



Missouri University of Science and Technology
Scholars' Mine

Engineering Management and Systems
Engineering Faculty Research & Creative Works

Engineering Management and Systems
Engineering

01 Nov 2015

Selecting Attributes, Rules, and Membership Functions for Fuzzy SoS Architecture Evaluation

Louis Pape

Siddhartha Agarwal

Cihan H. Dagli

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

Follow this and additional works at: https://scholarsmine.mst.edu/engman_syseng_facwork

 Part of the [Operations Research, Systems Engineering and Industrial Engineering Commons](#)

Recommended Citation

L. Pape et al., "Selecting Attributes, Rules, and Membership Functions for Fuzzy SoS Architecture Evaluation," *Procedia Computer Science*, vol. 61, pp. 176-182, Elsevier, Nov 2015.

The definitive version is available at <https://doi.org/10.1016/j.procs.2015.09.187>



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License](#).

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Engineering Management and Systems Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.



Complex Adaptive Systems, Publication 5
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2015-San Jose, CA

Selecting Attributes, Rules, and Membership Functions for Fuzzy SoS Architecture Evaluation

Louis Pape*, Siddhartha Agarwal, Cihan Dagli

Missouri University of Science & Technology, Rolla, MO, 65409, USA

Abstract

The development of the FILA-SoS meta-architecture approach to acknowledged systems of systems (SoS) analysis allows a relatively unbiased method for exploring a potential SoS architecture space. This paper delves more deeply into 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 shapes that work well. A wide range of options exist for all the individual elements of SoS assessment. Some recommendations for finding an appropriate combination for the adjustable parameters of fuzzy assessment models through random architecture chromosome testing and iteration are developed.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

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

Keywords: meta-architecture; systems of systems; acknowledged; fuzzy genetic; assessment

1. Modeling Systems of Systems (SoS)

1.1. Acknowledged SoS background

Many important features of the current state of civilization may be characterized as systems of systems (SoS). These include trade, infrastructure such as power, water, fuel, rail, or road systems, intermodal transportation, etc. A restricted segment of this class of systems is Acknowledged SoS. Acknowledged SoS involve mostly voluntary interactions between systems with an independent existence, i.e., these systems would continue even if they do not participate in the SoS, and a SoS Manager with persuasive authority and small budget [1]. Although there are typically advantages to joining the SoS, there are also costs and increased responsibilities for the systems.

* Corresponding author. Tel.: 1 314-233-5782; fax: 314-232-8847
E-mail address: lep7df@mst.edu

Component systems provide subsets of capabilities used to build up the desired overall SoS capability. Normally the SoS Manager has several choices among related systems from which to compose the SoS, and negotiates among them to create the SoS. The SoS typically evolves over time, as systems and their capabilities improve or decline. The SoS Manager normally has the opportunity to re-negotiate as the SoS requirements, the systems, or the environment changes [2].

The types of SoS being discussed here frequently include systems from differing domains, such as maritime and aeronautical, or different military services. Therefore it is extremely important to reach agreement on the definition of terms to be used within the SoS, or translations between commonly used terms when crossing boundaries between domains. The concept of operations for combining systems' capabilities to achieve the SoS purpose must likewise be agreed to among all the systems.

Acknowledged SoS rarely develop a new system; they almost always modify existing systems. Modifications are less expensive and faster to develop than new systems. The modified systems are expected to work together, i.e., interface, in a newly enabled way to achieve the unique SoS capability. When modifying component systems, it is unlikely that all systems (of a type) get the modification at the same time. The basis of the type of SoS being considered here is a variable number of systems of several subtypes with various capabilities, partially interfaced among themselves to achieve a powerful SoS capability not easily achieved in other ways. A simplified SoS meta-architecture consisting of a binary string indicating the presence or absence of the various systems and their mutual interfaces has been used.

1.2. The meta-architecture

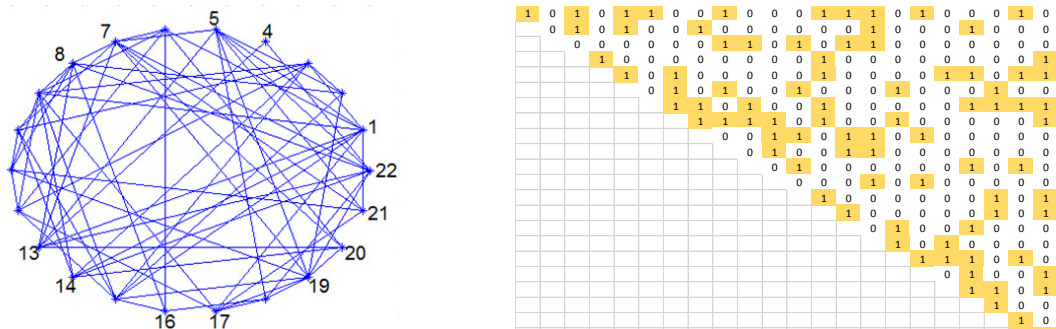


Figure 1. a) Network diagram of SoS systems and interfaces b) Upper triangular matrix representation of SoS architecture chromosome

The simple presence/absence representation within the SoS of systems (numbered if present in Figure 1a where interfaces are lines between the systems; or systems as shaded ones along the diagonal and interfaces in the off-diagonal elements in Figure 1b) is combined with cost data as discussed above (or sub capability performance data, or more complicated programs for evaluating each of the attributes as discussed in the next paragraph) to arrive at measures of each attribute.

1.3. Selecting SoS evaluation attributes

Important attributes of any SoS almost always include at least 1) performance in the desired, top level capability of the SoS, and 2) the cost (or inversely, affordability) of the project. "Costs" might also be incurred through the use of any scarce commodity such as bandwidth, message size or latency in a time critical application. Commonly identified key attributes frequently include something akin to robustness or resilience to loss of some systems from the SoS on any particular occasion when it is called to action. Availability, reliability, modularity, ability to self-organize, flexibility to altered purposes, responsiveness, versatility, agility, resilience, adaptability, and net-centricity are some of the attributes, defined in various ways by various authors that have been used as desirable

attributes of a SoS [3] [4] [5]. In keeping with the extremely simple SoS models used in these analyses, it is proposed that only attributes whose definition depends strongly on the meta-architecture be considered. One is certainly free to include attributes that do not depend on the architecture, but they will not effectively aid in distinguishing among different instances of the meta-architecture.

It is easy to posit a series of system types with dollar values per hour of operation. Examples include aircraft, helicopters, unmanned aerial vehicles, ships, command posts guard posts, etc. The development and testing of each new interface between similar or dissimilar systems would also have an estimated cost. The actual interfacing equipment, such as communication systems and antennas, even the installation of updated software in a fleet of vehicles, has a cost which may be estimated. The SoS Manager should bear, or at least defray, most of the direct costs of participation in the SoS. The systems should agree to provide the required capability to the SoS when requested.

If, for example, reliability is selected as a key SoS attribute, then reliability data (or at least estimates) must be had for each type of system. Appropriate reliability equations for the numbers and types of each system (perhaps including the interfaces) present in the architecture must be developed to produce a SoS reliability for any combination of the component elements as represented in the architecture chromosome. These equations would have to account for the serial or parallel combinations of the systems' capabilities within the meta-architecture. This tailoring of the attribute evaluation algorithm must be done to some extent even if the context of a different SoS is very close to an existing example, and even though many patterns may be reused.

A similar general form for evaluating the availability must be developed to account for any systems (with the capabilities they must have available) or interfaces represented as present in the architecture, from zero ones to 100% ones. A modularity measure may be found by a spectral analysis of the upper triangular matrix [6] or another method [5]. Each attribute may have an evaluation measure technique which is unique to the SoS context. The measurement technique need not be a closed form equation. Algorithm as simple or as complex as required, including if statements, case statements, possibly even re-entrant code, thereby changing parameterization for various thresholds in numbers of systems or interfaces, may be used to evaluate individual attributes of the SoS. Membership function shapes may also be selected to match the problem context better [6].

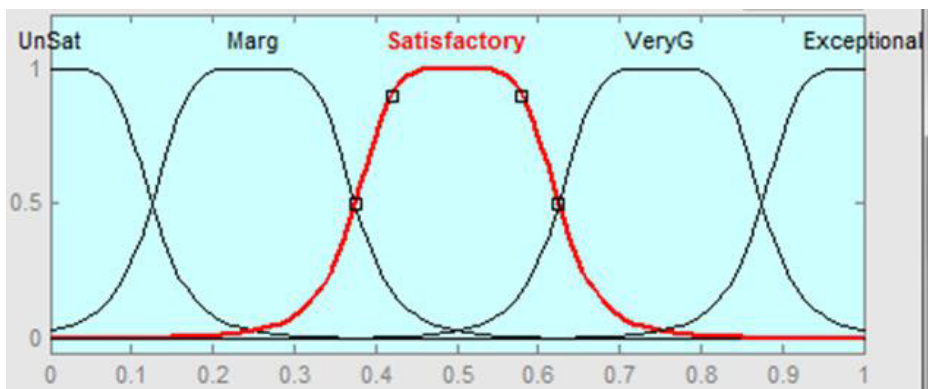


Figure 2. Gaussian trapezoidal fuzzy membership function example

1.4. Fuzzy evaluations of attributes

Stakeholders, SoS Managers, and modelers are usually satisfied with characterizing small regions around several points within the entire range of values of the attributes with indifference. For this reason, trapezoidal fuzzy membership functions are easy to justify and explain as shown in Figure 2. Other shapes might be used for special

reasons, but for project reviewer understanding, the trapezoid seems to resonate. US government acquisition reviews give these regions color evaluations such as red, green, gold, etc. [6]. The selection of the number of membership functions, whether they are shown in real or scaled units, and whether they are of equal size is also a matter of SoS context. When displaying results to stakeholders, real units seem to work better; when using them with modelers, scaled units sometimes work better for comparing differing models. Identifying the crossover points between adjacent membership functions is adequate to uniquely identify them once the general shape is chosen. The number of membership functions to use depends on the potential achievable precision and accuracy of the measures, the degree of precision desired across the whole potential range of the measurements by the stakeholders, as well as on the precision of the input data estimates. Another consideration is that if some subjective component, such as a Likert scale, is part of an attribute, does one want to give a responder the ability to select the middle of an *odd* number of choices, or force at least a better than, or worse than, average choice, through an *even* number of membership functions.

1.5. Rules for combining evaluations to an overall SoS assessment

Having selected a meta-architecture with a numbers of each type of system, a SoS capability buildup process from the systems' sub-capabilities, performance and other attributes identified along with the way to calculate them, the only thing remaining is how to combine the attribute valuations into an overall SoS assessment for any given architecture. A series of fuzzy inference system rules in the form "IF attribute 1 is in this range, AND/OR attribute 2 is in this range, and/or other attributes are either IN this range or NOT in that range (or ignored)... THEN the SoS takes on THIS overall assessment." The Matlab Fuzzy Toolbox™ allows the construction of this kind of rule based decision system quickly and easily with its graphical user interface. Rules take the form "If ANY attribute is Unacceptable, then SoS is Unacceptable," "If ALL the attributes are Marginal, then the SoS is Unacceptable," etc. Rules of this form in a fuzzy inference system transform a multidimensional objective optimization problem using all the attributes to a single variable optimization using the output of the fuzzy inference system (or fuzzy associative memory) for the SoS assessment. The fuzzy assessor is used to determine the fitness of any meta-architecture instance for a genetic algorithm, or a linear solver algorithm, to find a 'good' architecture to propose for negotiation in the later steps of FILA-SoS [2]

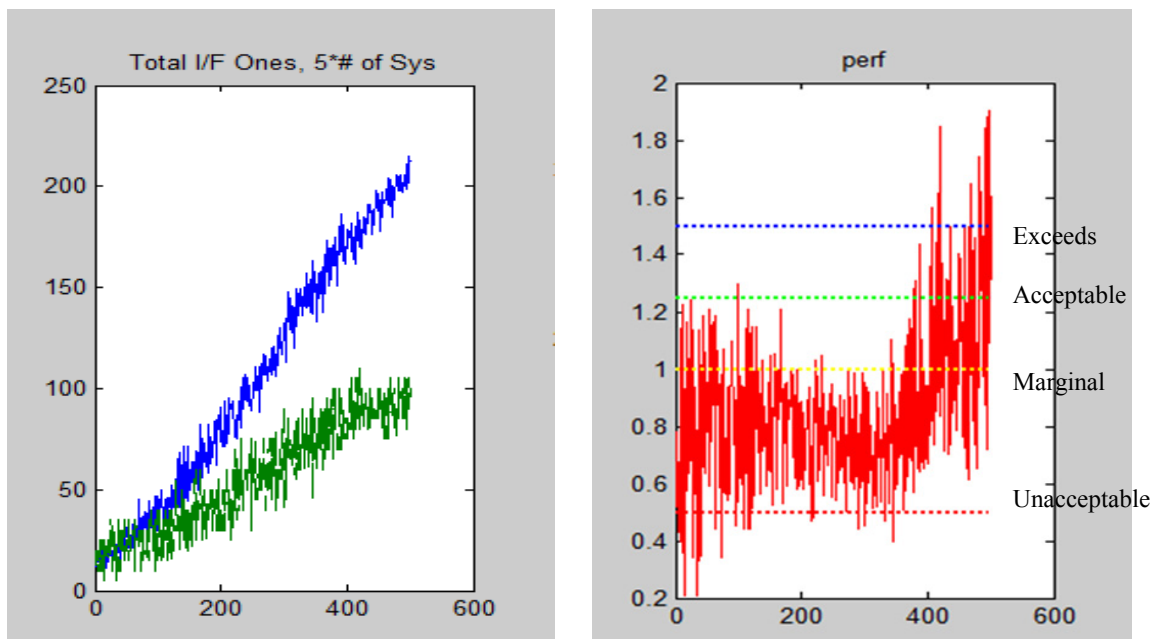


Figure 3. a) Number of ones in a Population of 500 chromosomes b) Performance attribute of each chromosome, with membership function crossover lines marked with colored dotted lines

2. Exploring the meta-architecture model

2.1. Attribute evaluations

The trick to exploring the SoS assessment model is to treat the number of ones in the meta-architecture chromosome as the independent variable. The number of ones in the chromosome may be varied by biasing the random bit filling of the chromosome toward either few or many ones. Build a population of n random chromosomes by filling successive chromosome bits with a one, if a random variable uniformly distributed from 0 to 1 is less than or equal to k/n , where k is the chromosome number within the population (otherwise a zero bit). Low numbered chromosomes in this trial population will have few ones, and higher numbered chromosomes of this population will have many ones. An example is shown in Figure 3a, where the number of ones in the interface region of the chromosome is plotted in blue, with five times the number of systems (so they can be shown on the same scale) plotted in red.

Plotting the evaluation of each chromosome for one of the attribute algorithms, shown in Figure 3b, allows one to confirm that the overall shape is at least close to something that was expected. Furthermore, showing the regions of the membership functions in Figure 3b at the same time allows one to confirm that the expectations are achieved with this attribute model. If all the chromosome evaluations cluster in only one or two membership function regions, then at least one (possibly more) of three things must be done: 1) adjust the membership function edges to be more discriminatory, given the results of the evaluation, 2) adjust the input data that feeds the evaluation algorithm for this attribute, or 3) adjust the algorithm so that it allows better differentiation of the attribute between chromosomes. Changes in any of these areas must be coordinated with the stakeholders. This is not as onerous as it might be imagined, because they usually want to agree on *something*, to be able to move forward. Even if some stakeholders don't particularly like some of the SoS plans or designs, they usually don't wish to be left behind, with no voice at all, as may happen if they disagree with every proposed model tune-up.

2.2. Combining attributes to the overall assessment

The fuzzy inference systems rules allow combining the individual attribute values for each chromosome to the overall assessment of the SoS architecture represented by the chromosome. It is frequently seen that multiple objectives are plotted in side by side bar charts or Kiviati charts as shown in Figure 4 or Figure 5. Even when the comparisons are properly all in one direction as in Figure 5, it can be difficult to decide which alternative is better without knowing the relative value between attributes. The fuzzy inference system explicitly allows one to say that more affordability is worth more than any other attribute. Combined with the freedom to choose the membership function shapes and crossover points, this allows a mathematically precise method to decide which of any two closely related architectures is the better, according to the rules laid down in the fuzzy inference system.

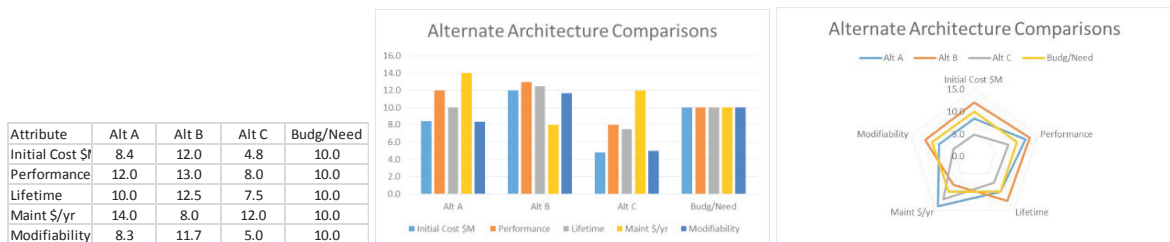


Figure 4. Relative architecture comparisons on a scale of 10 as the desired value; but some attribute values are better smaller

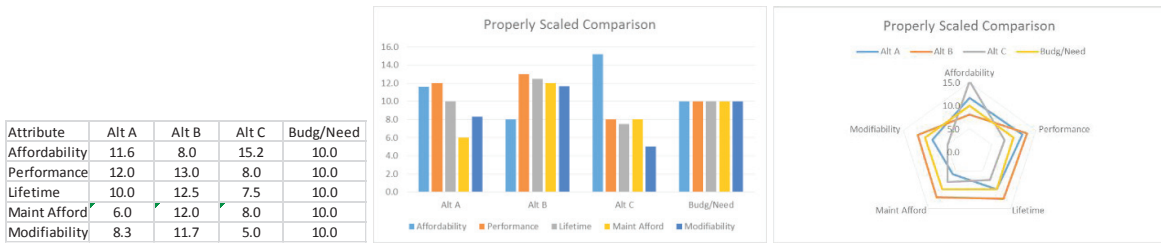


Figure 5. Properly scaled architecture comparison where all attributes are better when larger; they are still difficult to decide which is best

Plotting many SoS architecture chromosomes over a wide range of numbers of ones, along with each of the attribute evaluations, including the membership function edges, as seen in Figure 6, with the SoS assessment for the given rule bases, shows that some good chromosomes exist, gives confidence that the model is self-consistent and will produce a reasonable optimum for the starting recommendation to the negotiation process. Being self-consistent does not mean the model is necessarily correct, only that it is not *so* wrong that one need not look deeply to find serious flaws.

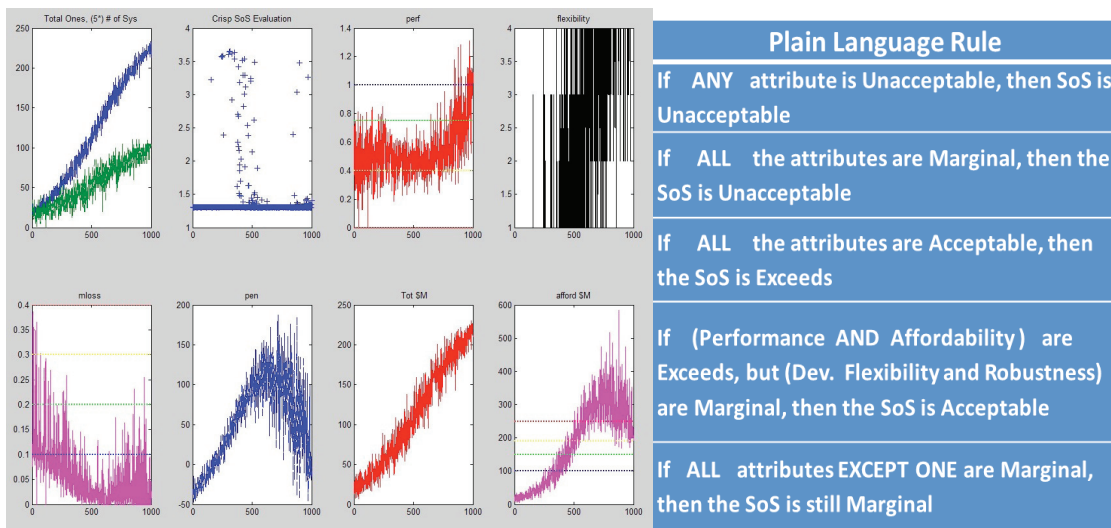


Figure 6. Plot of SoS assessment (second graph) and all attributes with membership function regions, and fuzzy inference system rules

3. Conclusions

Methods of identifying attributes to assess competing architecture alternatives from within a first order binary meta-architecture model from a priori knowledge and stakeholder discussions are suggested. A justification for trapezoidal membership functions, and an even number of MFs, with an example from existing acquisition policy was given. Exploration, by testing a large range of randomly generated chromosomes, helps determine if the model adequately represents the stakeholders’ concerns. Adjustments to each of the elements of a fuzzy inference system may be made to improve its value for selecting among meta-architecture instances. Exploration also helps assure that the model is sufficiently discriminatory to make a good choice among alternatives. Comparison with bar and Kiviart charts helped explain some problems with using those architecture representations for decision making. These tips and tricks should help practitioners design architecture models and better understand how to use them to compare or select among alternatives in the future.

Acknowledgements

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. The SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Department of Defense.

References

- [1] Director Systems and Software Engineering, OUSD (AT&L), “Systems Engineering Guide for Systems of Systems,” available from <http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf>, 2008.
- [2] L. E. Pape, S. Agarwal, C. H. Dagli, N. K. Ergin, D. Enke, A. Gosavi, R. Qin, R. Wang and R. D. Gottapu, “Flexible and Intelligent Learning Architectures for SoS (FILA-SoS): Architectural Evolution in Systems -of- Systems,” in *Conference on Systems Engineering Research*, Hoboken NJ, 2015.
- [3] D. S. Alberts, *The Agility Advantage: A Survival Guide for Complex Enterprises and Endeavors*, Washington DC: Center for Advanced Concepts and Technology, 2011.
- [4] B. Mekdeci, N. Shah, A. M. Ross, D. H. Rhodes and D. Hastings, “Revisiting the Question: Are Systems of Systems just (traditional) Systems or are they a new class of Systems?,” *Systems Engineering Advancement Research Initiative (SEARI)*, Cambridge, MA, 2014.
- [5] S. Sarkar, A. Dong, J. A. Henderson and P. A. Robinson, “Spectral Characterization of Hierarchical Modularity in Product Architectures,” *Journal of Mechanical Design*, 2013.
- [6] Srinivas Sethi, Anupama Sahu, Suwendu Kumar Jena, “Efficient load Balancing in Cloud Computing using Fuzzy Logic,” *IOSR Journal of Engineering*, 2012.