v} \ \forall\, v \in \boldsymbol{V} = \{1,2, \dots.,m\} \qquad (3.4)$$

Theorem 1 and 2 will help prove the central theorem 3 relating to limiting nature of the expectation values when there is an increase in number of sample points and increase in dimensions. For proofs of theorem 1 and 2 please refer to appendix of the paper (Köppen, Vicente-Garcia, & Nickolay, 2005).

Theorem 3. For fixed dimension $y > 1$ and the number of points $x \to \infty$ the expectation value $e_x(y) \to \infty$, the ratio of the non-dominated points ${}^{e_x(y)}/_{x} \to 0$ and for fixed $x > 1$ and dimension $y \to \infty$ the $e_x(y) \to x$

$$e_x(2) = \sum_{v=1}^{x} \frac{1}{\mathrm{v}} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{m} \qquad (3.5)$$

Equation (3.5) is a harmonic series and has been proved divergent. Since the series is divergent meaning forever increasing it can be deduced from eq. (3.4) that for $n > 2$ the following condition will always remain true i.e. $e_x(y) \geq e_x(y-1) .. \geq e_x(2)$. Hence, as $x \to \infty$ the expectation value $e_x(y) \to \infty$. Besides as $x \to \infty$ and taking limits over the expression,

${e_x(y)}/{x} \to 0$. Similarly for the second part of the theorem, if x is fixed and $x > 1$ all terms in eq. (3.4) tend to zero as $y \to \infty$ except when $v = 1$. Because when $v = 1$, then since $1^{\infty} = 1$ the total term equals *x* or $e_x(y) \to x$.

As the dimensionality of the solution space increases, the probability of finding any dominated solution will fall exponentially. This means that the Pareto set of x points will contain nearly all x points. This can also be expressed as for increasing number of sample points in the solution space, the number of non-dominated points will increase as well.

In a SoS architecting problem, component systems have multiple intra and inter system trade-offs that cannot be fitted into the mold of a single objective. Secondly, the number of solutions required for approximation increases exponentially with the dimensionality of the objective space (Shutze, Lara, & Coello, 2011). The SoS architect's aim is to maximize or minimize all the objective functionsKP_r, as the case may be. The SoS optimization problem can be formulated as follows:

Optimize $\boldsymbol{F} = \left\{f_{KP_1}(\boldsymbol{s}, \boldsymbol{IF}), \dots, f_{KP_r}(\boldsymbol{s}, \boldsymbol{IF}),\ \dots f_{KP_Z}(\boldsymbol{s}, \boldsymbol{IF})\ \right\}\ \forall\, r = \{1, 2, \dots, Z\}$

where $f_{KP_r}(\boldsymbol{s}, \boldsymbol{IF})$ is the value of the key performance attribute r for decision variables $\boldsymbol{s}$ and $\boldsymbol{IF}$.

Subject to

$$\sum_i s_i\, a_{ij} \geq 1 \qquad \forall\, j \in \boldsymbol{J} \qquad (3.1)$$

$$IF_{ik} = \{1\} \leftrightarrow \{\, s_i = 1\ \wedge s_k = 1\} \quad \forall\, i, k \in \boldsymbol{I} \qquad (3.2)$$

$$a_{ij} \in \{0,1\} \qquad \forall\, i \in \boldsymbol{I} \qquad (3.3)$$

$$s_i \in \{0,1\} \qquad \forall\, i \in \boldsymbol{I} \qquad (3.4)$$

$$IF_{ik} \in \{0,1\} \qquad \forall\, i, k \in \boldsymbol{I} \qquad (3.5)$$

This is a Z dimensional muti-objective optimization problem. Constraint (3.1) guarantees that at least one system for each capability is selected. Constraint (3.2) makes sure that an interface between two systems selected if and only if the two systems are selected in the meta-architecture. Constraints (3.3), (3.4), and (3.5) give the binary decision variables.

Similar problem has been solved earlier as a multi-level bi-objective optimization (Konur & Dagli, 2014) using gradient based methods. The bi-objective model cannot handle many objectives of the general model described. There are two basic issues that need to be addressed here, namely ambiguity in the definition of the KPA, number of objectives and NP completeness of the mathematical model formulated. In this research evolutionary algorithms (EA) that use non-gradient descent optimization procedures are selected to deal with the NP completeness issues, fuzzy logic is used to represent the ambiguity in KPA and fuzzy inference is used to accommodate many objectives in formulating the fitness function. Fuzzy logic also helps in helping in the search ability of EA since search ability decreases with increasing objectives (Ishibuchi, Tsukamoto, & Nojima, 2008). Hence the above model is converted to a form where any EA can be used. Each individual chromosome is coded as a finite length vector of variables. The possible values of the variables denote the size of the alphabet. In this case the size of the alphabet is two because

$s_i\ and\ IF_{ik}$ are the binary decision variables. The details of the steps of chromosome representation are as follows.

Chromosome Representation: The chromosome is made up of two parts combined together to form a long string. The length of the individual chromosome is $L_{ch} = L_s + L_{if}$. L_{ch} is the length of the chromosome, $L_s$is the first part made by vector $\boldsymbol{s}$. The second part or L_{if} is made by linearizing the matrix $\boldsymbol{IF}$ as shown in Table 2, 3, and 4.

Table 2 A solution in the form of a string containing systems

s_1	s_2	s_i	…	s_N
Systems L_s=N				

Table 3 A solution in the form of a string containing interfaces

IF 1 with 2	IF 1 with 3	IF 1 with N	IF 2 with 3	…	IF i with k	…	IF (N-1) with N
Interfaces $L_{Inf} = N * (N-1)\ /\ 2$							

Table 4 A solution in the form of a string containing both systems and interfaces

s_1	…	s_i	…	s_N	IF i with k	…	IF (N-1) with N
Systems and Interfaces $L_s + L_{Inf} = N + N * (N-1)\ /\ 2$							

With N participating systems the total number of variables become$(N + N * (N-1)\ /\ 2)$. The solution string is binary in nature wherein a one represents the presence and a zero means the absence of a system or interface. This representation can be used to solve this problem with evolutionary algorithms, evolutionary strategies (Beyer & Schwefel, 2002), swarm optimization or differential evolution (Storn & Price, 1997). The general outline of EA consists of these steps (Back & Schwefel, 1996):

```
“ $t = 0;$

$Initialization\ \ P(0) = \{\overline{a}_1(0), \dots., \overline{a}_\mu(0)\}\,, \in \boldsymbol{I^\mu}$

$Evaluation\ \ P(0) = \{\boldsymbol{\phi}(\overline{a}_1(0)), \dots., \boldsymbol{\phi}(\overline{a}_\mu(0))\};$

$While\ (termination\ condition\ for\ P(t) \neq true)\ do$

            Recombination  $P'(t) = r\boldsymbol{\theta_r}(P(t));$

            Mutation  $P''(t) = m\boldsymbol{\theta_m}(P'(t));$

$Evaluation\ \ P''(t) = \{\boldsymbol{\phi}(\overline{a}''_1(0)), \dots., \boldsymbol{\phi}(\overline{a}''_\mu(0))\};$

Selection $of\ indivdulas\ P(t+1) = s\boldsymbol{\theta_s}(P''^{(t)} \cup Q);$
```

$t = t + 1$; End do;”

Initially the generations are set to be zero. Then an initial population $P(0)$ of size μ is created with individuals represented by $\overline{a}$. The solutions or individuals are referred to as the chromosomes. Each individual in the population is evaluated by an objective function $\boldsymbol{\phi}$ to calculate the fitness value. Each of the consequent generations is created iteratively by applying operations, on the current population, that include recombination operator $r\boldsymbol{\theta_r}$, and mutation operator $m\boldsymbol{\theta_m}$. This process is run until the termination criterion is met and the algorithm stops creating new generations.

The new individuals in the next generation have a new size γ. The new population $P''(t)$ is evaluated using the objective function $\boldsymbol{\phi}$. The selection process $s\boldsymbol{\theta_s}$ selects some individual of size μ to create the population for the next generation where $t = t + 1$.”

With respect to the problem at hand the decision variables are $\boldsymbol{s}$ and $\boldsymbol{IF}$. Recall that $s_i\ and\ IF_{ik}$ are the binary decision variables in SoS. *Chromosome Initialization* will involve generating random binary values in all bits to start the population. *Fitness assessment* for a meta-architecture is explained in the following section where this population is evaluated for Z objectives. *Termination criteria* should be such that algorithm should not converge prematurely. Whereas the termination was based on a minimum number of generations until the best solution quality does not change. Other techniques for termination include a hitting a bound on the threshold quality of solution.

The process includes producing a meta-architecture using multi-objective evolutionary algorithms. Multiple objective decisions making (MODM) increases in difficulty with growing number of objectives (Key performance parameters). The probability of finding dominated solutions based on ten or more objectives is very low. To solve this problem the architectures assessment technique uses a fuzzy type II modular rule base approach (fuzzy networks) that allows multiple key performance parameters to be evaluated at the same time. The fuzzy rule base defines the preference of the decision maker in our case the Acknowledged SoS manager (Pape, 2013).

META-ARCHITECTURE ASSESSMENT AND SELECTION

In the previous section a methodology for generating the solution was explained. Now to determine the quality of the solution (SoS architecture) a technique is needed to assess it. The technique should be generic enough to be applied to many independent domains. For this the objective function is converted to fitness functions for population based algorithms. Architecture assessment is based on KPAs which are selected based on the domain of the problem. Multiple objectives produce a non-linear hypersurface. The optimization algorithm has to trace the surface to find the global minima or maxima. This process is very computationally expensive and tedious. Fuzzy associate memories can be used as a way combining multiple objectives in to one non-linear surface with many dimensions (Agarwal, Pape, & Dagli, 2014).

The first problem is dealing with ambiguity in calculating the values of various objectives. This situation is dealt by using type-1 fuzzy systems.

Secondly a method is needed to manage the preferences between KPAs in the fitness function. A tradeoff exists between the KPAs. This tradeoff is often non-linear and depends on a number of stakeholders of the architecture. Usually the tradeoffs are aggregated linearly through utility functions. For example if two KPA's are scalability and reliability. The tradeoff could be higher reliability and low scalability. Besides the tradeoffs depend on a group of stakeholders which include system architect, project manager, customers and so on. Some methods such as fuzzy Pareto dominance (He & Yen, 2014), ranking of alternatives (Wang & Yang, 2009), fuzzy goal programming (Hu, Teng, & Li, 2007), weighting the objectives (Marler & Arora, 2010) have been used previously to combine them in to a single objective. Fuzzy associative memory helps capture the non-linearity that exists between the KPAs and can accommodate the view of multiple decision makers at the same time.

The third key factor is that the assessment techniques should be able to bring in performance attributes requirements from a lower level of abstraction. Often there is a difficulty in assigning actual numerical values to the KPA because the needs and requirements are expressed as words by the stakeholders. For example an attribute such as *net-centricity* can be broken down into *interoperability* and *command & control communication support* capability. Some of the prominent methods to assess the architectures include the use case maps (UCM) (Folmer, van Gurp, & Bosch, 2003), Architecture Tradeoff Analysis Method (ATAM) (Kazman et. al, 1998), and Scenario based Architecture Analysis Method (SAAM) (Kazman, Abowd, Bass, & Clements, 1996). There have been comparisons of architecture evaluation methods to choose the correction option effectively (Babar, Zhu, & Jeffery, 2004)

A beneficial approach would be to not only capture the tradeoffs points between as many possible KPAs in a nonlinear fashion, be able to compute with words, incorporate multiple views from stakeholders and help in value aggregation from different levels of abstraction of each KPA.

None of the methods discussed above are able to address the issues described above. The domain independent method proposed here for a domain dependent architecture value aims to fill this gap in literature. The proposed assessment model is based type-II fuzzy inference engine. The values provide more realistic assessment of the SoS architecture's quality. The attributes will be domain adjusted and selectable, using guidance from subject matter experts.

As the reader may recall the architecture is described as a chromosome. The fuzzy assessor based assessment is used to evaluate the fitness of the chromosome during the meta-architecture generation process. This assessor can be also used to evaluate the architecture after the negotiation. The concepts of fuzzy logic systems (FLS) are explained below to understand the working of the assessor.

Crisp sets are those where an element is either a member of the set or not. Fuzzy logic (Zadeh, 1965) is an approach where a membership of the elements of a set is not true or false but is based on degrees of truth. A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 (not an element of the set) and 1(a member of the set). The input space is sometimes referred to as the universe of discourse.

Example 1

Continuous Example: Let U be the interval [0,100] representing the realiability of a system-of-systems. Then we may define fuzzy sets "Poor" and "Excellent" as membership functions.

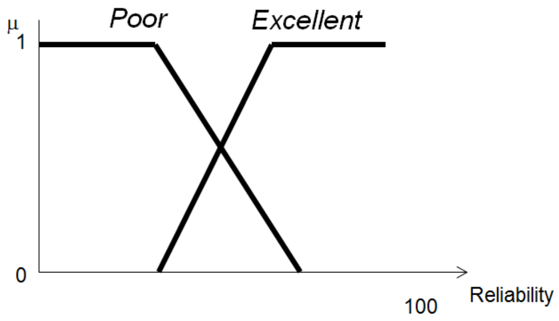


Figure 9 Membership functions ""Poor" and "Excellent" for fuzzy variable Reliability

TYPE-I FUZZY LOGIC SYSTEM

Type-1 fuzzy set (T1 FS) theory was originally introduced by Zadeh (1965). Some of the applications include control theory (Tzafestas, 1994), artificial intelligence (Hüllermeier,2005), and forecasting (Song, & Chissom, 1993). A typical Type-1 FLS has a fuzzifier, a rule section, fuzzy inference engine (FIS) and a defuzzifier or output processor. Figure 10 depicts the illustration of a type- 1FLS.

Fuzzy sets can be described as points in the unit hypercube $I^n = [0,1]^n$ (Kosko, 1992). A crisp value lies on the corner of the unit hypercube. A fuzzy system is a transformation S: I^n→ I^m that maps fuzzy sets in I^n to fuzzy sets in I^m. These continuous fuzzy systems behave as associative memories. A fuzzy associative memory (FAM) contains a matrix of fuzzy values which can map an input fuzzy set into an output fuzzy set followed by an appropriate superimposition operator (Chung & lee, 1996). The rules are able to express a non-linear relationship between the variables. The process is explained through a simple example.

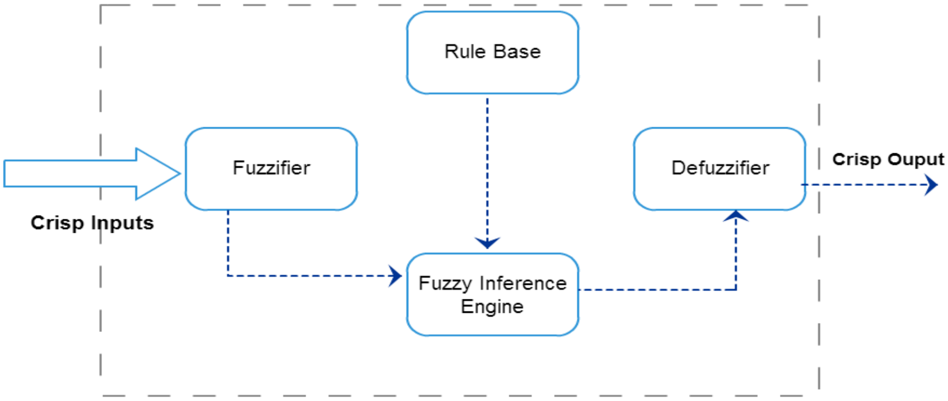


Figure 10 Overview of type-1 FLS

Example 2

The problem is to calculate the *architecture quality* of a system. For the sake of ease two inputs, *reliability* and *cost* are considered. The linguistic values for *reliability* are ‘*low’*, ‘*medium’* and ‘*high’*. The linguistic values for *cost* are ‘*cheap* and ‘*expensive’*. The choice of membership function is up to the user based on the domain of the problem, experience and computational difficulty. The membership function for *reliability* and *cost* in the universe of discourse, U, is given below

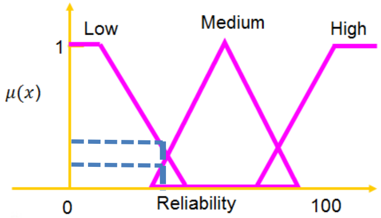


Figure 11 The membership functions for *reliability*

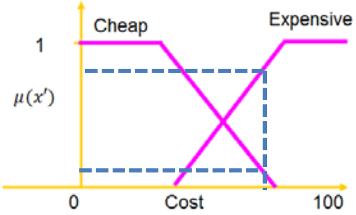


Figure 12 The membership functions for *cost*

The linguistic values for *architecture quality* are ‘*risky*, ‘*modest’*, and ‘*excellent’*. The membership function for *architecture quality* in the universe of discourse, U, is given below

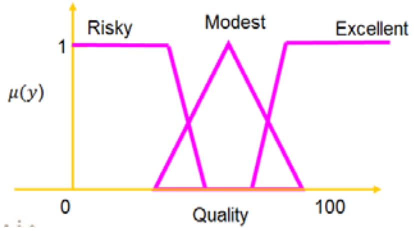


Figure 13 Overview of type-1 FLS

Step 1

The first process involves converting the crisp inputs into fuzzy sets. This is called the fuzzification process. The inputs are reliability = 35 and cost =80. The fuzzy values for these crisp values by using the membership functions of reliability as shown in the figure by dotted lines are:

$$\mu_{\text{reliability=low}}(35) = 0.3$$
$$\mu_{\text{reliability=medium}}(35) = 0.2$$
$$\mu_{\text{reliability=high}}(35) = 0$$

The fuzzy values for crisp values of cost are obtained by membership functions of cost in Figure 12 as

$$\mu_{\text{cost=cheap}}(80) = 0.1$$
$$\mu_{\text{cost=expensive}}(80) = 0.8$$

Step 2

After obtaining the fuzzy values from crisp inputs rules are needed to arrive at the final fuzzy output value. This is called the rules evaluation process. The rules for this problem are as follows:

"If the reliability is *low* or cost is *expensive*, then the quality is *risky*."

"If the reliability is *medium* and cost is *cheap*, then the quality is *modest*."

"If the reliability is *high* or cost is *cheap*, then the quality is *excellent*."

Definitions 5 and 6 are used in the rules containing disjunctions, OR and AND using the max and min operator. Each rule is evaluated below for explanation of the concept:

Rule 1

$$\mu_{quality=risky}(y) = \max[\mu_{low}(35), \mu_{expensive}(80)]$$
$$\mu_{quality=risky}(y) = \max[0.3,0.8] = 0.8$$

Rule 2

$$\mu_{quality=modest}(y) = \min[\mu_{medium}(35), \mu_{cheap}(80)]$$
$$\mu_{quality=modest}(y) = \max[0.1,0.2] = 0.1$$

Rule 3

$$\mu_{quality=excellent}(y) = \max[\mu_{high}(35), \mu_{cheap}(80)]$$
$$\mu_{quality=excellent}(y) = \max[0,0.1] = 0.1$$

To get the fuzzy values of the outputs, the FLS has to use fuzzy inference engine. Mamdani (1977) presented a method to synthesize the rules in fuzzy logic control. The Mamdani operator can be expressed as:

$$\varphi\left(\mu_A(x), \mu_B(y)\right) = \mu_A(x) AND\ \mu_B(y)$$
$$\min\left[0.8, \mu_{quality=risky}(y)\right]$$

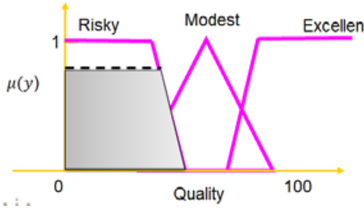


Figure 14 Overview of type-1 FLS

$$\min\left[0.1, \mu_{quality=modest}(y)\right]$$

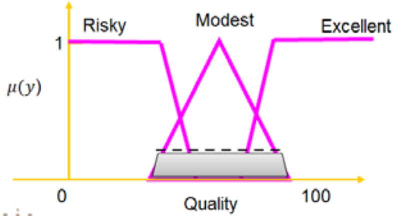


Figure 15 Overview of type-1 FLS

$$\min[0.1, \mu_{quality=excelelnt}(y)]$$

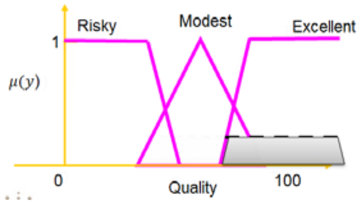


Figure 16 Overview of type-1 FLS

To defuzzify the outputs we use the center of gravity method. This process is called the defuzzification. The center of gravity of the areas defined by the rules is the final defuzzified answer. There are many other methods such as BOA (bisector of area), CDD (constraint decision defuzzification), COA (center of area) and so on. In center of gravity method we take the output from each contributing rule, and then we add them. The centroid of the region is calculated as

$$COG = \frac{\sum_{x=a}^{b} \mu_A(x) * x}{\mu_A(x)}$$

The calculation for COG is shown as follows:

$$\frac{(0+10+20)*0.8+(30+40+50+60)*0.2+(70+80+90+100)*0.5}{0.8*3+0.1*4+0.1*4} = 71.8$$

It means there is 71.8 % of chance of systems quality.

In relation to this model architecture evaluation methods have been developed (Pape & Dagli, 2013) to assess robustness of SoS architectures. Also type-1 fuzzy associative memory has been developed to evaluate SoS architectures (Pape et al., 2013). The attributes used for evaluation were Performance, Affordability, Developmental Flexibility, and Operational Robustness. Type-1 fuzzy sets are able to model the ambiguity in the input and output variables. But type-1 fuzzy sets are insufficient in characterizing the uncertainty present in the data. Type-2 fuzzy sets proposed by Zadeh can model uncertainty and minimize its effects in FLS (Mendel & John, 2002). The next section gives a brief overview of type-2 and interval type-2 fuzzy sets.

TYPE-2 FUZZY SETS

In this research, many stakeholders provide options for numerous KPAs. Type-II fuzzy sets are able to capture the uncertainties in these multiple opinions. Also later on if we can calculate the uncertainty we can use Non singleton type II fuzzy sets (NS-IT2FS) to extend my approach, where there is an inherent uncertainty incorporated in the inputs as well.

The cause of uncertainties in type-1 FLS includes the following:

- Different people might interpret different meanings to the same words being used in antecedent and consequent rules
- There is often uncertainty present in the input data which is not a single crisp value but has a given distribution if a group of decision makers are involved
- Similarly the outputs may not have a singleton value but a distribution over which the outputs range due to multiple experts

These gaps are not addressed by type-1 fuzzy because their membership functions are totally crisp. Whereas, type-2 fuzzy sets are able to model such uncertainties due to the fact that their membership function are fuzzy themselves and are three-dimensional in nature. The structure of rules in a type-1 FLS and a type-2 FLS is the same, but in type-II the antecedents and the consequents are represented by type-2 fuzzy sets. A type-2 FLS contains a fuzzifier, a rule base, a fuzzy inference engine, and an output processor. The output processor includes type-reducer and defuzzifier. The type reducer reduces the type-2 FS to a type-1 FS whereas the defuzzifier converts the type-1 FS to a crisp number. The structure of the type-2 fuzzy associative memory maps inputs to type-2 fuzzy terms. Rules are made to describe the relationship between inputs and output using the linguistic terms of each input's membership functions. Type-2 FLSs are computationally demanding because of type-reduction. Interval type-2 (IT2) FSs (Liang & Mendel, 2000) are a special case of type-2 FSs extensively used for their less computational cost. IT2 FSs are often useful when there is an uncertainty involved in determining the exact membership functions, or when there are multiple stakeholders' opinions on the same fuzzy variable (Wu, 2013). A general procedure for IT2FS is illustrated in the Figure 17. It is similar to type-1 FS, except fuzzifier converts the crisp inputs to IT2 FS, the outputs of the inference engine are IT2 FSs, there is another element called the type-reducer which converts the IT2FS values to type-1 FS before passing them to the defuzzifier.

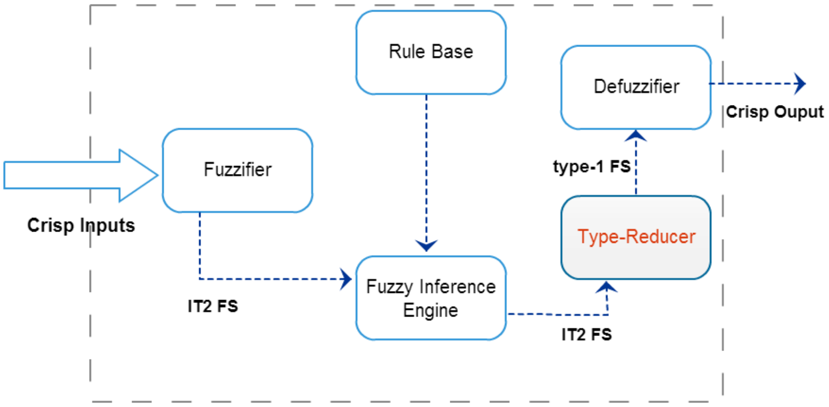


Figure 17 Overview of type-2 FLS

An example of an IT2 FS, $\tilde{Y}$, is shown in Figure 18. A type-2 FS has two membership functions hence for each value of the linguistic variable the membership degree is not a number but an interval. This is because a straight line parallel to membership axis will cut the membership functions at two places. One of them will be lower forming the lower interval and the other one will form the higher interval of the degree. The two membership functions are denoted by $\bar{Y}$ (upper MF) and $\underline{Y}$ (lower MF). The area between them is the footprint of uncertainty (FOU).

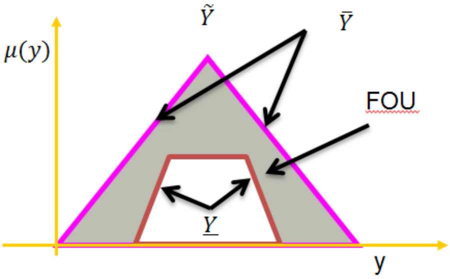


Figure 18 Membership function for a type-2 FLS

Given $\tilde{Y}_1^n$ are IT2 FSs antecedents or inputs, and $Z^n = [\underline{z}^n, \bar{z}^n]$ interval of a consequent output where $n = 1,2, \dots, N$ and $k = 1,2, \dots, K$

The steps in an IT2 FLS are demonstrated as follows:

- Consider the rule base of an IT2 FLS comprising of N rules assuming that the nth rule is :
  - IF y_1 is $\tilde{Y}_1^n$ and….. and y_K is $\tilde{Y}_K^n$, THEN z is Z^n
- Calculate the membership of all inputs in the vector $\boldsymbol{y}' = (y'_1, y'_2, \dots y'_K)$ on each $\tilde{Y}_1^n$ for $n = 1,2, \dots, N$ and $k = 1,2, \dots, K$
  - Membership is $[\mu_{\underline{Y}_k^n}(y'_k), \mu_{\bar{Y}_k^n}(y'_k)]$
- When the nth rule $H^n(\boldsymbol{y}')$ for the input vector, fires the output interval computed as:

$$[\underline{h}^n, \bar{h}^n] = [\mu_{\underline{Y}_1^n}(y'_1) \times \dots \times \mu_{\underline{Y}_K^n}(y'_K), \mu_{\bar{Y}_1^n}(y'_1) \times \dots . \times \mu_{\bar{Y}_K^n}(y'_K)]$$

$$Z_{Cos}(y') = \bigcup_{\substack{h^n \in H^n(\boldsymbol{y}') \\ z^n \in Z^n}} \frac{\sum_{n=1}^{N} h^n z^n}{\sum_{n=1}^{N} h^n} = [z_l, z_r]$$

The lower z_l and upper limits z_r of the outputs can be calculated as follows.

$$z_l = \begin{matrix} min \\ x \in [1, N-1] \end{matrix} \frac{\sum_{n=1}^{x} \bar{h}^n \underline{z}^n + \sum_{n=x+1}^{N} \underline{z}^n \underline{h}^n}{\sum_{n=1}^{x} \bar{h}^n + \sum_{n=x+1}^{N} \underline{h}^n} \equiv \frac{\sum_{n=1}^{L} \bar{h}^n \underline{z}^n + \sum_{n=L+1}^{N} \underline{z}^n \underline{h}^n}{\sum_{n=1}^{L} \bar{h}^n + \sum_{n=L+1}^{N} \underline{h}^n}$$

$$z_r = \begin{matrix} max \\ x \in [1, N-1] \end{matrix} \frac{\sum_{n=1}^{x} \underline{h}^n \bar{z}^n + \sum_{n=x+1}^{N} \bar{z}^n \bar{h}^n}{\sum_{n=1}^{x} \underline{h}^n + \sum_{n=x+1}^{N} \bar{h}^n} \equiv \frac{\sum_{n=1}^{R} \underline{h}^n \bar{z}^n + \sum_{n=R+1}^{N} \bar{z}^n \bar{h}^n}{\sum_{n=1}^{R} \underline{h}^n + \sum_{n=R+1}^{N} \bar{h}^n}$$

The KPA properties include:

- Range of Values of KPA for evaluating SoS capability C can be provided with different levels of linguistic granularization as shown in the example above

- Depending on the problem the type of member ship function is required the represent the ambiguity in each KPA.
- The crisp value of each KPA is hard to determine. Hence they are aggregated using the parts that account for each KPA. For example it is difficult to find an absolute value of net-centricity of a SoS. Since it can be viewed as a composition of interoperability and communication with ground control system, both these values are computed and aggregated using type-1 fuzzy inference. Type-1 is used since there is less ambiguity while calculating the value for each KPA.
- Later all KPAs are aggregated using type-II inference since there is more inherent ambiguity amongst them that can be taken into account.
- This way the crisp values are first fuzzified and fed into fuzzy inference system for type-1. This is later defuzzified to obtain values for each KPA. This is fain fuzzified using type-2 inference and later defuzzified to obtain SoS architecture quality.

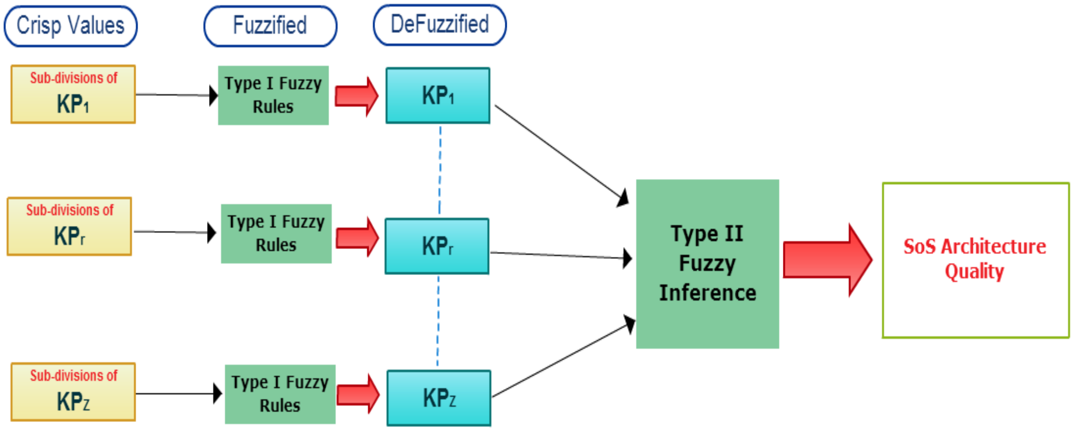


Figure 19 General Structure of Architecture Assessment Function

Based on the assessment scheme of the architecture a compromised solution is selected. The implementation of a meta-architecture through a negotiation process is explained in the next section.

SoS Negotiation Approach

The Acknowledged SoS manager negotiates with systems that are selected as part of the meta-architecture during the meta-architecture generation process. A negotiation procedure is necessary for the actualization or implementation of the meta-architecture generated. Since the SoS manager cannot his force his demands on participating systems, negotiation helps in achieving an architecture that is physically feasible. The SoS manager negotiation mechanism consist of three phases of

- Modeling the opponent
- Making a decision based on the previous offer
- Finally generating a counteroffer

A bilateral counteroffer based negotiation mechanism is chosen between an SoS manager and an individual system under multiple attributes. The attributes or issues are assumed to be independent of each other and are bargained simultaneously. Modeling the opponent involves characterizing the opponent's negotiation behavior which might be cooperative, semi-cooperative or non-cooperative. A decision mechanism is needed to reject the offer for no further negotiation, or accept the offer as it is currently or negotiate for another round to bargain further. In case of further negotiation rounds a counter –offer generation mechanism is needed. Counter offers in automated negotiation are classified on the bases of constraints used to bargain such as time taken to negotiate, value of the overall utility achieved by a party over a set of issues, or constraints based on available resources. The next section gives an overview of the negotiation mechanism and variables used to explain the problem, describes the strategy to model the opponent, illustrates the strategy for making a decision on the negotiation offer of the opponent, and finally several utility based concession curves are proposed for the SoS manager to make counteroffer.

General Negotiation Protocol

In this section the variables used in the describing the protocol are listed for the user. The negotiation strategy is designed for a one to many participants and is not mediated by any coordinator. The structure consists of a SoS manager and multiple systems selected as part of the solution in the meta-architecture. Let us define:V_p : *p= {1, 2,…, P}*: Attributes for bilateral negotiation; t_{max}: Total round of negotiations possible; $t = \{0,1, \dots t_{max}\}$; $V_p^{SoS}(t)$: The value of the attribute V_p for SoS manager at time t; $V_p^S(t)$: The value of the attribute V_p for system owner at time t

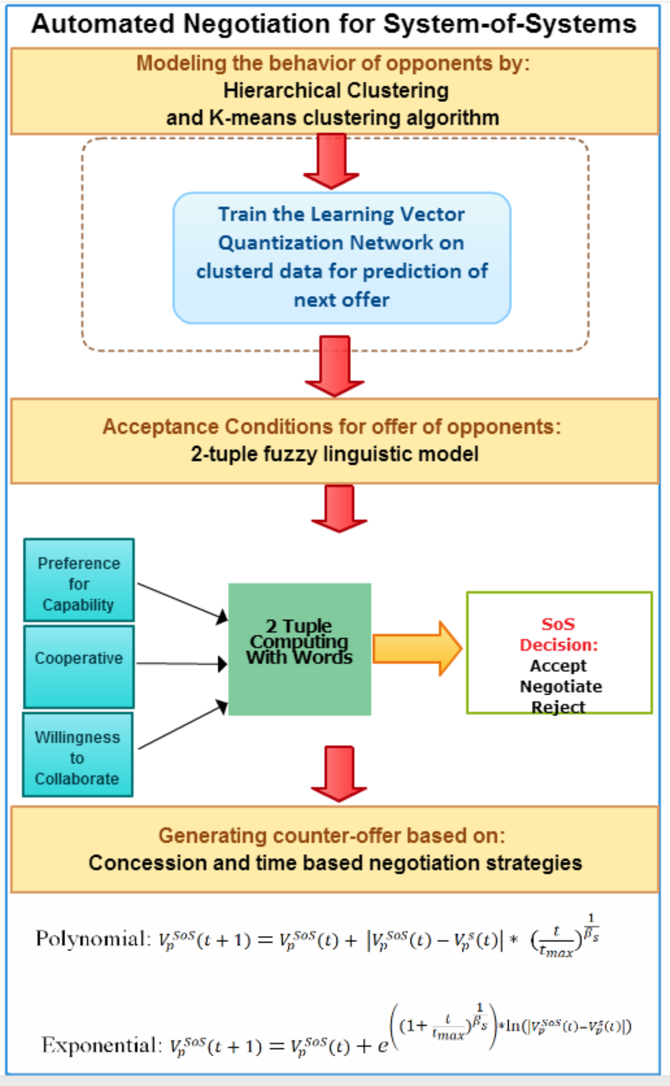

Automated Negotiation for System-of-Systems
Modeling the behavior of opponents by:
Hierarchical Clustering
and K-means clustering algorithm
Train the Learning Vector Quantization Network on clusterd data for prediction of next offer
Acceptance Conditions for offer of opponents:
2-tuple fuzzy linguistic model
Preference for Capability
Cooperative
Willingness to Collaborate
2 Tuple Computing With Words
SoS Decision:
Accept
Negotiate
Reject
Generating counter-offer based on:
Concession and time based negotiation strategies
Polynomial: $V_p^{SoS}(t+1) = V_p^{SoS}(t) + \left|V_p^{SoS}(t) - V_p^{s}(t)\right| * \left(\frac{t}{t_{max}}\right)^{\frac{1}{\beta_s}}$
Exponential: $V_p^{SoS}(t+1) = V_p^{SoS}(t) + e^{\left((1+\frac{t}{t_{max}})^{\frac{1}{\beta_s}}\right)*\ln(|V_p^{SoS}(t)-V_p^{s}(t)|)}$

Figure 20 Three Salient Features of Automated Negotiation

MODELING THE OPPONENT

The SoS coordinator records both the offer and the counteroffer for each system. It calculates the amount of concession in each issue for each system. Concessions in all issues are calculated for each system i (see Table 2). After recording this data it is used for clustering which can reveal any behavioral groupings in counter-offers. For example, a cooperative system would agree to work for less money than a non-cooperating system would. Similarly, a non-cooperating system would ask for more money in lieu of time taken to prepare for participation. The clustering is done in multi-dimensional space of the number of negotiation attributes P. The following notation describes the clustering operation:

o_g: *g* ∈ ***G***, ***G***= *{1, 2,…, NoB}*: - the number of observations made

P - the number of issues or attributes of negotiation present

L- the number of clusters the user either predicted or defined

C_h - the h^{th} cluster, a subset of h $= \{1, 2, \ldots, L\}$

Table 2 Concession calculated by SoS manager for each system

System j	NA_1	NA_2	NA_p
	$\Delta_1 = NA_{1\text{SoS}}$- $NA_{1\text{S}}$	$\Delta_2 = NA_{2\text{SoS}}$- $NA_{2\text{S}}$	$\Delta_\text{p} = NA_{p\text{SoS}}$- $NA_{p\text{S}}$

HIERARCHICAL CLUSTERING

Hierarchical clustering is a type of agglomerative clustering (Freeman, 1994). It builds a hierarchy of clusters such that clusters at one level are combined as clusters at the next level. It does not require the number of clusters in advance to proceed with. This process creates a cluster tree which is known as the *dendogram*. Hierarchical clustering algorithms require very little a priori knowledge of the data and are a non-parametric method of auto-classification (Johnson, 1967). Multi-level clustering assists the user in deciding at how many clusters are appropriate for his problem. It is often used as precursor to many other clustering algorithms to give an overview of how many clusters might be present in the data. The basic methodology of this clustering method is explained as follows:

1. Given *N* data points are to be clustered.
2. Assign a cluster based on each data appoint, which results in *N* clusters
3. A similarity metric (distance) is chosen to quantify the separation between the clusters. Similarity metric parameter defines how the distance between clusters is calculated.

Some common options are:

a. *Average Linkage*: The distance between any two clusters is estimated as the average of the distances between all the points in those clusters.
b. *Complete Linkage*: The distance between any two clusters is the distance between the farthest points in those clusters.
c. *Single Linkage:* The distance between two clusters is the shortest distance between any member of one cluster to anyone in the other cluster.

4. Calculate all pair-wise distances between clusters making a $N\ X\ N$ matrix
5. The most similar pair of clusters is merged into a single cluster and then all distances from this new cluster to all other clusters are evaluated to update the matrix.
6. In each iteration two closest data points are merged until there is a single large cluster containing all the original data points.

The decision maker can choose an appropriate level by looking at the dendogram and hence arrive at the number of clusters that can be used as input for the clustering algorithms. Clustering through k-means is explained in the next section.

K-means Clustering Algorithm

K-means clustering is one of the many unsupervised learning techniques (Grira, Crucianu, & Boujemaa, 2004) currently used to mine the underlying features of a dataset. Some of the popular techniques include partition around mediods (Kaufman, & Rousseeuw, 1990) where the major difference between k-means is that the algorithm uses mediods instead of centroids and the cluster centers may or may not be necessarily one of the data points, Fuzzy c-means (Pal & Bezdek, 1995) is based on k-means and on the concept that each data point has degree of being a member of a particular cluster, Expectation-Maximization (EM) algorithm(Moon, 1996), and Grid-Based Methods (Ilango & Mohan, 2010).

K-Means is useful in the cases where the user can gauge the count of clusters actually present. It is also computationally very less expensive as compared to other algorithms. K-means attempts to divide the data set into a predefined number of clusters such that the total distance between the members of each cluster and its respective centroid is minimized. Let us explain the major tenets of the algorithm.

Suppose there are N sample feature vectors o_1, o_2, ..., o_N and it is known they can be divided in L clusters where $L < N$. Let m_k be the mean of the vectors in cluster k. This suggests the following procedure for finding the k means:

```
• Make initial guesses for the means $\boldsymbol{c_1}, \boldsymbol{c_2}, \ldots, \boldsymbol{c_L}$
• Until the means do not change

    o Use the estimated means to classify the samples into clusters by allocating each data
      point to the group that has the closest mean.
    o For $i$ from $1$ to $L$
        ▪ Replace $\boldsymbol{c_i}$ with the mean of all of the samples for cluster $i$
    o End for

• End until
```

The similarity metric often chosen for k-means is the distance measure

$\left\| x_a^{(j)} - c_j \right\|$between a data point $x_a^{(j)}$ and the cluster center c_j. K-means minimizes the sum of distances from each object to its cluster centroid, over all clusters which is represented as a cost function J.

$$J = \sum_{j=1}^{L} \sum_{a=1}^{n} \left\| x_a^{(j)} - c_j \right\|^2$$

J is the sum of all distances of n data points from their corresponding clusters.

Hierarchical clustering alone might not be enough to determine the number of clusters required to give as input to the clustering algorithms. A number of inputs are used as clusters for k-means.

TRAINING A LVQ NETWORK

The LVQ Algorithm

A training set consisting of a training vector $\boldsymbol{x} = \{x_1, x_2, x_3, ..x_n\}$ and target output pairs are assumed to be given. The inputs form the input layer of the LVQ network. The numbers of neurons in the network are same as the number of classes present in the data. Let there be J classes present in the data where $k = \{1,2, .., J\}$. So there are J neurons in the output layer. All input vectors are connected to all the neurons in the network as shown in the Figure 21. The weights are also called the codebook vectors. The weight vector joining the inputs to the neuron k can be expressed as $\boldsymbol{w_k} = \{w_{1k}, x_{2k}, x_{3k}, ..x_{nk}\}$. Basically the codebook vectors act as piece wise linear functions to classify the data.

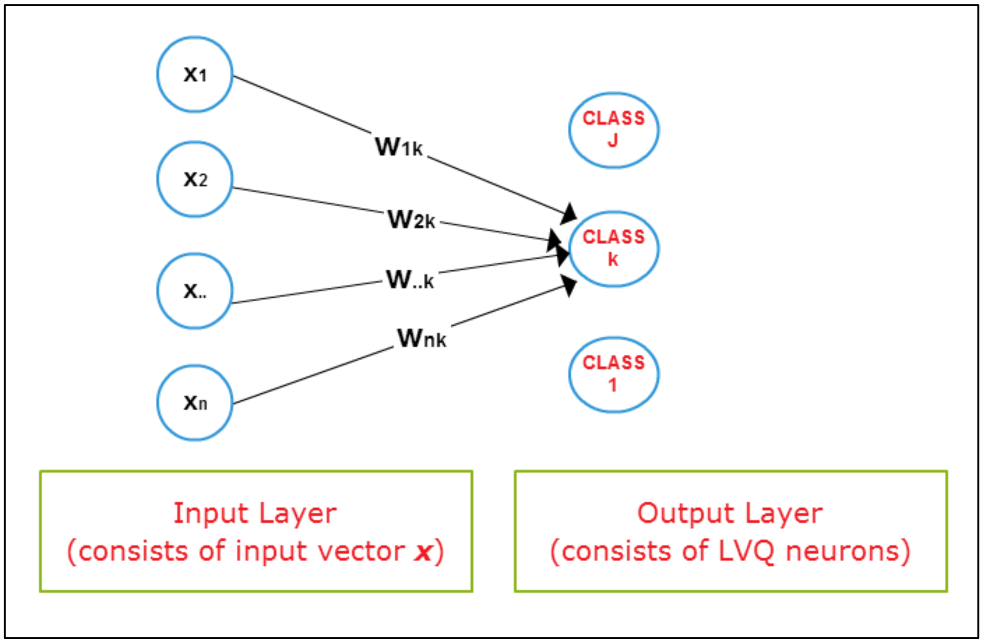


Figure 21 Structure of Learning Vector Quantization Network

The training process can be explained using the following rules:

Rule 1:

Initialize first J inputs as J weight vectors, given J classes are present in the data. Other techniques include randomly selecting J inputs from the data for initializing weights.

Rule 2

While termination criterion ≠ true

For each input vector

Calculate the distance metric $D(k)$ from the all the weight vectors.

$$D(k) = \sum_{i=1}^{n} \|\mathrm{x}_i - w_{ik}\|^2$$

Choose the k that makes $D(k)$ minimum since that is to minimized. Check whether k or predicted class of the input vector is same as the target class. If the input $\boldsymbol{x}$ and the associated weight vector $\boldsymbol{w_k}$ have the identical class tag, then update the weight vector by the attraction rule (bring it closer to the input)

$$\boldsymbol{w_k}(new) = \boldsymbol{w_k}(\boldsymbol{old}) + \boldsymbol{\eta}\ (\boldsymbol{x} - \boldsymbol{w_k}(\boldsymbol{old}))$$

If the input $\boldsymbol{x}$ and the associated weight vector $\boldsymbol{w_k}$ have different class tags, then move them apart by repulsion rule: $\boldsymbol{w_k}(new) = \boldsymbol{w_k}(\boldsymbol{old}) - \boldsymbol{\eta}\ (\boldsymbol{x} - \boldsymbol{w_k}(\boldsymbol{old}))$

Termination of training may depend upon a fixed number of iterations or setting the minimum threshold of the learning rate.

Making a Decision Based on Current Round of Negotiation

The decision to accept, reject or negotiate further with a system is based on the cooperative behavior of the system, willingness to collaborate, and the SoS's preference for acquiring that capability. After identifying the class of behavior the SoS coordinator can use a fuzzy inference engine to decide whether he wishes to accept the systems offer, reject the offer or further negotiate. Since all the three parameters are difficult to compute numerically the SoS coordinator has fuzzy linguistic model to aid in decision making.

The problem is handled as multi-criteria decision making using 2-tuple fuzzy linguistic model. The fuzzy linguistic approach represents qualitative variables as linguistic values by use of linguistic variables (Herrera & Martínez, 2000). The 2-tuple fuzzy linguistic representation model represents the linguistic information by means of a 2-tuple $(s, \alpha)$where s is a linguistic label and α is a numerical value that represents the value of the symbolic translation.

If a variable can take words in natural languages as its values, it is called a linguistic variable, where the words are characterized by fuzzy sets defined in the universe of discourse in which the variables are defined. The linguistic variable is represented by a set of membership functions.

Definition 1: Let β be the result of aggregation of the indexes of a set of labels assessed in a linguistic term set S. Then, $\beta \in [0, g]$, where $g + 1$ is the cardinality of the set S.

Let $i = round(\beta)$ and $\alpha = \beta - i$ are two values such that, $i \in [0, g]$, and $\alpha \in [-0.5,0.5]$, α is then called the symbolic translation.

Definition 2: The aggregation of the indexes β can be converted to s_i the closest index label to β and α the symbolic translation.

$$\Delta(\beta) = (s_i, \alpha)$$

For example a set S composed of four terms could be where S= {$s_0 = VL$, $s_1 = L$, $s_2 = M$, $s_3 = H$}shown in Figure 22. The first step is to assign a 2-tuple value for each alternative based on each attribute by the SoS manager. Subsequently calculate an aggregated value for each alternative over all attributes using 2-tuple Linguistic Aggregation. Finally all the alternatives are ranked based on this output. Some definitions and concepts are presented below to clarify the approach.

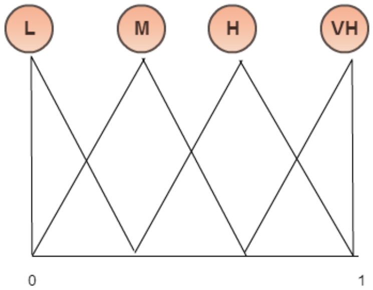


Figure 22 A set of four linguistic terms with their semantics

Table 5 General 2-tuple Linguistic Problem

Attributes/Alternatives	A1	A2
P1	(s_1, α_1)	(s_3, α_4)
P2	(s_2, α_2)	(s_1, α_1)
P3	(s_0, α_5)	(s_3, α_3)
2-tuple Linguistic Aggregation	$\beta_{A1} = (s_3, \alpha_{12})$	$\beta_{A2} = (s_2, \alpha_6)$

For the sake of ease we assume all α the symbolic translation as zero. Then alternative A1 has an aggregated value for all attributes (P1, P2, P3) as

$$\Delta\,\beta_{11} = (M, 0) => (i + \alpha_1) = 1, \beta_{11} = 1$$

$$\Delta\,\beta_{12} = (H, 0) => (i + \alpha_2) = 2, \beta_{11} = 2$$

$$\Delta\,\beta_{13} = (L, 0) => (i + \alpha_5) = 0, \beta_{11} = 0$$

$$\beta_{A1} = \frac{1+2+0}{3} = 1; i = round(1) = 1;\ \alpha = 0; hence\ (s_{A1}, \alpha_{A1}) = (M, 0)$$

$$\Delta\,\beta_{21} = (VH, 0) => (i + \alpha_1) = 3, \beta_{11} = 3$$

$$\Delta\,\beta_{11} = (M, 0) => (i + \alpha_1) = 1, \beta_{11} = 1$$

$$\Delta\,\beta_{11} = (VH, 0) => (i + \alpha_1) = 3, \beta_{11} = 3$$

$$\beta_{A2} = \frac{3+1+3}{3} = 3.33; i = round(3.33) = 3;\ \alpha = 0.33; hence\ (s_{A1}, \alpha_{A1}) = (VH, 0.33)$$

The aggregation is based on LOWA for a set of 2-tuples. Comparing or ranking the alternatives is done using the 2-Tuple Comparison Operators and alternative A2 is higher w.r.t to the rules given. The decision maker would choose alternative A2 over A1.

On the same note when this approach is applied to the SoS manager it can divide linguistic terms in classes for making a decision on choosing the alternatives. For example if the aggregated value of the alternative lies within the set of $\{s_0 = VL, s_1 = L\ \}$ the alternative is rejected. The SoS manager has a choice of making 3 kinds of decisions based on the aggregated linguistic terms of the alternatives namely: Decision of SoS :{ Negotiate, Accept, or Reject}.

Proposing an Offer

A counteroffer is made to move closer to an agreement in the multi-attribute offer space. It involves deciding the amount of concession to be made, taking into account effect of time elapsed so far and the behavior both the offer proposer and the opponent party. In all this makes quite a challenge to design offer generating strategy. An SoS coordinator can employ different time dependent and behavior dependent strategies to generate the next offer once he/she has arrived at a decision to negotiate further. An alternating protocol of offers and counteroffers is employed to reach a final decision agreeable to both parties. The convergence of a negotiation strategy (Yu, Ren, & Zhang, 2013) indicates that the negotiating agents are certain to come to an agreement if the space of available solutions within the problem is not an empty set. The following sections give an outline for three kinds of tactics based on resources, behavior and time (Matos, Sierra & Jennings, 1998).

Resource Dependent Tactics

Resource dependent tactics depend on the quantity of resource available (Faratin, Sierra, and Jennings, 1998). The tactic aims to become conciliatory with reduction in amount of resources. Resources could be time, number of systems interested in a particular negotiation or funding availability. $U(t) = \rho + (1-\rho)e^{-resource(t)}$ where $resource(t)$ is the resource available at time t.

BEHAVIOR DEPENDENT TACTICS

Behavior dependent tactics are induced from the actions of the negotiation opponent (Axelrod, 1984). The tactics include Relative Tit-For-Tat (Relative-TFT) which accounts for in percentage the behavior exhibited by the opponent over a certain time period. On the contrary, Random Absolute Tit-For-Tat (Random-TFT) accounts for the behavior in absolute terms. These tactics work well under no time restrictions or deadlines.

TIME DEPENDENT TACTICS

These tactics model the fact that the agent is likely to concede more rapidly as the negotiation deadline approaches. Two functions are generally employed for this purpose: the polynomial function and the exponential function (Faratin, Sierra, & Jennings, 1998). These functions represent an infinite number of possible tactics, one for each value of β_S (Coehoon and Jennings, 2004). The parameter β_S needs to be selected to ensure the convexity (or concavity) of the utility curve. The β however must be classified into one of the following three forms to change the behavior of the equations (Faratin):

$\beta >> 1$: This choice is made if the opponent is Conceder (reluctant) (SoS starts losing ground fairly quickly) and function is concave
$\beta = 1$: This choice is made if the opponent is Linear (SoS concedes equal amount in each round of negotiation)
$0 < \beta < 1$: This choice is made if the opponent is Boulware (SoS concedes slowly till the deadline is nearly up) and function is convex

For the exact same value (big) of strategy parameter β the polynomial function is supposed to concede quicker at the start than the exponential one after which they behave similarly (Sierra, Faratin, & Jennings, 1999). β can be used in both the equations listed below to generate the new offer by the SoS coordinator. According to the assigned class of the systems offer the SoS coordinator can choose to have different values for the strategy parameter β. For non-cooperative systems the value of β is high and for a very cooperative system its value should be kept low. Faratin has suggested exponential functions besides with the polynomial function shown below (). The common characteristic among the two functions is that both exhibit convexity w.r.t. t, and their degree of convexity is determined through the parameter β.

Polynomial: $V_p^{SoS}(t+1) = V_p^{SoS}(t) + \left|V_p^{SoS}(t) - V_p^{S}(t)\right| * \left(\frac{t}{t_{max}}\right)^{\frac{1}{\beta_S}}$

Exponential: $V_p^{SoS}(t+1) = V_p^{SoS}(t) + e^{\left((1+\frac{t}{t_{max}})^{\frac{1}{\beta_S}}\right) * \ln(|V_p^{SoS}(t) - V_p^{S}(t)|)}$

Here $0 \leq \beta_S \leq 1$ is the system's strategy parameter and t is current round of negotiation s.t. $t > 1$, $V_i^{SoS}(t)$ is the SoS's offer to the system at current negotiation round t, $V_i^{S}(t)$ is the system's offer to the SoS at time t, $V_i^{SoS}(t+1)$ is the SoS's new offer to the system (using the equations) and t_{max} is the maximum number of negotiations possible (Bahrammirzaee, Chohra, & Madani, 2013).

It is expected that by the use of these equation based offer generations the SoS manger can respond to a system on each issue.

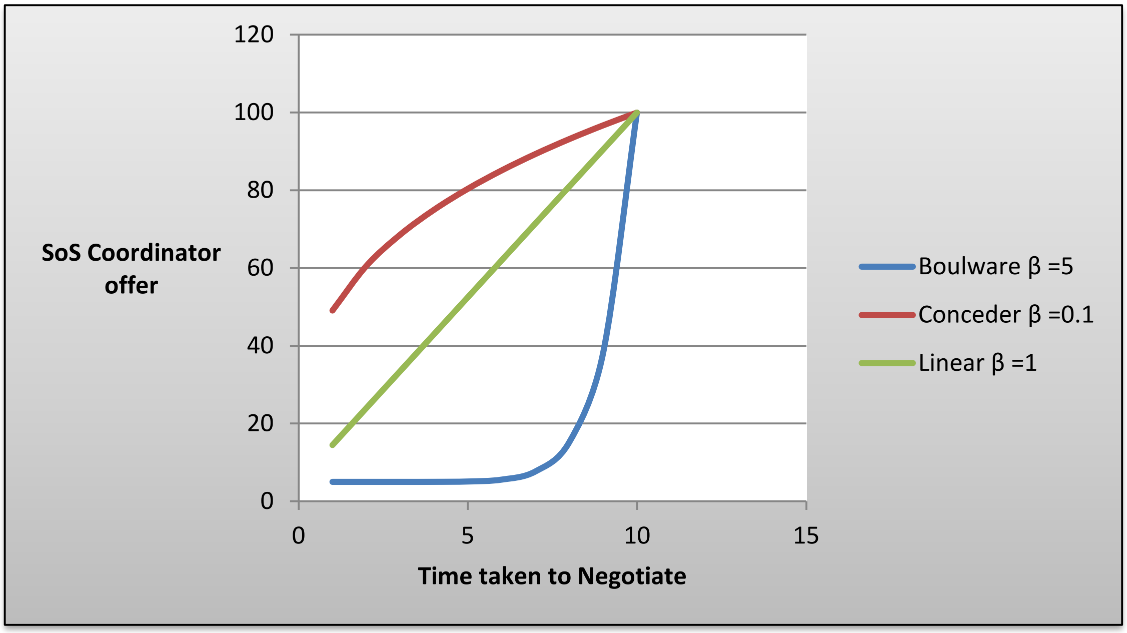


Figure 23 Examples of Concession Curves for the Polynomial Time-dependent Family of Tactics

Nevertheless a negotiator might not just respond aggressively to an aggressive opponent or quickly conceding to as conceding opponent. There are can be number of behaviors theta are possible as shown in Figure 23 based on the negotiator's attitude (Baarslag, 2014). For example, the first tactic can be described as matching the exact style of negotiation of the opponent. Where a negotiator may cooperate (or conceding) when up against a cooperative opponent, on the other hand negotiator may behave competitively (not yielding easily) with a competing system (aggressive). This negotiator can be termed as a *matcher*. The other contrary tactic is for a negotiator to behave in complete contrast to the opponent. In this tactic negotiator is cooperative towards a non-cooperative (competing) opponent. The negotiator also adopts non-yielding strategy (aggressive) to its cooperative opponents. Such a negotiator can also be called an *inverter*. In literature four types of behaviors are considered prominent namely, Inverter, Conceder, Competitor, and Matcher.

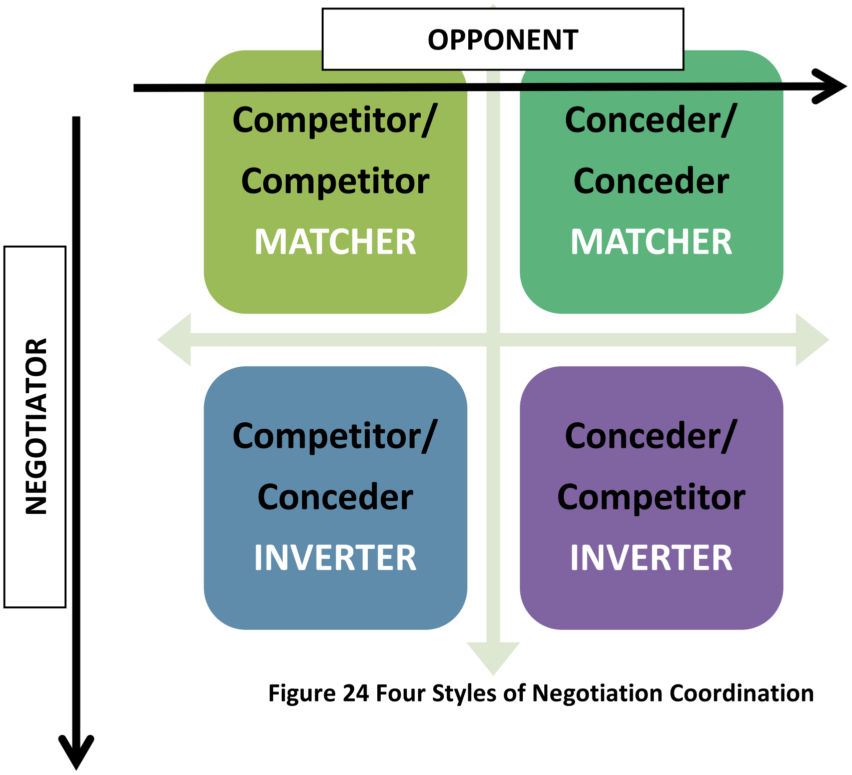


Figure 24 Four Styles of Negotiation Coordination

OVERALL NEGOTIATION PROTOCOL

The overall negotiation protocol can be illustrated as a set of statements as follows:

1) Send an offer to all systems simultaneously
2) Receive a counter-offer from all systems
3) Model the opponent behavior-(clustering)
4) First make decision on set of systems with capability i $where\ i =\ 1\ to\ M$
5) Need to select at least one system from each capability i
   a. Select a system with the best offer amongst them for the same capability if no system within a particular capability class is accepted
   b. Do so for each capability i to be acquired
   c. Form the architecture using the selecting systems and the interfaces
6) Evaluate the overall architecture quality based on the systems selected in one epoch (may contain multiple negotiation rounds)
7) If the architecture is not of a predefined quality then go for a second epoch for systems not yet selected

The next sections describe the Demonstration of meta-architecture generation results and counteroffer generation model.

Genetic Algorithm Application

The parameters used are described in Table 6. Each model was run for 100 generations and 50 times to obtain a better assessment of the stochastic techniques used. The model with the highest architecture value in 50 iterations is presented here in each case. Increasing the generations to 300 did not affect the maximum architecture quality. Hence, it was reasonable to keep the same architecture's quality that was obtained in smaller simulation time. The population size was kept as 50, probability of mutation is 0.2, size of dormant selection for next population is kept as 2, and lastly the population fraction maintained at the end of each epoch was 0.5. The best value obtained is 6.48. The set of systems selected and the interfaces is presented as circular graph in Figure 26. The systems not selected are marked as red asterisks. Systems selected are named in Table 8.

Table 6 The parameters used in GA

Generations	100,200
Population Size	50
Probability of Mutation	0.2
Tournament Selection Size	2
Population fraction kept for next generation	0.5

The best architecture obtained by GA is illustrated in Figure 25 and 26. A total number of 13 systems were selected.

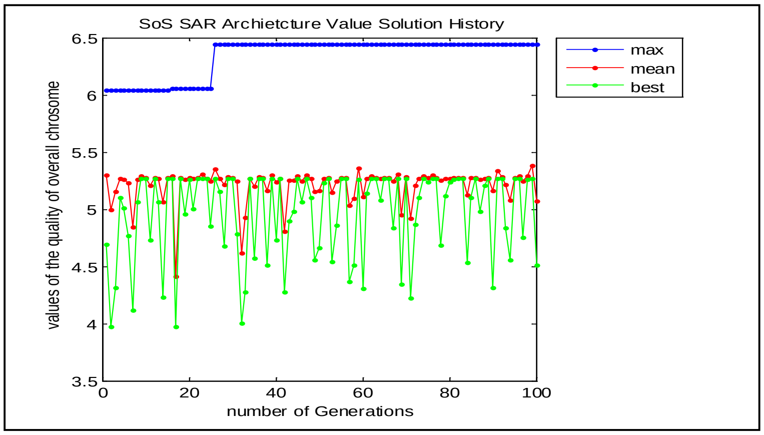


Figure 25 Maximum, Minimum and Best SoS Architecture Values

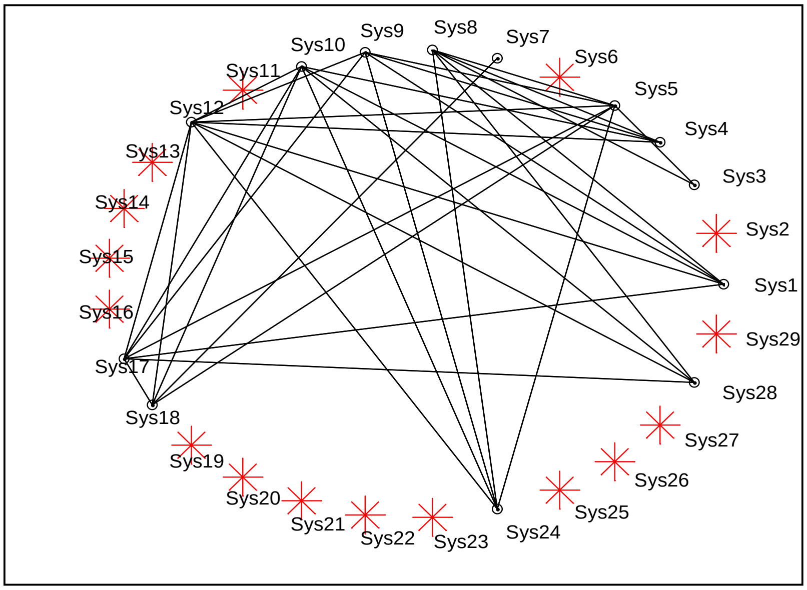


Figure 26 Undirected Graph of Systems Selected the Best SoS Architecture Obtained Through GA

BPSO APPLICATION

PSO algorithms start with a group of a randomly generated population (particles in PSO). Population individuals are evaluated by a fitness function. Both update the population and search based on the best value achieved. PSO does not have genetic operators (e.g., crossover and mutation). Particles update is based on individual position, velocity and on the best position and velocity of the swarm leader. All the above procedures are valid for PSO and BPSO. The major difference between BPSO with real-valued version is that velocities of the particles are defined in terms of probabilities that a bit will change to one or zero. Usually a sigmoid function is used to map all real valued velocities to the range of [0, 1]. The number of iterations was usually 100, population size was kept at 50, cognitive and social parameters were both equal to 2, and constriction factor was 1. The maximum and minimum velocity was maintained between -4 and 4, and inertia weight decreased linearly based on number of iterations. These are all standard parameters in PSO. The parameters used for BPSO are listed in Table 7. The best architecture obtained is depicted in Figures 27 and 28.

Table 7 The parameters used in BPSO

Iterations	100,200
Population Size	40
Cognitive Parameter	2
Social Parameter	2
Constriction Factor	1
[velocity min, velocity max]	[-4, 4]
Inertia Weight	(Maximum iterations-Current iteration)/ Maximum iterations

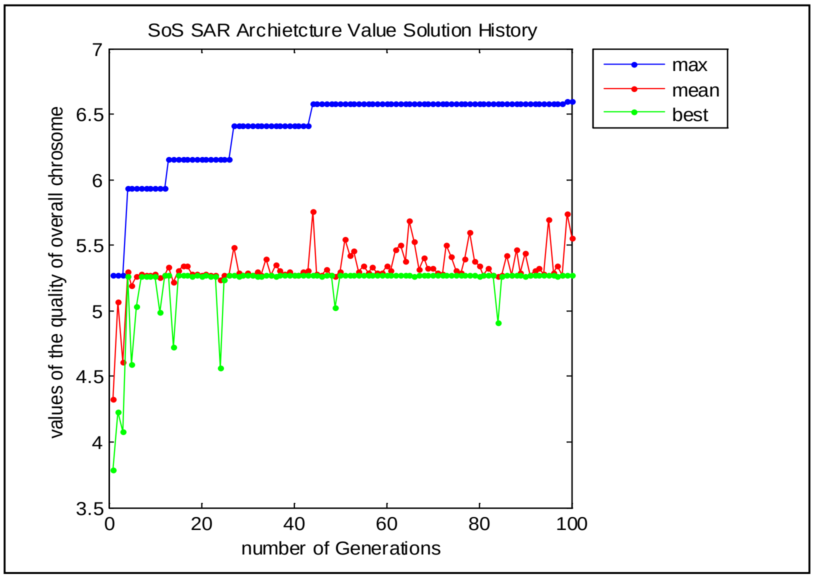


Figure 27 Max, Min and Mean SoS Architectural Value Histories Obtained Over 100 Generations via BPSO

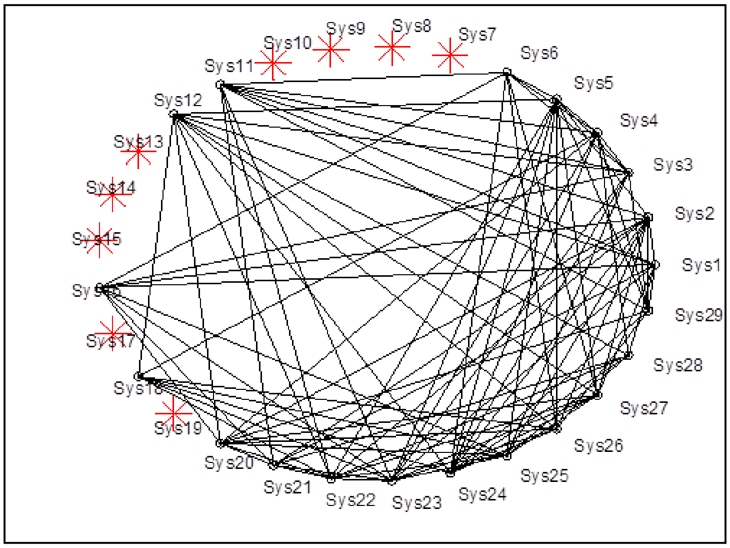


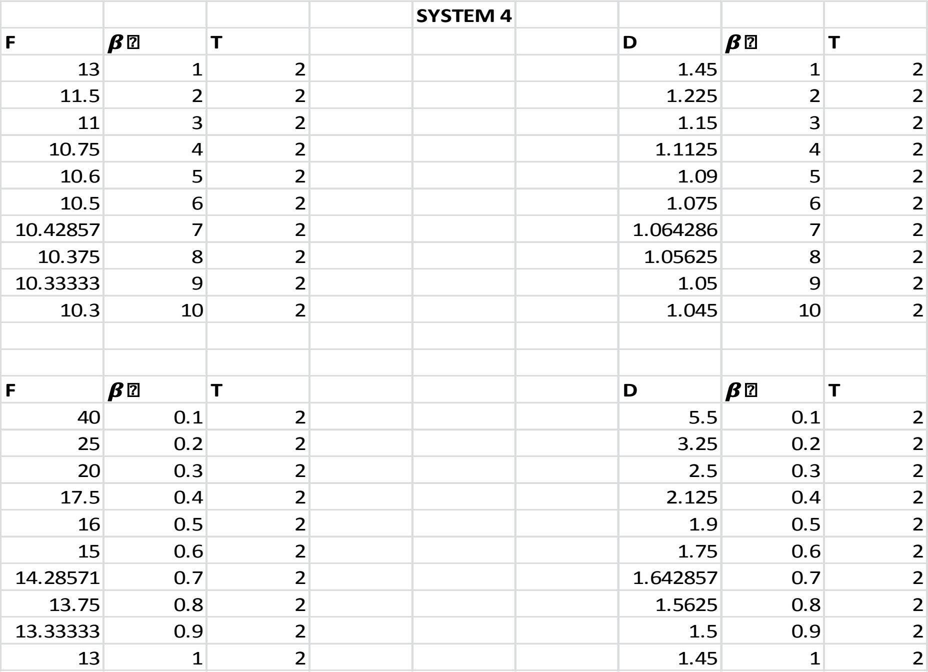
Figure 28 Circular Undirected, Graph of Systems Selected in the Best SoS Architecture Obtained through BPSO

Table 8 . Systems and the Capabilities Selected in the Best Architecture by the BPSO and GA

Systems Selected by BPSO	Capabilities Provided	Systems Selected by GA	Capabilities Provided
Systems 1,2-Cutter	2-7,9,10	Systems 1-Cutter	2-7,9,10
Systems 3,4-Helicopter	1-8,10	Systems 3-Helicopter	1-8,10
Systems 5,6-Aircraft	3,5,8,10	Systems 4,5-Aircraft	3,5,8,10
Systems 11,12,16,17-UAV	1,3,4,7,9,10	Systems 7,8,9,10,12,17-UAV	1,3,4,7,9,10
Systems 20-22 -Fish Vessel	3,4,6,7,9,10	Systems 18-Fish Vessel	3,4,6,7,9,10
Systems 23 –Civilian Ship	3,4,6,7,9,10	Systems 24 –Coordination Control	5,6,9,10
Systems 24,25 –Coordination Control	5,6,9,10	Systems 28-Communication	10
Systems 26,27,28,29-Communication	10		

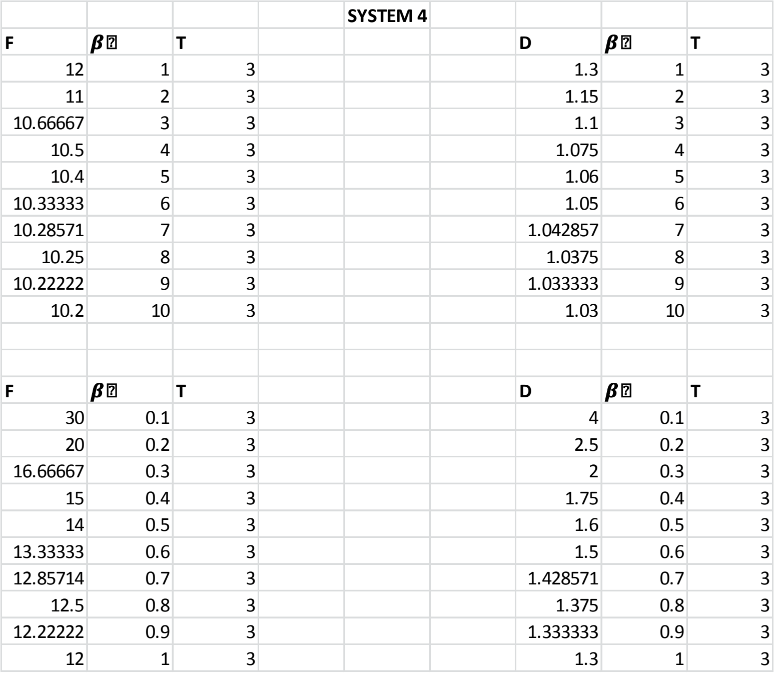
Demonstration of Counteroffer Generation

Different values of t and β are able to generate different values of SoS offers to the systems in the second round. The values in Figure 29 and 30 are based on the Polynomial: $V_p^{SoS}(t+1) = V_p^{SoS}(t) + \left|V_p^{SoS}(t) - V_p^{s}(t)\right| * (\frac{t}{t_{max}})^{\frac{1}{\beta_s}}$. Since the performance provided by system 4 is more than asked for by the SoS manger. The SoS manager negotiates only for the deadline and funding. The performance provided by system 4 is more by 1.33 units. The funding demanded is more by 3 units and higher deadline of 0.45 is also requested. The SoS manger can adopt a conceder behavior or an aggressive behavior against its opponent. In each case the amount of concession depends on SoS manager's preference. This is later represented using the tables based on a different value of β which is the strategy parameter.



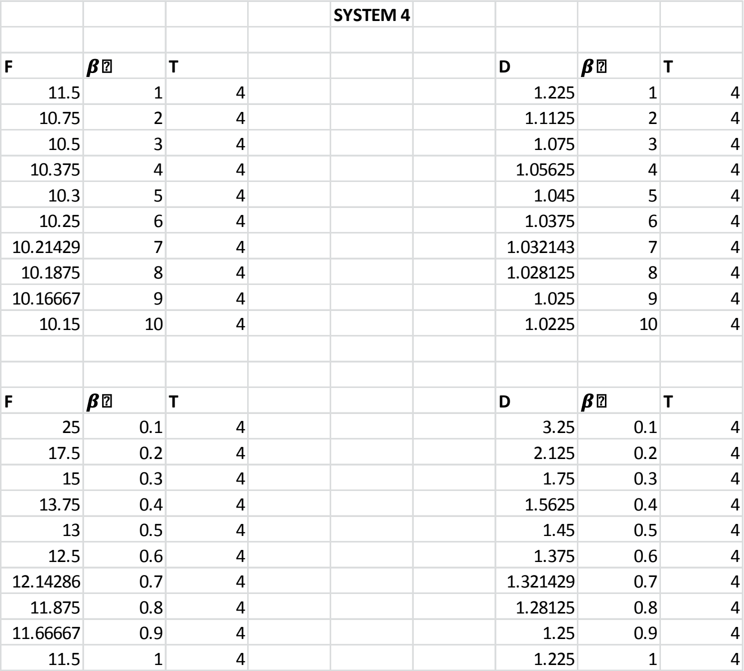
				SYSTEM 4				
F	β⍰	T				D	β⍰	T
13	1	2				1.45	1	2
11.5	2	2				1.225	2	2
11	3	2				1.15	3	2
10.75	4	2				1.1125	4	2
10.6	5	2				1.09	5	2
10.5	6	2				1.075	6	2
10.42857	7	2				1.064286	7	2
10.375	8	2				1.05625	8	2
10.33333	9	2				1.05	9	2
10.3	10	2				1.045	10	2
F	β⍰	T				D	β⍰	T
40	0.1	2				5.5	0.1	2
25	0.2	2				3.25	0.2	2
20	0.3	2				2.5	0.3	2
17.5	0.4	2				2.125	0.4	2
16	0.5	2				1.9	0.5	2
15	0.6	2				1.75	0.6	2
14.28571	0.7	2				1.642857	0.7	2
13.75	0.8	2				1.5625	0.8	2
13.33333	0.9	2				1.5	0.9	2
13	1	2				1.45	1	2

Figure 29 Counteroffers for Funding and Deadline by SoS to System 4 for t=2



				SYSTEM 4				
F	β⍰	T				D	β⍰	T
12	1	3				1.3	1	3
11	2	3				1.15	2	3
10.66667	3	3				1.1	3	3
10.5	4	3				1.075	4	3
10.4	5	3				1.06	5	3
10.33333	6	3				1.05	6	3
10.28571	7	3				1.042857	7	3
10.25	8	3				1.0375	8	3
10.22222	9	3				1.033333	9	3
10.2	10	3				1.03	10	3
F	β⍰	T				D	β⍰	T
30	0.1	3				4	0.1	3
20	0.2	3				2.5	0.2	3
16.66667	0.3	3				2	0.3	3
15	0.4	3				1.75	0.4	3
14	0.5	3				1.6	0.5	3
13.33333	0.6	3				1.5	0.6	3
12.85714	0.7	3				1.428571	0.7	3
12.5	0.8	3				1.375	0.8	3
12.22222	0.9	3				1.333333	0.9	3
12	1	3				1.3	1	3

Figure 30 Counteroffers for Funding and Deadline by SoS to System 4 for t=3



				SYSTEM 4				
F	**β**	**T**				**D**	**β**	**T**
11.5	1	4				1.225	1	4
10.75	2	4				1.1125	2	4
10.5	3	4				1.075	3	4
10.375	4	4				1.05625	4	4
10.3	5	4				1.045	5	4
10.25	6	4				1.0375	6	4
10.21429	7	4				1.032143	7	4
10.1875	8	4				1.028125	8	4
10.16667	9	4				1.025	9	4
10.15	10	4				1.0225	10	4
F	**β**	**T**				**D**	**β**	**T**
25	0.1	4				3.25	0.1	4
17.5	0.2	4				2.125	0.2	4
15	0.3	4				1.75	0.3	4
13.75	0.4	4				1.5625	0.4	4
13	0.5	4				1.45	0.5	4
12.5	0.6	4				1.375	0.6	4
12.14286	0.7	4				1.321429	0.7	4
11.875	0.8	4				1.28125	0.8	4
11.66667	0.9	4				1.25	0.9	4
11.5	1	4				1.225	1	4

Figure 31 Counteroffers for Funding and Deadline by SoS to System 4 for t=4

CONCLUDING REMARKS

The goal of this research is to model the evolution of the architecture of an acknowledged Systems of Systems (SoS) that accounts for the ability and willingness of constituent systems to support the SoS capability development. The Wave Process Model provides a framework for modeling methodology, and this research provides different sets of modules to be integrated with the rest of them. In particular, the research focuses on the impact of individual system behavior on the SoS capability and architecture evolution processes. Numerous systems have dissimilar goals, therefore integration and assimilation of information is needed to guide them to larger missions in the face of uncertainty and attacks. This research takes a step towards achieving that capability by introducing a new analysis framework that uses modeling tools to expose foreseeable SoS level impacts for decision makers early in the lifecycle, when such impacts can be managed less expensively and more solutions to possible problems can be put on the table. Different behaviors of the systems for the same architecture can help us generate possible negotiated architecture qualities. This is a very quick and effective approach to adapt communication strategies in SoS environment. Our attempt is to present an integrated acknowledged SoS architecting model 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.

APPENDIX A: LIST OF PUBLICAS RESULTED AND PAPERS SUBMITTED FROM FILA-SOS RESEARCH

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.

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.

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.

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

Ergin, D., & Dagli, C., Incentive Based Negotiation Model for System of Systems Acquisition. (Accepted by Systems Engineering Journal)

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

APPENDIX B: CITED AND RELATED REFERENCES

Abo-Sinna, M. A., & Baky, I. A. (2007). Interactive Balance Space Approach for Solving Multi-Level Multi-Objective Programming Problems. *Information Sciences, 177(16)*, 3397-3410.

Abraham, A., & Jain, L. (2005). *Evolutionary Multi-objective Optimization* (pp. 1-6). Springer London.

Acheson, P. (2010, April). Methodology for Object-Oriented System Architecture Development. In *Systems Conference, 2010 4th Annual IEEE* (pp. 643-646). IEEE.

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*.

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.

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.

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.

Ahn, J. H., Ryu, Y., & Baik, D. K. (2012). An Archietcture Description method for Acknowledged System of Systems based on Federated Architeture. *Advanced Science and Technology Letters*, *5*.

Alberts, D. S., (2011). *The Agility Advantage: A Survival Guide for Complex Enterprises and Endeavors*. Washington DC: Center for Advanced Concepts and Technology.

Alberts, D. S., Garstka, J. J., & Stein, F. P. (1999). {Network Centric Warfare: Developing and Leveraging Information Superiority}.

Alfaris, A. A. F. (2009). *The Evolutionary Design Model (EDM) for the Design of Complex Engineered Systems: Masdar City as a Case Study* (Doctoral dissertation, Massachusetts Institute of Technology).

Alves, M. J., Dempe, S., & Júdice, J. J. (2012). Computing the Pareto Frontier of a Bi-objective Bi-level Linear Problem using a Multi-objective Mixed-integer Programming Algorithm. *Optimization*, *61*(3), 335-358.

Amberg, M. (1996, October). Modeling Adaptive Workflows in Distributed Environments. In *Proc. of the 1st Int. Conf. on Practical Aspects of Knowledge Management, Basel, 30th-31th Oct*.

An, B., Lesser, V., & Sim, K. M. (2011). Strategic agents for multi-resource negotiation. *Autonomous Agents and Multi-Agent Systems*, 23(1), 114-153.

Anandalingam, G., & Friesz, T. L. (1992). Hierarchical Optimization: an Introduction. *Annals of Operations Research*, *34*(1), 1-11.

ASD(NII), D. (2010). *DoD Architecture Framework Version 2.02 (DoDAF v2.02)*. Washington DC: Department of Defense.

AT&L, O. U. S. D. (2008). Systems Engineering Guide for Systems of Systems. *Washington, DC: Pentagon*.

Arnold, A., Boyer, B., & Legay, A. (2013). Contracts and Behavioral Patterns for SoS: The EU IP DANSE Approach. *arXiv preprint arXiv:1311.3631*.

Axelrod, R. M. (2006). *The evolution of cooperation*. Basic books, Inc., Publishers, New York, USA.

Babar, M. A., Zhu, L., & Jeffery, R. (2004). A framework for classifying and comparing software architecture evaluation methods. In *Software Engineering Conference, 2004. Proceedings. 2004 Australian* (pp. 309-318). IEEE.

Bac, M., & Raff, H. (1996). Issue-by-issue Negotiations: the Role of Information and Time Preference. *Games and Economic Behavior*, *13*(1), 125-134.

Back, T., & Schwefel, H. P. (1996, May). Evolutionary computation: An overview. In Evolutionary Computation, 1996., *Proceedings of IEEE International Conference on* (pp. 20-29), IEEE.

Bader, J., & Zitzler, E. (2011). HypE: An algorithm for fast hyper volume-based many-objective optimization. *Evolutionary Computation*, 19(1), 45-76.

Bahrammirzaee, A., Chohra, A., & Madani, K. (2013). An adaptive approach for decision making tactics in automated negotiation. *Applied intelligence*, 39(3), 583-606.

Baky, I. A. (2009). Fuzzy Goal Programming Algorithm for Solving De-centralized Bi-level Multi-objective Programming Problems. *Fuzzy Sets and Systems*, *160*(18), 2701-2713.

Baky, I. A. (2010). Solving Multi-level Multi-objective Linear Programming Problems through Fuzzy Goal Programming Approach. *Applied Mathematical Modelling*, *34*(9), 2377-2387.

Baarslag, T., Hindriks, K., & Jonker, C. (2014). Effective acceptance conditions in real-time automated negotiation. *Decision Support Systems*, 60, 68-77.

Bergey, J. K., Blanchette Jr, S., Clements, P. C., Gagliardi, M. J., Klein, J., Wojcik, R., & Wood, W. (2009). US Army Workshop on Exploring Enterprise, System of Systems, System, and Software Architectures.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. The *Journal of Machine Learning Research,* 13(1), 281-305.

Bertsekas, D. P. (1999). *Nonlinear programming* (2nd ed.). Athena Scientific. ISBN 1886529-00-0.

Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies–A comprehensive introduction. *Natural Computing*, 1(1), 3-52.

Bi, X., & Xiao, J. (2012). Classification-based self-adaptive differential evolution and its application in multi-lateral multi-issue negotiation. *Frontiers of Computer Science*, 6(4), 442-461.

Bigham, J., & Du, L. (2003, July). Cooperative negotiation in a multi-agent system for real-time load balancing of a mobile cellular network. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-agent Systems* (pp. 568-575). ACM.

Binmore, K., & Vulkan, N. (1999). *Applying game theory to automated negotiation*. Netnomics, 1(1), 1-9.

Blanchard, B. S., & Fabryeky, W. J. (2010). *Systems Engineering and Analysis*. Upper Saddle River, NJ: Prentice Hall.

Bonabeau, E. (2002). Agent-based modeling: Methods and Techniques for Simulating Human Systems. *Proceedings of the National Academy of Sciences of the United States of America*, *99 (Suppl 3)*, 7280-7287.

Bouleimen, K. L. E. I. N., & Lecocq, H. O. U. S. N. I. (2003). A New Efficient Simulated Annealing Algorithm for the Resource-constrained Project Scheduling Problem and Its Multiple Mode Version. *European Journal of Operational Research*, *149*(2), 268-281.

Bradley, S. P., Hax, A. C., & Magnanti, T. L. Applied Mathematical Programming. 1977.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained Project Scheduling: Notation, Classification, Models, and Methods. *European Journal of Operational Research*, *112*(1), 3-41.

Buttner, R. (2006, December). A classification structure for automated negotiations. In *Web Intelligence and Intelligent Agent Technology Workshops*, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on (pp. 523-530). IEEE.

Cara, A. B., Wagner, C., Hagras, H., Pomares, H., & Rojas, I. (2013). Multi-objective Optimization and Comparison of Non-singleton type-1 and Singleton interval type-2 Fuzzy Logic Systems. *Fuzzy Systems, IEEE Transactions on*, *21*(3), 459-476.

Carbonneau, R., Kersten, G. E., & Vahidov, R. (2008). Predicting opponent's moves in electronic negotiations using neural networks. *Expert Systems with Applications*, 34(2), 1266-1273

Chattopadhyay, D., Ross, A. M., & Rhodes, D. H. (2008, April). A Framework for Trade-space Exploration of Systems of Systems. In *6th Conference on Systems Engineering Research*, Los Angeles, CA.

Chen, S., & Weiss, G. (2013). An efficient automated negotiation strategy for complex environments. *Engineering Applications of Artificial Intelligence*, 26(10), 2613-2623.

Christian III, J. A. (2004). *A Quantitative Approach to Assessing System Evolvability*. Houston: NASA Johnson Space Center.

Chung, F. L., & Lee, T. (1996). On fuzzy associative memory with multiple-rule storage capacity. *Fuzzy Systems, IEEE Transactions on*, 4(3), 375-384.

CJCSI 6212.01F. (12 Mar 2012). *Net Ready Key Performance Parameter (NR KPP).* Washington DC: US Dept of Defense.

Clouthier, R. J., Diamrio, M. J., & Polzer, H. W. (2009). Net Centricity and System of Systems. In M. Jamshidi, *System of Systems Engineering* (pp. 150-168). Hoboken NJ: John Wiley & Sons.

Clune, J., Mouret, J. B., & Lipson, H. (2013). The Evolutionary Origins of Modularity. *Proceedings of the Royal Society b: Biological sciences*, *280 (1755)*, 20122863.

Coehoorn, R. M., & Jennings, N. R. (2004, March). Learning on opponent's preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of the 6th International Conference on Electronic Commerce* (pp. 59-68). ACM.

Coello, C. A. C. (1999). *List of references on evolutionary multi-objective optimization*. Laboratorio Nacional de Informática Avanzada, México, 29.

Coello, C. A. C., & Lamont, G. B. (2004). *Applications of Multi-objective Evolutionary Algorithms* (Vol. 1). World Scientific.

Cohon, J. L. (1985). Multi-criteria Programming: Brief Review and Application. *Design optimization*, 163.

Coleman, J. W., Malmos, A. K., Larsen, P. G., Peleska, J., & Hains, R. (2012). COMPASS Tool Vision for a System of Systems Collaborative Development Environment. In *International Conference on System of Systems Engineering*.

Contag, G., Laing, C., Pabon, J., Rosenberg, E., Tomasino, K., & Tonello, J. (2013). Nighthawk System Search and Rescue (SAR) Unmanned Vehicle (UV) System Development. *SE4150 Design Project, Naval Postgraduate School*.

Cox Jr., L. A. (2009). *Risk Analysis of Complex and Uncertain Systems (Vol. 129)*. US: Springer-Verlag.

Crossley, W. A., & Laananen, D. H. (1996). Conceptual Design of Helicopters via Genetic Algorithm. *Journal of Aircraft*, *33*(6), 1062-1070.

Dagli, C. H., & Kilicay-Ergin, N. (2008). System of Systems Architecting. Jamshidi, M. (Ed.). In *System of Systems Engineering: Innovations for the Twenty-first Century (Vol. 58). John Wiley & Sons.*, 77-100.

Dagli, C., Ergin, N., Enke, D., Gosavi, A., Qin, R., Colombi, J.,Agarwal,S., ... & Pape, L. (2013). *An Advanced Computational Approach to System of Systems Analysis & Architecting Using Agent-Based Behavioral Model* (No. SERC-2013-TR-021-2). MISSOURI UNIV OF SCIENCE AND TECHNOLOGY ROLLA.

Dagli, C. H., Singh, A., Dauby, J. P., & Wang, R. (2009, December). Smart Systems Architecting: Computational Intelligence Applied to Trade Space Exploration and System Design. In *Systems Research Forum* (Vol. 3, No. 02, pp. 101-119). World Scientific Publishing Company.

Dahmann, J., Lane, J., Rebovich, G., & Baldwin, K. (2008, April). A Model of Systems Engineering in a System of Systems Context. In *Proceedings of the Conference on Systems Engineering Research, Los Angeles, CA, USA*.

Dahmann, J., Baldwin, K. J., & Rebovich Jr, G. (2009, April). Systems of Systems and Net-Centric Enterprise Systems. In *7th Annual Conference on Systems Engineering Research, Loughborough*.

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.

Dahmann, J. S., & Baldwin, K. J. (2008, April). Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering. In *Systems Conference, 2008 2nd Annual IEEE* (pp. 1-7). IEEE.

Dahmann, J. (2012). INCOSE SoS Working Group Pain Points. In *Proc TTCP-JSA-TP4 Meeting*. Retrieved from http://www.ndia.org/Divisions/Divisions/SystemsEngineering/Documents/NDIA-SE-MS-SoS_2013-08-20_Dahmann.pdf

Dauby, J. P., & Dagli, C. H. (2011). The Canonical Decomposition Fuzzy Comparative Methodology for Assessing Architectures. *Systems Journal, IEEE,5*(2), 244-255.

Dauby, J. P., & Upholzer, S. (2011). Exploring Behavioral Dynamics in Systems of Systems. *Procedia Computer Science*, *6*, 34-39.

Deb, K. (1999). Solving goal programming problems using multi-objective genetic algorithms. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (Vol. 1). IEEE.

Deb, K. (2000). An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, *186*(2), 311-338.

Deb, K. (2001). *Multi-objective Optimization using Evolutionary Algorithms* (Vol. 16). John Wiley & Sons.

Deb, K., & Gupta, H. (2006). Introducing Robustness in Multi-objective Optimization. *Evolutionary Computation*, *14*(4), 463-494.

Deb, K.; Jain, H., "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Non-dominated Sorting Approach, Part I: Solving Problems With Box Constraints," *Evolutionary Computation, IEEE Transactions on* , vol.18, no.4, pp.577,601, Aug. 2014

Deb, K., & Sinha, A. (2009, January). Solving Bi-level Multi-objective Optimization Problems using Evolutionary Algorithms. In *Evolutionary Multi-Criterion Optimization* (pp. 110-124). Springer Berlin Heidelberg.

DeLaurentis, D., Marais, K., Davendralingam, N., Han, S. Y., Uday, P., Fang, Z., & Gurainiello, C. (2012). *Assessing the Impact of Development Disruptions and Dependencies in Analysis of Alternatives of System of Systems*. Hoboken NJ: Stevens Institute of Technology, Systems Engineering Research Center,

Department of the Navy, (1997). *Contractor Performance Assessment Reporting System (CPARS)*. Washington DC.

Díaz, E., Tuya, J., & Blanco, R. (2003, October). Automated Software Testing Using a Metaheuristic Technique based on Tabu Search. In *Automated Software Engineering, 2003. Proceedings. 18th IEEE International Conference on* (pp. 310-313). IEEE.

Díaz, E., Tuya, J., Blanco, R., & Javier Dolado, J. (2008). A Tabu Search Algorithm for Structural Software Testing. *Computers & Operations Research*,*35*(10), 3052-3072.

Director Systems and Software Engineering, OUSD (AT&L). (2008). *Systems Engineering Guide for Systems of Systems*. Available from http://acq.osd.mil/se/doc/SE-Guid-for-SoS.pdf.

Djavanshir, G. R., Alavizadeh, A., & Tarokh, M. J. (2012). From System-of-Systems to Meta-Systems: Ambiguities and Challenges. *System of Systems*.

DoD, A. S. D. "DoD Architecture Framework Version 2.0 (DoDAF V2. 0)."*Department of Defense, Washington DC* (2009).

DoD, Navy. "Contractor Performance Assessment Reporting System (CPARS)," Washington DC (1997).

Dolado, J. J. (2000). A Validation of the Component-based Method for Software Size Estimation. *Software Engineering, IEEE Transactions on*, *26*(10), 1006-1021.

Dolado, J. J. (2001). On the Problem of the Software Cost Function. *Information and Software Technology*, *43*(1), 61-72.

Dombkins, D. (1996). *Project Managed Change: The Application of Project Management Techniques to Strategic Change Programs*. Centre for Corporate Change, Australian Graduate School of Management, University of New South Wales.

Dombkins, D. (2007). Complex Project Management. *South Carolina: Booksurge Publishing*.

Dombkins, D. H. (2013).Realizing Complex Policy: Using a Systems-of-Systems Approach to Develop and Implement Policy. *Editor's Introduction*, *Volume II, Issue 5,* 22.

Dudek, G., Jenkin, M. R., Milios, E., & Wilkes, D. (1996). A Taxonomy for Multi-agent Robotics. *Autonomous Robots*, *3*(4), 375-397.

Dudenhoeffer, D. D., & Jones, M. P. (2000). A Formation Behavior for Large-scale Micro-robot Force Deployment. In *Simulation Conference, 2000. Proceedings. Winter* (Vol. 1, pp. 972-982). IEEE.

Dutta, P. K. (1999). *Strategies and Games: Theory and Practice*. MIT Press.

Dyer, J. S. (2005). MAUT—Multi-attribute utility theory. *In Multiple criteria decision analysis: state of the art surveys* (pp. 265-292). Springer New York.

Eichfelder, G. (2010). Multi-objective Bi-level Optimization. *Mathematical Programming*, *123*(2), 419-449.

Engelbrecht, A. P. (2006). *Fundamentals of computational swarm intelligence*. John Wiley & Sons.

Epperly, T. G. W. (1995). *Global Optimization of Non-convex Non-linear Programs using Parallel Branch and Bound* (Doctoral dissertation, UNIVERSITY OF WISCONSIN–MADISON).

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

Faratin, P., Sierra, C., & Jennings, N. R. (1998). Negotiation Decision Functions for Autonomous Agents. *Robotics and Autonomous Systems*, *24*(3), 159-182.

Farmani, R., & Wright, J. A. (2003). Self-Adaptive Fitness Formulation for Constrained Optimization. *Evolutionary Computation, IEEE Transactions on*, *7*(5), 445-455.

Fatima, S. S., Wooldridge, M., & Jennings, N. R. (2002, July). Multi-issue negotiation under time constraints. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*: part 1 (pp. 143-150). ACM.

Fatima, S. S., Wooldridge, M., & Jennings, N. R. (2004). An Agenda-based Framework for Multi-issue Negotiation. *Artificial Intelligence*, *152*(1), 1-45.

Fatima, S. S., Wooldridge, M., & Jennings, N. R. (2006). Multi-issue negotiation with deadlines. *Journal Artificial Intelligence Research*. (JAIR), 27, 381-417.

Flanigan, D., & Brouse, P. (2012). System of Systems Requirements Capacity Allocation. *Procedia Computer Science*, *8*, 112-117.

Fogel, D. B. (2006). *Evolutionary Computation: Toward a new Philosophy of Machine Intelligence* (Vol. 1). John Wiley & Sons.

Folmer, E., van Gurp, J., & Bosch, J. (2003, May). Scenario-based Assessment of Software Architecture Usability. In *ICSE Workshop on SE-HCI* (pp. 61-68).

Fonseca, C. M., & Fleming, P. J. (1995). An Overview of Evolutionary Algorithms in Multi-objective Optimization. *Evolutionary Computation*, *3*(1), 1-16.

Fonseca, C. M., & Fleming, P. J. (1998). Multi-objective Optimization and Multiple Constraint Handling with Evolutionary Algorithms. I: A Unified Formulation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *28*(1), 26-37.

Freeman, L. (1994). Displaying Hierarchical Clusters. *INSNA Connections*, 17(2), 46-52.

Fry, D. N., & DeLaurentis, D. A. (2011, June). Measuring Net-centricity. In *System of Systems Engineering (SoSE), 2011 6th International Conference on*(pp. 264-269). IEEE.

Gao, J., Buldyrev, S. V., Stanley, H. E., & Havlin, S. (2011, January). Networks Formed from Interdependent Networks. *Nature Physics, 8,* 40-48, doi:10.1038/NPHYS2180.

Gao, Y., Zhang, G., & Lu, J. (2009). A Fuzzy Multi-objective Bi-level Decision Support System. *International Journal of Information Technology & Decision Making*, *8*(01), 93-108.

Garg, R., & Singh, D. (2011). ε–Pareto Dominance Based Multi-objective Optimization to Workflow Grid Scheduling. In *Contemporary Computing* (pp. 29-40). Springer Berlin Heidelberg.

Garrett, R. K., Anderson, S., Baron, N. T., & Moreland, J. D. (2011). Managing the Interstitials, a System of Systems Framework Suited for the Ballistic Missile Defense System. *Systems Engineering*, *14*(1), 87-109.

Garvey, P., & Pinto, A. (2009, June). Introduction to Functional Dependency Network Analysis. In *The MITRE Corporation and Old Dominion, Second International Symposium on Engineering Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts*.

Gatti, N., & Amigoni, F. (2004, July). A cooperative negotiation protocol for physiological model combination. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems,*-Volume 2 (pp. 655-662). IEEE Computer Society.

Ge, B., Hipel, K. W., Yang, K., & Chen, Y. (2013). A Data-centric Capability-focused Approach for System-of-systems Architecture Modeling and Analysis. *Systems Engineering*, *16*(3), 363-377.

Gegov, A. (2010). *Fuzzy Networks for Complex Systems A Modular Rule Base Approach*. Springer-Verlag Berlin Heidelberg.

Giachetti, R. E. (2012). A Flexible Approach to Realize an Enterprise Architecture. *Procedia Computer Science*, *8*, 147-152.

Glover, F. (1989). Tabu Search – Part I. *ORSA Journal on Computing*, *1*(3), 190-206.

Glover, F. (1990). Tabu Search – Part II. *ORSA Journal on Computing*, *2*(1), 4-32.

Goldberg, D. E. (1990). Genetic Algorithms in Search, Optimization and Machine Learning. *Reading: Addison-Wesley*.

Gonzalez-Zugasti, J. P., Otto, K. N., & Baker, J. D. (2000). A Method for Architecting Product Platforms. *Research in Engineering Design*, *12*(2), 61-72.

Gries, M. (2004). Methods for Evaluating and Covering the Design Space during Early Design Development. *Integration, the VLSI journal*, *38*(2), 131-183.

Grira, N., Crucianu, M., & Boujemaa, N. (2004). Unsupervised and semi-supervised clustering: a brief survey. A review of machine learning techniques for processing multimedia content, *Report of the MUSCLE European Network of Excellence* (FP6).

Guariniello, C., & DeLaurentis, D. (2013). Dependency Analysis of System-of-Systems Operational and Development Networks. *Procedia Computer Science*,*16*, 265-274.

Guttman, R. H., & Maes, P. (1998). Cooperative vs. competitive multi-agent negotiations in retail electronic commerce. In *Cooperative Information Agents II Learning, Mobility and Electronic Commerce for Information Discovery on the Internet* (pp. 135-147). Springer Berlin Heidelberg.

Hadka, D., & Reed, P. (2013). Borg: An auto-adaptive many-objective evolutionary computing framework. *Evolutionary Computation*, 21(2), 231-259.

Haimes, Y. Y. (2012). Modeling Complex Systems of Systems with Phantom System Models. *Systems Engineering*, *15*(3), 333-346.

Han, S. Y., & DeLaurentis, D. (2013). Development Interdependency Modeling for System-of-Systems (SoS) using Bayesian Networks: SoS Management Strategy Planning. *Procedia Computer Science*, *16*, 698-707.

Hansen, P., Jaumard, B., & Savard, G. (1992). New Branch-and-bound Rules for Linear Bi-level Programming. *SIAM Journal on Scientific and Statistical Computing*, *13*(5), 1194-1217.

Hao, J., & Leung, H. F. (2012, December). ABiNeS: An adaptive bilateral negotiating strategy over multiple items. In Proceedings of the The *2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology,*-Volume 02 (pp. 95-102). IEEE Computer Society.

Hassan, R., & Crossley, W. (2007). Approach to Discrete Optimization Under Uncertainty: The Population-Based Sampling Genetic Algorithm. *AIAA Journal*,*45*, 2799-2809.

Hassan, R., De Weck, O., & Springmann, P. (2004, May). Architecting a Communication Satellite Product Line. In *22nd AIAA International Communications Satellite Systems Conference & Exhibit (ICSSC). Monterey, CA*.

He, Z., Yen, G. G., & Zhang, J. (2014). Fuzzy-Based Pareto Optimality for Many-Objective Evolutionary Algorithms. *Evolutionary Computation, IEEE Transactions on*, *18*(2), 269-285.

Henson, S. A., Henshaw, M. J. D., Barot, V., Siemieniuch, C. E., Sinclair, M. A., Jamshidi, M., ... & DeLaurentis, D. (2013, June). Towards a Systems of Systems Engineering EU Strategic Research Agenda. In *System of Systems Engineering (SoSE), 2013 8th International Conference on* (pp. 99-104). IEEE.

Herrera, F., & Martínez, L. (2000). A 2-tuple fuzzy linguistic representation model for computing with words. *Fuzzy Systems, IEEE Transactions on*, 8(6), 746-752.

Herroelen, W., De Reyck, B., & Demeulemeester, E. (1998). Resource-constrained Project Scheduling: a Survey of Recent Developments. *Computers & Operations Research*, *25*(4), 279-302.

Hill Climbing. (n.d.). Retrieved October 24, 2013, from http://en.wikipedia.org/wiki/Hill_climbing

Hindriks, K., & Tykhonov, D. (2008, May). Opponent modelling in Automated Multi-issue Negotiation using Bayesian Learning. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*,-Volume 1 (pp. 331-338). International Foundation for Autonomous Agents and Multiagent Systems.

Holland, J. H. (1973). Genetic Algorithms and the Optimal Allocation of Trials. *SIAM Journal on Computing*, *2*(2), 88-105.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994, June). A niched Pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence. Proceedings of the First IEEE Conference on* (pp. 82-87). IEEE.

Hu, C. F., Teng, C. J., & Li, S. Y. (2007). A fuzzy goal programming approach to multi-objective optimization problem with priorities. *European Journal of Operational Research*, 176(3), 1319-1333.

Hunt, B. R., Lipsman, R. L., Rosenberg, J. M., Coombes, K. R., Osborn, J. E., & Stuck, G. J. (2006). *A Guide to MATLAB: for Beginners and Experienced Users*. Cambridge University Press.

Hwang, C. L., Masud, A. S. M., Paidy, S. R., & Yoon, K. P. (1979). *Multiple objective Decision Making, Methods and Applications: a State-of-the-art Survey* (Vol. 164). Berlin: Springer.

Ilango, M. R., & Mohan, V. (2010). A survey of grid based clustering algorithms. International *Journal of Engineering Science and Technology*, 2(8), 3441-3446.

INCOSE, (2011). *SYSTEMS ENGINEERING HANDBOOK v 3.2.2.* San Diego: INCOSE.

Ishibuchi, H., Tsukamoto, N., & Nojima, Y. (2008, June). Evolutionary many-objective optimization: A short review. In *IEEE Congress on Evolutionary Computation* (pp. 2419-2426).

Ito, T., Zhang, M., Robu, V., Fatima, S., & Matsuo, T. (Eds.). (2009). *Advances in agent-based complex automated negotiations* (Vol. 233). Springer.

Ito, T., Zhang, M., Robu, V., Fatima, S., & Matsuo, T. (2012). *New trends in agent-based complex automated negotiations*. Springer.

Jamshidi, M. (2008). System of Systems Engineering - New Challenges for the 21st century. *Aerospace and Electronic Systems Magazine, IEEE*, *23*(5), 4-19.

Jackson, S., & Ferris, T. L. (2013). Resilience Principles for Engineered Systems. *Systems Engineering*, *16*(2), 152-164.

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M. J., & Sierra, C. (2001). Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2), 199-215.

Jia, L., Wang, Y., & Fan, L. (2011, December). Uniform Design Based Hybrid Genetic Algorithm for Multi-objective Bi-level Convex Programming. In *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on* (pp. 159-163). IEEE.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3), 241-254.

Johnston, W., Mastran, K., Quijano, N., & Stevens, M. (2013). Unmanned Vehicle Search and Rescue Initiative. *SE4150 Design Project, Naval Postgraduate School*.

Joint Staff (2010). CJCSM 3500.04C, *UNIVERSAL JOINT TASK LIST (UJTL)*. Washing DC: Department of Defense.

Jonker, C. M., Robu, V., & Treur, J. (2007). An Agent Architecture for Multi-attribute Negotiation using Incomplete Preference Information. *Autonomous Agents and Multi-Agent Systems*, *15*(2), 221-252.

Jordan, P. R., Kiekintveld, C., & Wellman, M. P. (2007, May). Empirical game-theoretic analysis of the TAC supply chain game. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems* (p. 193). ACM.

Karnik, N. N., Mendel, J. M., & Liang, Q. (1999). Type-2 Fuzzy Logic Systems. *Fuzzy Systems*, IEEE Transactions on, 7(6), 643-658.

Kaufman, L., & Rousseeuw, P. J. (1990). *Partitioning around medoids (program pam). Finding groups in data: an introduction to cluster analysis*, 68-125.

Kazman, R., Abowd, G., Bass, L., & Clements, P. (1996). Scenario-based analysis of software architecture. *Software, IEEE*, 13(6), 47-55.

Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., & Carriere, J. (1998, August). The architecture tradeoff analysis method. In *Engineering of Complex Computer Systems, 1998. ICECCS'98. Proceedings. Fourth IEEE International Conference on* (pp. 68-78). IEEE.

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

Kilicay-Ergin, N., Enke, D., & Dagli, C. (2012). Biased Trader Model and Analysis of Financial Market Dynamics. *International Journal of Knowledge-based and Intelligent Engineering Systems*, *16*(2), 99-116.

Kinnunen, M. J. (2006). *Complexity Measures for System Architecture Models* (Doctoral dissertation, Massachusetts Institute of Technology).

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. S*cience, 220*(4598), 671-680.

Koch, P. N., Evans, J. P., & Powell, D. (2002). Inter-digitation for Effective Design Space Exploration using iSIGHT. *Structural and Multidisciplinary Optimization*, *23*(2), 111-126.

Konur, D., & Dagli, C. (2014). Military System of Systems Architecting with Individual System Contracts*. Optimization Letters*, 1-19.

Konur, D., & Golias, M. M. (2013). Cost-stable Truck Scheduling at a Cross-dock Facility with Unknown Truck Arrivals: A Meta-heuristic Approach. *Transportation Research Part E: Logistics and Transportation Review*, *49*(1), 71-91.

Köppen, M., Vicente-Garcia, R., & Nickolay, B. (2005, January). Fuzzy-pareto-dominance and its application in evolutionary multi-objective optimization. In *Evolutionary Multi-Criterion Optimization* (pp. 399-412). Springer Berlin Heidelberg.

Kosko, B. (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence/Book and Disk (Vol. 1)*. Prentice Hall.

Kraus, S. (1996). An Overview of Incentive Contracting. *Artificial Intelligence*, *83*(2), 297-346.

Kraus, S. (2001). Automated Negotiation and Decision Making in Multi-agent Environments. In *Multi-agent Systems and Applications* (pp. 150-172). Springer Berlin Heidelberg.

Krothapalli, N. K. C., & Deshmukh, A. V. (1999). Design of Negotiation Protocols for Multi-agent Manufacturing Systems. *International Journal of Production Research*, *37*(7), 1601-1624.

Lafleur, J. M. (2012). *A Markovian State-Space Framework for Integrating Flexibility into Space System Design*. Atlanta: Georgia Institute of Technology School of Aerospace Engineering Doctoral Thesis.

Lamar, B. W., & Bedford, M. A. (2009). Min-additive Utility Functions. *MITRE Corporation*.

Lane, J. A., & Bohn, T. (2013). Using SysML Modeling to Understand and Evolve Systems of Systems. *Systems Engineering*, *16*(1), 87-98.

Li, C., & Chiang, T. W. (2013). Complex Neurofuzzy ARIMA Forecasting — A New Approach Using Complex Fuzzy Sets. *Fuzzy Systems, IEEE Transactions on*, *21*(3), 567-584.

Li, M., Lin, D., & Wang, S. (2010). Solving a Type of Bi-objective Bi-level Programming Problem Using NSGA-II. *Computers & Mathematics with Applications*, *59*(2), 706-715.

Liang, Q., & Mendel, J. M. (2000). Interval type-2 fuzzy logic systems: theory and design. *Fuzzy Systems, IEEE Transactions on*, 8(5), 535-550.

Lin, R., Kraus, S., Wilkenfeld, J., & Barry, J. (2006). An automated agent for bilateral negotiation with bounded rational agents with incomplete information. *Frontiers in Artificial Intelligence and Applications*, 141, 270.

Lopes, F., Wooldridge, M., & Novais, A. Q. (2008). Negotiation among Autonomous Computational Agents: Principles, Analysis and Challenges. *Artificial Intelligence Review*, *29*(1), 1-44.

Lu, J., Zhang, G., & Dillon, T. (2008). Fuzzy Multi-objective Bi-level Decision Making by an Approximation Kth-Best Approach. *Journal of Multiple-Valued Logic & Soft Computing*, *14*.

Luo, X., Jennings, N. R., Shadbolt, N., Leung, H. F., & Lee, J. H. M. (2003). A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. *Artificial Intelligence*, 148(1), 53-102.

Luzeaux, D., System-of-Systems (and Large-Scale Complex Systems) Engineering, presentation at CSDM Conference, 2013.

Maier, M. W., & Rechtin, E. (2009). *The Art of Systems Architecting, 3rd ed.* Boca Raton: CRC Press.

Malan, R., & Bredemeyer, D. (2001). Architecture Resources. *Defining Non-Functional Requirements*. Retreived from http://www.bredemeyer.com/pdf_files/ArchitectureDecisions.pdf

Marler, R. T., & Arora, J. S. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization, 41(6),* 853-862.

Marsa-Maestre, I., Lopez-Carmona, M. A., Klein, M., Ito, T., & Fujita, K. (2012). Addressing Utility Space Complexity In *Negotiations Involving Highly Uncorrelated, Constraint-Based Utility Spaces*. Computational Intelligence.

Maskin, E. S. (1996). Theories of the Soft Budget-constraint. *Japan and the World Economy*, *8*(2), 125-133.

Matos, N., Sierra, C., & Jennings, N. R. (1998, July). Determining successful negotiation strategies: An evolutionary approach. In *Multi Agent Systems, 1998. Proceedings. International Conference on* (pp. 182-189). IEEE.

Mekdeci, B., Shah, N., Ross, A. M., Rhodes, D. H., & Hastings, D. (2014). *Revisiting the Question: Are Systems of Systems just (traditional) Systems or Are they a new class of Systems?* Cambridge, MA: Systems Engineering Advancement Research Initiative (SEAri).

Mendel, J. M. (2013). On KM algorithms for Solving Type-2 Fuzzy Set Problems. *Fuzzy Systems, IEEE Transactions on*, *21*(3), 426-446.

Mendel, J. M., & John, R. B. (2002). Type-2 Fuzzy Sets Made Simple. *Fuzzy Systems, IEEE Transactions on*, *10*(2), 117-127.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, *21*(6), 1087-1092.

Mezura Montes, E., & Coello Coello, C. A. (2005). A Simple Multi-membered Evolution Strategy to Solve Constrained Optimization Problems. *Evolutionary Computation, IEEE Transactions on*, *9*(1), 1-17.

Michalewicz, Z., & Schoenauer, M. (1996). Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, *4*(1), 1-32.

Miettinen, K. M., & Optimization, N. L. M. O. (1999). Kluwer Academic Publisher.

Migdalas, A., Pardalos, P. M., & Värbrand, P. (Eds.). (1998). *Multilevel Optimization: Algorithms and Applications* (Vol. 20). Springer.

Min, B. K., & Chang, S. H. (1991). System Complexity Measure in the Aspect of Operational Difficulty. *Nuclear Science, IEEE Transactions on*, *38*(5), 1035-1039.

Moon, T. K. (1996). The Expectation-Maximization Algorithm. *Signal Processing Magazine*, IEEE, 13(6), 47-60.

Mordecai, Y., & Dori, D. (2013). I5: A Model-Based Framework for Architecting System-of-Systems Interoperability, Interconnectivity, Interfacing, Integration, and Interaction. In *International Symposium of the International Council on Systems Engineering (INCOSE)*.

Mostafavi, A., Abraham, D., Noureldin, S., Pankow, G., Novak, J., Walker, R., ... & George, B. (2012). Risk-Based Protocol for Inspection of Transportation Construction Projects Undertaken by State Departments of Transportation. *Journal of Construction Engineering and Management*, *139*(8), 977-986.

Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. The *Computer Journal*, 7(4), 308-313.

Nguyen, T. D., & Jennings, N. R. (2006). Managing commitments in multiple concurrent negotiations. *Electronic Commerce Research and Applications*, 4(4), 362-376.

Osman, M. S., Abo-Sinna, M. A., Amer, A. H., & Emam, O. E. (2004). A Multi-level Non-linear Multi-objective Decision-making under Fuzziness. *Applied Mathematics and Computation*, *153*(1), 239-252.

Pal, N. R., & Bezdek, J. C. (1995). On cluster validity for the fuzzy c-means model. *Fuzzy Systems, IEEE Transactions on*, 3(3), 370-379.

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.

Pedrycz, W., Ekel, P., & Parreiras, R. (2011). *Fuzzy Multi-criteria Decision-making: Models, Methods and Applications*. John Wiley & Sons.

Pieume, C. O., Marcotte, P., Fotso, L. P., & Siarry, P. (2011). Solving Bi-level Linear Multi-objective Programming Problems. *American Journal of Operations Research*, *1*, 214.

Pitsko, R., & Verma, D. (2012). Principles for Architecting Adaptable Command and Control Systems. *Procedia Computer Science*, *8*, 135-140.

Rahwan, I., Kowalczyk, R., & Pham, H. H. (2002, January). Intelligent Agents for Automated One-to-Many E-commerce Negotiation. In *Australian Computer Science Communications* (Vol. 24, No. 1, pp. 197-204). Australian Computer Society, Inc.

Ravindran, A., Reklaitis, G. V., & Ragsdell, K. M. (2006). *Engineering Optimization: Methods and Applications*. John Wiley & Sons.

Räihä, O. (2008). Applying Genetic Algorithms in Software Architecture Design.

Rela, L. (2004). Evolutionary Computing in Search-based Software Engineering.

Rios, L. M., & Sahinidis, N. V. (2013). Derivative-free Optimization: a Review of Algorithms and Comparison of Software Implementations. *Journal of Global Optimization*, *56*(3), 1247-1293.

Ricci, N., Ross, A. M., Rhodes, D. H., & Fitzgerald, M. E. (2013). Considering Alternative Strategies for Value Sustainment in Systems-of-Systems (Draft). *Systems Engineering Advancement Research Initiative, Cambridge MA*.

Rosenau, W. (1991). Coalition Scud Hunting in Iraq, 1991. *RAND Corporation*.

Ross, A. M., Rhodes, D. H., & Hastings, D. E. (2008). Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value. *Systems Engineering*, 246 – 262.

Ross, S. M. (2014). *Introduction to Probability Models, 8th Edition*. Academic Press.

Rostker, B. (200, July 25). *Iraq's Scud Ballistic Missiles*. Retrieved Sep 12, 2013, from Iraq Watch: http://www.iraqwatch.org/government/US/Pentagon/dodscud.htm

Runarsson, T. P., & Yao, X. (2000). Stochastic Ranking for Constrained Evolutionary Optimization. *Evolutionary Computation, IEEE Transactions on*, *4*(3), 284-294.

Russell, S. (2009). Artificial Intelligence: A Modern Approach Author: Stuart Russell, Peter Norvig, Publisher: Prentice Hall Pa.

Sanz, J. A., Fernández, A., Bustince, H., & Herrera, F. (2013). IVTURS: A Linguistic Fuzzy Rule-Based Classification System Based On a New Interval-Valued Fuzzy Reasoning Method With Tuning and Rule Selection. *IEEE T. Fuzzy Systems*, *21*(3), 399-411.

Saxena, D. K., Duro, J. A., Tiwari, A., Deb, K., & Zhang, Q. (2013). Objective reduction in many-objective optimization: Linear and nonlinear algorithms. *Evolutionary Computation, IEEE Transactions on*, 17(1), 77-99.

Schäfer, R. (2001, October). Rules for using Multi-attribute Utility Theory for Estimating a User's Interests. In *Ninth Workshop Adaptivität und Benutzenmodellierung in Interaktiven Softwaresystemen* (pp. 8-10).

Schreiner, M. W., & Wirthlin, J. R. (2012). Challenges Using Modeling and Simulation in Architecture Development. *Procedia Computer Science*, *8*, 153-158.

Schutze, O., Lara, A., & Coello Coello, C. A. (2011). On the Influence of the Number of Objectives on the Hardness of a Multi-objective Optimization Problem*. Evolutionary Computation, IEEE Transactions on, 15(4),* 444-455.

Schwartz, M. (2010, April). Defense Acquisitions: How DoD Acquires Weapon Systems and Recent Efforts to Reform the Process. LIBRARY OF CONGRESS WASHINGTON DC CONGRESSIONAL RESEARCH SERVICE.

Selva, D., & Crawley, E. F. (2013). VASSAR: Value Assessment of System Architectures using Rules. *IEEE Aerospace Conference* (pp. 1-21). Big Sky MT: IEEE.

Shi, X., & Xia, H. S. (2001). Model and Interactive Algorithm of Bi-level Multi-objective Decision-making with Multiple Interconnected Decision Makers. *Journal of Multi-Criteria Decision Analysis*, *10*(1), 27-34.

Shtub, A., Bard, J. F., & Globerson, S. (1994). *Project Management: Engineering, Technology, and Implementation*. Prentice-Hall, Inc..

Singer, Y. (2006, November). Dynamic Measure of Network Robustness. In *Electrical and Electronics Engineers in Israel, 2006 IEEE 24th convention of* (pp. 366-370). IEEE.

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).

Sierra, C., Faratin, P., & Jennings, N. R. (1999). A service-oriented negotiation model between autonomous agents. In *Collaboration between Human and Artificial Societies* (pp. 201-219). Springer Berlin Heidelberg.

Simpson, T. W., & D'souza, B. S. (2002). Assessing Variable Levels of Platform Commonality within a Product Family using a Multi-objective Genetic Algorithm. In *Proceeding of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA*.

Singh, A., & Dagli, C. H. (2009, March). Multi-objective Stochastic Heuristic Methodology for Tradespace Exploration of a Network Centric System of Systems. In *Systems Conference, 2009 3rd Annual IEEE* (pp. 218-223). IEEE.

Singh, H. K., Isaacs, A., & Ray, T. (2011). A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *Evolutionary Computation, IEEE Transactions on*, 15(4), 539-556.

Smartt, C., & Ferreira, S. (2012). Constructing a General Framework for Systems Engineering Strategy. *Systems Engineering*, *15*(2), 140-152.

Song, Q., & Chissom, B. S. (1993). Fuzzy time series and its models. *Fuzzy sets and systems*, 54(3), 269-277.

SPEC Innovations. (2014). *Model-Based Systems Engineering Tools*. Retrieved August 20,2014, from https://www.innoslate.com/systems-engineering.

Storn, R., & Price, K. (1997). Differential Evolution–a simple and efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341-359.

Suarez, R. (2004, Dec 9). *Troops Question Secretary of Defense Donald Rumsfeld about Armor*. Retrieved Apr 14, 2014, from PBS NewsHour: http://www.pbs.org/newshour/bb/military-july-dec04-armor_12-9/

Sumathi, S., & Paneerselvam, S. (2010). *Computational Intelligence Paradigms: Theory & Applications using MATLAB*. CRC Press.

Trans-Atlantic Research and Education Agenda in Systems of Systems (T-AREA-SOS) Project, "The Systems of Systems Engineering Strategic Research Agenda," Loughborough University, Loughborough, 2013.

Talbot, F. B. (1982). Resource-constrained Project Scheduling with Time-resource Tradeoffs: The Non-preemptive Case. *Management Science*, *28*(10), 1197-1210.

Taleb, N. N. (2004). *Fooled by Randomness*. New York: Random House Trace Paperbacks.

Thompson, M. (2002). Iraq: The Great Scud Hunt. *Time Magazine*, *23*.

Tzafestas, S. G. (1994). Fuzzy systems and fuzzy expert control: an overview. *The Knowledge Engineering Review*, 9(03), 229-268.

Wall, M. B. (1996). *A Genetic Algorithm for Resource-constrained Scheduling* (Doctoral dissertation, Massachusetts Institute of Technology).

Wang, J. Q., & Zhang, H. Y. (2013). Multi-criteria Decision-making Approach based on Atanassov's Intuitionistic Fuzzy Sets with Incomplete Certain Information on Weights. Fuzzy Systems, IEEE Transactions on, 21(3), 510-515.

Wang, R., & Dagli, C. H. (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.

Wang, R., & Dagli, C. H. (2013). Developing a Holistic Modeling Approach for Search-based System Architecting. *Procedia Computer Science*, *16*, 206-215.

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)*.

Wang, Y., & Cai, Z. (2012). Combining Multi-objective Optimization with Differential Evolution to Solve Constrained Optimization Problems. *Evolutionary Computation, IEEE Transactions on*, *16*(1), 117-134.

Wang, Y., Cai, Z., Guo, G., & Zhou, Y. (2007). Multi-objective Optimization and Hybrid Evolutionary Algorithm to Solve Constrained Optimization Problems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *37*(3), 560-575.

Wang, Y., & Yang, Y. (2009). Particle swarm optimization with preference order ranking for multi-objective optimization. *Information Sciences*, 179(12), 1944-1959.

Wanyama, T., & Homayoun Far, B. (2007). A Protocol for Multi-agent Negotiation in a Group-choice Decision Making Process. *Journal of Network and Computer Applications*, *30*(3), 1173-1195.

Wappler, S. (2007). *Automatic Generation of Object-oriented Unit Tests using Genetic Programming* (Doctoral dissertation, Universitätsbibliothek).

Wappler, S., & Wegener, J. (2006, July). Evolutionary Unit Testing of Object-oriented Software using Strongly-typed Genetic Programming. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation* (pp. 1925-1932). ACM.

Warfield, J. N. (1973). Binary Matrices in Systems Modeling. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-3* (No. 5, September), 441-449.

Weiss, W. E. (2008, December). Dynamic Security: An Agent-based Model for Airport Defense. In *Simulation Conference, 2008. WSC 2008. Winter* (pp. 1320-1325). IEEE.

Welby, S. P. Systems Engineering FY 2012 Annual Report.

Woldesenbet, Y. G., Yen, G. G., & Tessema, B. G. (2009). Constraint Handling in Multi-objective Evolutionary Optimization. *Evolutionary Computation, IEEE Transactions on*, *13*(3), 514-525.

Wollkind, S., Valasek, J., & Ioerger, T. R. (2004, August). Automated conflict resolution for air traffic management using cooperative multiagent negotiation. In *AIAA guidance, navigation, and control conference* (pp. 16-19).

Wooldridge, M., & Parsons, S. (2000, August). Languages for Negotiation. In *ECAI* (pp. 393-400).

Wu. D., (2013) A Brief Tutorial on Interval Type-2 Fuzzy Sets and Systems.

Wu, D., & Mendel, J. M. (2007). Uncertainty Measures for Interval Type-2 Fuzzy Sets. *Information Sciences*, *177*(23), 5378-5393.

Yang, S., Li, M., Liu, X., & Zheng, J. (2013). A Grid-Based Evolutionary Algorithm for Many-Objective Optimization. *IEEE Trans. Evolutionary Computation*, 17(5), 721-736.

Yen, G.G.; Zhenan He, "Performance Metric Ensemble for Multiobjective Evolutionary Algorithms," *Evolutionary Computation, IEEE Transactions on* , vol.18, no.1, pp.131,144, Feb. 2014

Yi, J. S., Kang, Y., Stasko, J. T., & Jacko, J. A. (2007), Nov/Dec). Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics, 13(6),* 1224-1231.

Yu, C., Ren, F., & Zhang, M. (2013). An adaptive bilateral negotiation model based on Bayesian learning. In *Complex Automated Negotiations: Theories, Models, and Software Competitions* (pp. 75-93). Springer Berlin Heidelberg.

Yu, O. Y., Guikema, S. D., Briaud, J. L., & Burnett, D. (2012). Sensitivity Analysis for Multi-attribute System Selection Problems in Onshore Environmentally Friendly Drilling (EFD). *Systems Engineering*, *15*(2), 153-171.

Yu, T. L., Yassine, A. A., & Goldberg, D. E. (2007). An Information Theoretic Method for Developing Modular Architectures using Genetic Algorithms. *Research in Engineering Design*, *18*(2), 91-109.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3), 338-353.

Zadeh, L. A. (1975). Fuzzy Logic and Approximate Reasoning. *Synthese*, *30*(3-4), 407-428.

Zhang, T., Hu, T., Zheng, Y., & Guo, X. (2012). An Improved Particle Swarm Optimization for Solving Bi-level Multi-objective Programming Problem. *Journal of Applied Mathematics*, *2012*.

Zhang, T., Hu, T., Chen, J. W., Wan, Z., & Guo, X. (2012, November). Solving Bi-level Multi-objective Programming Problem by Elite Quantum Behaved Particle Swarm Optimization. In *Abstract and Applied Analysis* (Vol. 2012). Hindawi Publishing Corporation.

Zhang, G., Lu, J., & Gao, Y. (2008). An Algorithm for Fuzzy Multi-objective Multi-follower Partial Cooperative Bi-level Programming. *Journal of Intelligent and Fuzzy Systems*, *19*(4), 303-319.

Zhang, G., Lu, J., & Gao, Y. (2008). Fuzzy Bi-level Programming: Multi-objective and Multi-follower with Shared Variables. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *16*(supp02), 105-133.

Zhang, G., & Lu, J. (2010). Fuzzy Bi-level Programming with Multiple Objectives and Cooperative Multiple Followers. *Journal of Global Optimization*, *47*(3), 403-419.

Zhang, Q., & Li, H. (2007). MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6), 712-731.

Zheng, R., Chakraborty, N., Dai, T., & Sycara, K. (2013, May). Multi-agent negotiation on multiple issues with incomplete information. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (pp. 1279-1280). International Foundation for Autonomous Agents and Multiagent Systems.

Zheng, Y., Wan, Z., & Wang, G. (2011). A Fuzzy Interactive Method for a Class of Bi-level Multi-objective Programming Problem. *Expert Systems with Applications*, *38*(8), 10384-10388.

Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., & Zhang, Q. (2011). Multi-objective Evolutionary Algorithms: A Survey of the State of the Art. *Swarm and Evolutionary Computation*, *1*(1), 32-49.