01 Nov 2016

Genetic Algorithm Optimization of SoS Meta-Architecture Attributes for Fuzzy Rule Based Assessments

Andrew Renault

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

Recommended Citation

A. Renault and C. H. Dagli, "Genetic Algorithm Optimization of SoS Meta-Architecture Attributes for Fuzzy Rule Based Assessments," *Procedia Computer Science*, vol. 95, pp. 95-102, Elsevier, Nov 2016.
The definitive version is available at https://doi.org/10.1016/j.procs.2016.09.298

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 95 (2016) 95 – 102

Procedia Computer Science

Complex Adaptive Systems, Publication 6
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2016-Los Angeles, CA

Genetic Algorithm Optimization of SoS Meta-Architecture Attributes for Fuzzy Rule Based Assessments

Andrew Renault*, Cihan Dagli

Missouri University of Science and Technology, Rolla, MO-65409, USA

Abstract

The analysis of an acknowledged systems of systems (SoS) meta-architecture requires a preliminary method for potential trade space exploration to ensure compliance to evolving capability requirements. It is important to assess the SoS meta-architecture concept to ensure that it satisfies all stakeholder needs and requirements in the early stages of development. There are numerous linguistic terms called key performance attributes (KPAs) that could be used to assess the different aspects of the architectures capabilities, however, too many KPAs could complicate the assessment. The initial population of suitable KPAs is reduced through non-derivative based optimization employed by a genetic algorithm (GA) that generates the ideal KPA candidates though optimal rank selection. A Mamdani-type rule based fuzzy inference system (MRBFIS) is then used to make a fuzzy assessment of the SoS meta-architecture concept using GA optimized and assessed KPAs as MRBFIS inputs. The MRBFIS is a beneficial addition to an architecture assessment because it enables a nonlinear output that allows a more dynamic and adjustable assessment. The integrated assessment method detailed in this paper utilizes the GA optimized KPAs and the MRBFIS to provide a valuable fuzzy assessment of SoS meta-architecture concepts to determine if the architecture is feasible and acceptable.

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology

Keywords: rule based fuzzy inference systems; systems of systems; meta-architecture; genetic algorithm, assessment

1. Introduction

The systems architecting generation process involves several stages that lead to the reduction of systems ambiguity and the development of a conceptual systems architecture. The development of the acknowledged systems of systems (SoS) meta-architecture concept requires a preliminary method for exploring a potential SoS conceptual architecture trade space [1] to ensure compliance to stakeholder needs and requirements. The systems

* Corresponding author. Tel.: 1-425-328-1013
E-mail address: andrew11@engineer.com

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology
doi:10.1016/j.procs.2016.09.298

architecting process consists of scoping, aggregating, partitioning, integrating and finally validating the systems architecture. The architecting process is also the process by which standards, protocols, rules, system structures, and interfaces are created in order to achieve the requirements of a system [2]. It is important that the SoS meta-architectures capabilities be assessed by the systems architect in the very early stages of architecture development to determine if the architecture is feasible and meets all known customer needs and requirements.

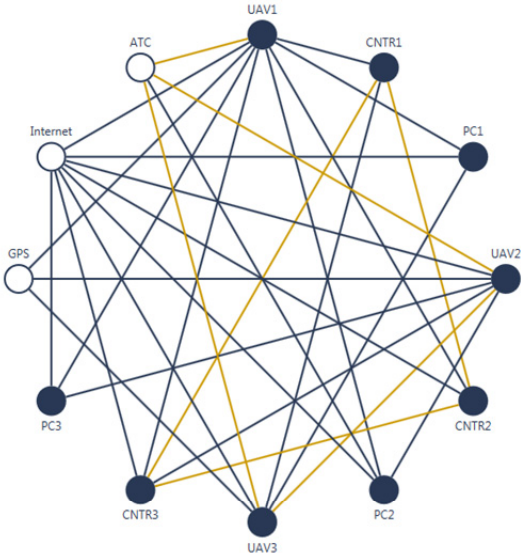
There is a current lack of formal methods of architecture assessment [4, 5] made specifically for architecture concepts. A wide range of options exist for assessing conceptual SoS meta-architecture capabilities. Linguistic terms called key performance attributes (KPAs), or also called *Ilities* [3], can be used to represent the fuzzy capabilities of the SoS meta-architecture concept. A preliminary overall assessment by suitable KPAs is beneficial to compare the architecture with the customer requirements as they are understood at the time of development. A large population of suitable KPAs could be reduced to an appropriate quantity through genetic algorithm (GA) rank selection methods, since too many KPAs could complicate the initial architecture assessment.

The integrated assessment model that is detailed in this paper, and summarized in Figure 7, is intended to provide a formal method of assessing conceptual SoS meta-architecture concepts in the early stages of development. The conceptual architecture is first assessed by GA optimized KPAs using established methods and systems engineering tools [6] of the architects choice to generate individual KPA scores. The assessed KPA scores are then used as fuzzy inputs to a Mamdani-type rule based fuzzy inference system (MRBFIS). The MRBFIS provides an If-Then rule based assessment output that is nonlinear and more adjustable than other assessment methods.

1.1. Acknowledged Systems of Systems (SoS) Meta-Architecture

The SoS meta-architecture is the conceptual model that defines the structure, behaviour, and more views of the SoS. It is essential to create a tangible concept that fulfils all known customer requirements. The architecture concept may evolve, adapt, and improve throughout the generation stages until an acceptable architecture is realized. Systems scoping and systems aggregation translates the customer needs into specific KPAs such as; modularity, reliability, robustness, flexibility, and others that best represent the architectures overall capabilities.

The SoS meta-architecture that is to be used as an example in this paper is the semi-autonomous aerial surveillance (SAAS) SoS meta-architecture. Figure 1a represents the network diagram of the SAAS SoS meta-architecture with the upper triangular (UT) matrix of system interfaces shown on Figure 1b. It is assumed that the SAAS SoS meta-architecture components and interfaces are acceptable and an assessment will be made on this concept using five KPAs that best represent the overall system capabilities based on the customer requirements.



	UAV1	CNTR1	PC1	UAV2	CNTR2	PC2	UAV3	CNTR3	PC3	GPS	Internet	ATC
UAV1	1	1	1	0	1	1	0	1	1	1	1	1
CNTR1		1	0	0	1	0	1	1	0	0	0	0
PC1			1	0	0	0	1	0	0	0	1	0
UAV2				1	0	1	1	1	1	1	1	1
CNTR2					1	0	0	1	0	0	1	0
PC2						1	0	0	0	0	1	1
UAV3							1	0	0	1	1	1
CNTR3								1	0	0	1	0
PC3									1	0	1	0
GPS										0	0	0
Internet											0	0
ATC												0

Figure 1. (a) SAAS SoS meta-architecture network diagram (b) SAAS SoS meta-architecture UT matrix

1.2. Selection of the Key Performance Attributes

A formal method of architecture assessment is beneficial to the systems architect to assess to the SAAS SoS meta-architecture to ensure compliance to stakeholder needs and requirements. The assumption is that the SAAS SoS meta-architecture (see Figure 1a) is at the very early stages of conceptual development immediately after the customer needs, capability requirements, and feasibility studies have been validated. A KPA architecture assessment is an effective method to evaluate the tangibility of the generated SoS meta-architecture concept systems and interfaces prior to the development of a prototype.

There are numerous suitable KPAs that could be used to assess the SoS meta architecture. Each KPA is unique and some may carry more importance individually while other KPAs are dependent on the ranking of others collectively. It may be difficult to select the optimum KPAs to be considered for the assessment and that is why ten KPAs are initially selected as a starting point. The population of ten KPAs is to be reduced to five KPAs because this quantity is practical and effective when used as fuzzy assessment inputs to the MRBFIS. The KPA population is be reduced to five attributes similar to what is shown on Figure 2 below:

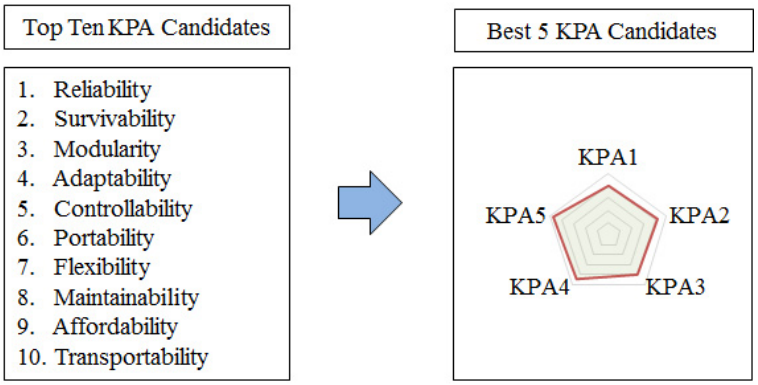


Figure 2. KPA selection for the SoS meta-architecture fuzzy assessment

1.3. Review of Related Work

Pape et al [1] explore the development of the Flexible and Intelligent Learning Architectures for SoS (FILA-SoS) meta-architecture approach to acknowledged SoS analysis that allows a method for exploring a potential SoS architecture space. This paper [1] details the process of building the lists of desirable fuzzy attributes of a SoS, developing rules for combining attribute values to an overall assessment, and discovering membership function characteristics that work well. Pape et al [1] also have recommendations for finding the appropriate combination for the adjustable parameters of fuzzy assessment models through random architecture chromosome testing and iteration.

Ke Dou et al [3] write about the difficulty in effectively documenting requirements for non-functional properties (ilities) and that the language used is often ambiguous and imprecise. The problem statement indicated that there is lack foundational scientific and engineering knowledge, and corresponding language, needed to manage the broad range of non-functional system properties for complex systems. These properties include *ilities* such as changeability, affordability, dependability, usability, and many more. Selecting the incorrect *ilities* can attribute to costly project and operational failures, major surprises late in development, unacceptable risks, costs, and difficulties in developing and certifying critical systems. The paper proposes an embedded theory-systems (ETS) alternative approach [3] to replace the natural language with theories (models) in an expressive formal language to drive theory evolution and validation.

C.H. Nguyen et al [7] state that the determination of fuzzy information that includes the estimation of their membership functions play a significant role in fuzzy system design as well as in the design of fuzzy rule based classifiers (FRBCSs). There can be problem of linguistic term design along with their fuzzy-set-based semantics because the term sets of attributes may not been interpreted as a formalized structure. The problem of the natural

semantics of terms behind the linguistic literal has not been addressed. In this paper, the problem is introduced in the design of optimal linguistic terms and propose a method of the design of FRBCSs. A series of experiments concerning 17 Machine Learning datasets is reported in this paper [7]. It was that concluded that fuzzy rule based systems (FRBSs), including Mamdani fuzzy rule based systems (MFRBSs), and FRBCSs are flexible, useful and practically relevant structures, since they can deal with fuzzy linguistic information and have a potential ability to adapt to many problems in various application areas.

1.4. Selecting SoS Evaluation Attributes by Genetic Algorithm Optimization

The systems architect selects the best ten KPAs and assigns a rank multiplier (between 0 & 1) to each KPA candidate based on a combination of a heuristic approach and matrices found within the systems engineering toolbox [6]. The rank multiplier is used to signify the percentage the KPA scores in capturing the primary customer requirements. The rank multiplier of a KPA could change with changes to the architecture. For example, the rank multiplier for the KPA of portability could potentially decrease when this need is no longer a mandatory customer requirement. The KPA rank multiplier values as shown in Figure 3 are estimated values of the SoS meta architecture capability attributes (see Figure 1a). The best five KPAs will be selected by GA rank selection method because this KPA quantity is sufficient for assessment purposes and is also ideal for use as fuzzy inputs to the MRBFIS.

The SoS meta-architecture KPA chromosome consists of a ten bit binary string and each bit of the chromosome represents respective KPAs as shown on Figure 3. Each iteration of the chromosome has exactly five 1 and five 0 bits for every non-repeating combination that exists. The selection process involves a genetic algorithm rank selection method to identify the best KPA arrangements based on the best combined score as illustrated on Figure 4. The GA rank selection method has the advantage of adjusting the KPAs to best represent the customer requirements and architecture changes to determine the highest scoring SoS meta architecture assessment chromosome.

SoS Meta-Architecture KPA 10 bit binary chromosome										
KPA Number	1	2	3	4	5	6	7	8	9	10
Chromosome 1 (start)	0	0	0	0	0	1	1	1	1	1
252 non-repeating iterations										
Chromosome 252 (end)	1	1	1	1	1	0	0	0	0	0
Note: Every chromosome contains an equal number of 1s & 0s										

Rank multipliers (r_n) assigned To each bit

KPA#	KPA Description	r_n
1	Reliability	1
2	Survivability	0.8
3	Modularity	0.85
4	Adaptability	0.7
5	Controllability	1
6	Portability	0.65
7	Flexibility	0.9
8	Maintainability	0.7
9	Affordability	0.85
10	Transportability	0.75

Figure 3. SoS meta-architecture KPA chromosome population and genetic algorithm rank criteria

The GA rank selection is facilitated using MATLAB® code and chromosomes were generated to produce a population with 252 non-repeating iterations of equal number of 1s and 0s combinations within the ten bit binary chromosome. Each chromosome generated a respective score as follows:

where b_n = the logic state of the chromosome bit that represents KPA(n)
and r_n = the KPA rank multiplier
and C_S = total chromosome score

$$C_S = b_1(r_1) + b_2(r_2) + b_3(r_3) + b_4(r_4) + b_5(r_5) + b_6(r_6) + b_7(r_7) + b_8(r_8) + b_9(r_9) + b_{10}(r_{10}) \quad (1)$$

The results of the GA rank selection method is shown in Figure 4 and only the highest scoring chromosomes are selected while the remainder are ignored. The best KPA chromosome selected reliability, modularity, controllability, flexibility, and affordability. Using formula (1) above, the highest scoring chromosome is calculated:

$$C_{S(max)} = 1(1) + 0(0.8) + 1(0.85) + 0(0.7) + 1(1) + 0(0.65) + 1(0.9) + 0(0.7) + 1(0.85) + 0(0.75) = \mathbf{4.60}$$

SoS Meta-Architecture Genetic Algorithm Optimized KPA Chromosome											
KPA#	KPA1	KPA2	KPA3	KPA4	KPA5	KPA6	KPA7	KPA8	KPA9	KPA10	$C_{S(max)}$
GA Optimized Chromosome	1	0	1	0	1	0	1	0	1	0	4.60

KPA#	KPA Description
1	Reliability
3	Modularity
5	Controllability
7	Flexibility
9	Affordability

Figure 4. SoS meta-architecture GA optimization of KPA chromosome using the rank selection method

1.5. Fuzzy Evaluations of Attributes

The GA using rank selection determined the best KPA chromosome as shown in Figure 4. The next stage of the assessment process involves a fuzzy assessment of the SoS meta-architecture using the MRBFIS. The MRBFIS was designed using MATLAB® and contained five inputs and one output as illustrated on Figure 6a. The advantage of the MRBFIS is that it can provide a nonlinear logic based output that is not possible using the sum or average of the five input KPAs. Each of the five KPA input variables assigned to the MRBFIS have five membership functions (MF) named from "lowest" to "highest" as illustrated on Figure 5a. The output variables assigned to the MRBFIS have five MFs named from "lowest" to "highest" as illustrated on Figure 5b. Let U be a *universe of discourse* and a fuzzy value in U is characterized by a fuzzy set F in U. For all MRBFIS inputs and outputs, a membership function $\mu F: U \rightarrow [0, 1]$ is defined for all of the fuzzy ranges.

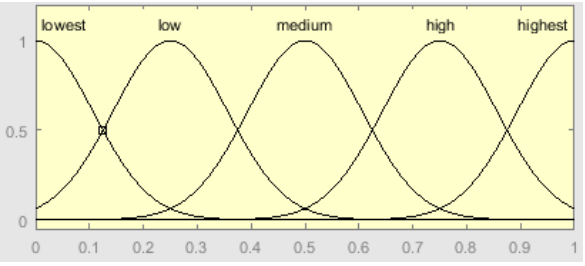
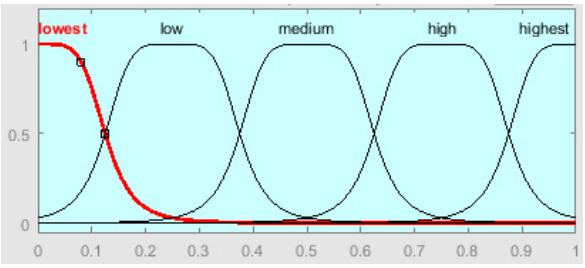


Figure 5. (a) Generalized bell curve MF



(b) Gaussian curve MF

The MRBFIS contained twenty five individual If-Then rules that were developed using the MATLAB® fuzzy logic designer application. The rules are configured manually and are specific to the individual KPAs illustrated in Figure 4. Changes to different KPA selections though GA optimization would likely require a manual revision to the fuzzy rule base. The fuzzy rules when combined with the five MFs allow an adjustable nonlinear MRBFIS output. The MRBFIS rule base determines the assessment output based on the logical arrangement of inputs in respect to the parameters of the MFs. For example, it is possible to have the "high" MRBFIS output even when a single KPA input variable only scored in the "medium" MF range as shown in the example of Rule 1 below:

Rule 1: If (Reliability is high) and (Controllability is high) and (Modularity is medium) and (Affordability is high) and (Flexibility is high) then (output1 is high) (1)

It is also possible to have a "low" MRBFIS output even though four KPA input variables scored in the "high" range as shown in the example of Rule 2 as follows:

Rule 2: If (Reliability is low) and (Controllability is high) and (Modularity is high) and (Affordability is high) and (Flexibility is high) then (output1 is low) (1)

An overview of the MRBFIS is shown in Figure 6a and the MRBFIS generated output surfaces (MATLAB® screenshots) can be viewed in Figure 6b. The generated output surfaces illustrate the nonlinear output capabilities of the MRBFIS that are more adjustable than the sum or average of the KPA inputs. The MRBFIS is capable of making a fuzzy assessment of the SAAS SoS meta-architecture using the five KPA input variables that were optimized by

the genetic algorithm. It is expected that this MRBFIS will effectively assess the SAAS SoS meta-architecture to generate a numeric or linguistic assessment output that will rate the architectures compliance with customer needs and capability requirements.

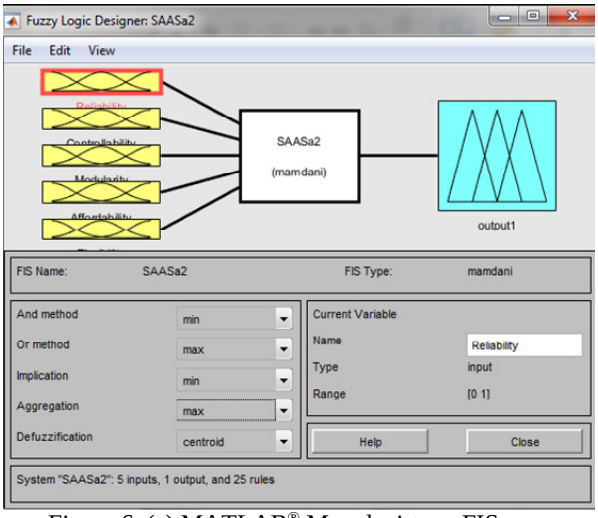
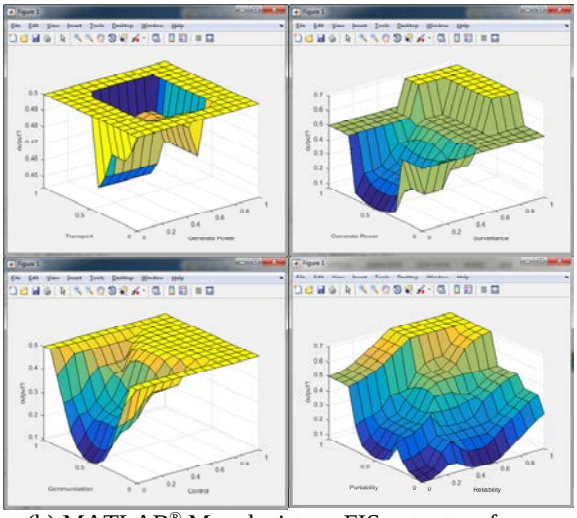


Figure 6. (a) MATLAB® Mamdani-type FIS



(b) MATLAB® Mamdani-type FIS output surfaces

2. Proposed SoS Meta-Architecture Assessment Model using GA Optimized KPAs and a MRBFIS

The SAAS SoS meta-architecture assessment model (see Figure 7) shows the process that is used to assess the SAAS SoS meta-architecture (see Figure 1a) in the early stages of development. The best five KPAs (see Figure 4) are selected through non-derivative based optimization employed by a genetic algorithm (GA).

The five KPAs selected by the GA are used individually to assess SAAS SoS meta-architecture prior to being used as inputs to the MRBFIS. The systems architect determines an assessment method to generate a score for each KPA (between 0 & 1). The KPA scores are then to be supplied as MRBFIS inputs used in the fuzzifier stage of the assessment model. The KPA scores are determined by the systems architect using systems engineering matrices commonly found in the *Systems Engineering Toolbox* [6].

The inference section of the assessment model is the process that formulates the mapping from the five given KPA inputs to a single output using fuzzy logic and If-Then rules. The process of fuzzy inference involves all of the formulated integrated controls contained in the MFs (see Figure 5), logical operations, and the twenty five If-Then rules. The fuzzy If-Then rules are expressions of the form “If X Then Y”, where X and Y are labels of fuzzy sets characterized by the appropriate MFs. The set of If-Then rules relate to a fuzzy logic system that are stored together and called a fuzzy rule base. It should be noted that the twenty five If-Then rules is an arbitrary number for the example in this paper. The assessment model is also effective with fewer MRBFIS If-Then rules.

The final stage of the assessment model is called the defuzzifier and this stage converts the fuzzy sets inputted into the MRBFIS into a crisp output that is either a linguistic or numeric value. The output of the defuzzifier stage (see Figure 7) will be a numeric value that represents the total assessment score that is used to determine if the SAAS meta-architecture architecture concept shown in Figure 1a is acceptable or unacceptable.

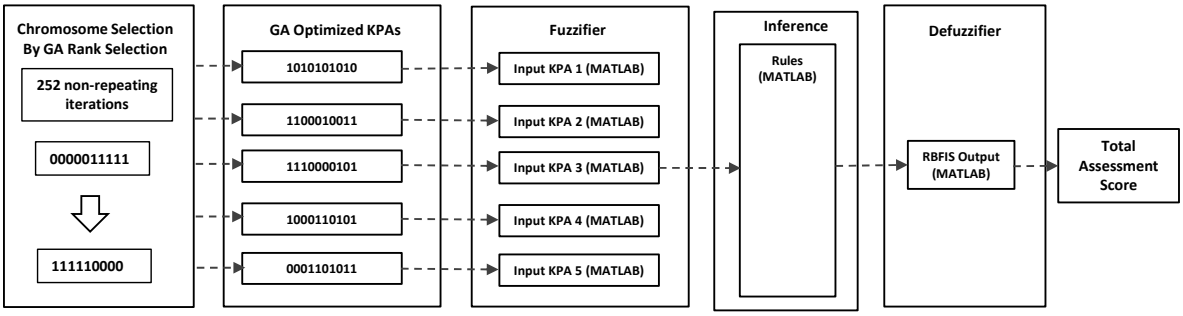


Figure 7. SAAS SoS meta-architecture assessment model using GA optimized KPAs

3. Application of the Proposed Model

The SAAS SoS meta-architecture model (see Figure 7) utilizes a GA to determine the best KPAs to serve as input variables to the MRBFIS. The membership functions and the If-Then rules of the MRBFIS allow for an adjustable nonlinear output that provides a customized assessment based on the systems architects expectations. The model is capable of assessing the SAAS SoS meta-architecture concept (see Figure 1a) once numeric assessment values are determined for the five KPA inputs to the MRBFIS. An arbitrary value of 0.80 (80%) or above is expected for the SAAS SoS architecture concepts total score to be considered as acceptable architecture. It should be noted higher or lower total assessment scores could also be selected if desired by the systems architect. Arbitrary values were given for KPA1 though KPA5 inputs. The following example as shown in Figure 8, demonstrates a fuzzy assessment of the SAAS SoS meta-architecture using the MRBFIS and the KPA1 though KPA5 (MRBFIS inputs) being equal to 0.89, 0.84, 0.58, 0.89, and 0.95 respectively:

MATLAB® code to run:

```
fismat = readfis('SAASa2');
out1 = evalfis([.89 .84 .58 .89 .95],fismat)
out1 = 0.8167
```

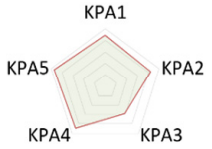
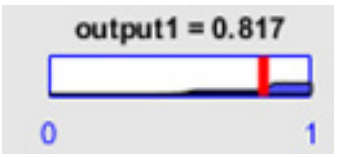
INPUT				OUPUT
KPA Kiviat Chart	KPA	KPA Name	FIS Input	MATLAB Mandani FIS Output
	1	Reliability	0.89	
	2	Controlability	0.84	
	3	Modularity	0.58	
	4	Affordability	0.89	
	5	Flexibility	0.95	

Figure 8. SAAS SoS meta-architecture KPA assessment using MATLAB® code.

The SAAS SoS meta-architecture concept (see Figure 1) assessment output of the MRBFIS is 0.817 as shown above in Figure 8. The SAAS SoS meta-architecture concept as shown in Figure 1a would be acceptable as achieving an assessment score of 81.67% with above 80% being acceptable. The MRBFIS If-Then rule based structure allowed for the architecture to score in high range; even though the KPA input variable Modularity was in the medium range.

The MRBFIS can also determine which KPAs have a greater influence in the architecture assessment. For example, if the MRBFIS inputs of KPA1 and KPA3 were interchanged, then the MRBFIS output would be 0.739, which is an unacceptable architecture score of 73.9%. An average of the five KPA inputs shown in Figure 8 would be 0.83 or 83%. The MRBFIS has the advantage of enabling a nonlinear fuzzy rule based assessment output that more adjustable than a sum or average of the KPA inputs.

4. Conclusions and Future Work

After the customer needs have been assessed and the SoS meta-architecture analysis phase is initiated, it is necessary to create a tangible concept that effectively fulfils all of the stakeholder requirements. The architecture is likely to evolve and improve throughout the concept generation stages after an assessment is made. The GA rank selection of KPAs and the utilization of a MRBFIS can provide clarity on the assessment of SoS meta-architecture concepts and also provide a formal method of assessing SoS capability attributes as they are understood at the time.

The assessment model as described in this paper and shown in Figure 7 is used to assess the SAAS SoS meta-architecture (see Figure 1a) at the early stages of development prior to program commitment and funding. The MRBFIS, when used as the final stage of an architectures assessment method, enables a nonlinear output that is more adjustable than existing assessment methods. This model may prove to be beneficial during the preliminary tradespace exploration stages of the conceptual architecture prior to program development. This assessment model may also help prevent costly project failures, system ambiguity late in development, unacceptable risks, cost overrun, and the inability to develop critical systems.

Future work shall research formal methods of calculating the rank multiplier (see Figure 3) for each KPA prior to using the GA rank selection method for the optimum chromosome selection. A detailed method of calculating individual KPA FIS input scores (as shown in Figure 8) will also be researched. Assessing the individual KPAs may involve the use of a combination of various systems engineering tools. A formal method of individual KPA assessment may include the utilization of the risk assessment matrix (RAM), the design structure matrix (DSM), quality function deployment (QFD), and other assessment methods found in the systems engineering toolbox [6].

References

1. Pape L, Agarwal S, Dagli C. *Selecting Attributes, Rules, and Membership Functions for Fuzzy SoS Architecture Evaluation*. Procedia Computer Science, Volume 61, 2015, pp. 176-182.
2. De Weck OL, Roos D, Magee CL. *Engineering Systems: Meeting Human Needs in a Complex Technological World*. Massachusetts: MIT Press, 2013
3. Dou K, Wang X, Tang C, Ross A. *An Evolutionary Theory-Systems Approach to a Science of the Ilities*. Procedia Computer Science, Volume 44, 2015, pp. 433-442.
4. Renault A. *A Model for Assessing UAV System Architectures*. Procedia Computer Science, Volume 61, 2015, pp. 160-167.
5. Rodano M, Giammarco K. *A Formal Method for Evaluation of a Modeled System Architecture*. Procedia Computer Science, Volume 20, 2013, pp. 210–215.
6. Goldberg BE, Everhart K, Stevens R, Babbitt N, Clemens P, Stout L. *Systems Engineering "Toolbox" for Design Oriented Engineers*. NASA Report 1358, MSFC, December 1994.
7. Nguyen CH, Pedrycz W, Duong TL, Tran TS. *A genetic design of linguistic terms for fuzzy rule based classifiers.* International Journal of Approximate Reasoning 54, 2013, pp. 1–21.
8. Rudziński F. *A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifier.* Applied Soft Computing 38, 2016, pp. 118–133.
9. Kumar R, Jyotishree M. *Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms.* International Journal of Machine Learning and Computing, Volume 2, 2012, pp. 365-370.
10. Pape LE, Agarwal S, Dagli CH, Ergin NK, Enke D, Gosavi A, R. Qin, Wang R, Gottapu RD. *Flexible and Intelligent Learning Architectures for SoS Architectural Evolution in Systems-of-Systems*. Procedia Computer Science, Volume 44, 2015, pp. 76-85.
11. Adler CO, Dagli CH. *Study of the Use of a Genetic Algorithm to Improve Networked System-of-Systems Resilience*. Procedia Computer Science, Volume 36, 2014, pp. 49-56.
12. Bartolomei JE, Hastings DE, De Neufville R, Rhodes DH. *Engineering Systems Multiple-Domain Matrix: An Organizing Framework for Modeling Largescale Complex Systems.* Systems Engineering 15.1, 2012, pp. 41–61.
13. Sharon A, De Weck OL, Dori1 D. *Model-Based Design Structure Matrix: deriving a DSM from an Object-Process Model.* In Second International Symposium on Engineering Systems, 2009, pp. 1-12.
14. Burge S. *A Functional Approach to Quality Function Deployment*. BHW and the Systems Engineering Company. Technical Paper 0001/sb, 2007.
15. Verma R, Maher T, Pullman M. *Effective Product and Process Development Using Quality Function Deployment.* Cornell University, Article 561, 1998.