Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 61 (2015) 168 – 175
Procedia Computer Science

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

Application of Value Focused Thinking and Fuzzy Systems to Assess System Architecture

Gene Lesinski*

United States Military Academy, West Point, New York 10996, USA

Abstract

Since a majority of resources are obligated during the design phase of a system lifecycle, critical assessment of candidate functional and system architectures is vital to identify optimal architectures before proceeding to subsequent lifecycle phases. Common challenges associated with generation and evaluation of system functional architectures include search of the expansive design space and assessment of key performance attributes that are particularly "fuzzy" and qualitative in early architecture development. Several assessment approaches have been presented in the literature to address the assessment challenge to include Quality Function Deployment (QFD), Analytical Hierarchy Process (AHP), Value-Focused Thinking (VFT), and fuzzy logic. In this research we combine the use of value functions and fuzzy assessment to assess a functional and system architecture. There are several benefits of a methodology that combines value-focused thinking and fuzzy assessment. A distinct advantage of the methodology presented is the explicit inclusion of the customer in the assessment process through validation of the TPM value functions Involving the customer in TPM value function development and validation ensures the customer has direct input regarding the TPMs and their associated value across the range of discourse The methodology presented is flexible enough to assess architectures early in the process when things are "fuzzy" as well as later when subsystem and component performance are well defined. The methodology can also be used to analyze and assess impacts of interface changes within the system architecture. . The methodology is domain independent and can be coupled with executable models linked to scenarios. The assessment methodology is applied to the architecture for a soldier knowledge acquisition system for which the key performance attributes are affordability, performance, flexibility, updateability, and availability.

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

Keywords: Architecture; Assessment; Fuzzy Systems; Value Focused Thinking

---

* Corresponding author. Tel.: 8457-729-3677.
E-mail address: Eugene.Lesinski@usma.edu

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology
doi:10.1016/j.procs.2015.09.182

1. Introduction

A systems architect transforms a customer need into an initial conceptual solution and iteratively refines this concept to an appropriate level of detail to communicate with design engineers and the customer. Since a majority of resources are obligated during the design phase of the system lifecycle, selection the optimal architecture at each stage of system design is critical prior to proceeding to the next architectural level of detail. Architecture depicts the concepts, functions, components, and interfaces of the proposed system. Architecture development requires the hierarchical reduction of ambiguity through the iterative application of scoping, aggregating, partitioning, integrating, and certification of the system architecture. Common challenges associated with generation and evaluation of system functional architectures include search of an expansive design space and assessment of key performance attributes that are particularly “fuzzy” and qualitative in early architecture development. Additionally, architecture generation and assessment are typically domain specific tasks with numerous valid alternative approaches. Several architecture generation and assessment approaches have been presented in the literature to address these challenges.

This paper presents a methodology to assess functional and system architecture which employs customer validated Technical Performance Measure (TPM) value functions coupled with fuzzy assessment of key performance attributes to distinguish among architecture alternatives. A value function is generated for each TPM with customer input and calibration. TPM values that support a particular Measure of Performance (MOP) are combined to give an overall MOP score. MOPs that support a particular Measure of Effectiveness (MOE) are combined to give an overall MOE score. MOEs that support a particular Key Performance Attribute (KPA) are combined to give an overall KPA score. At the highest level within the assessment hierarchy are the customer provided KPAs. The KPAs are characterized by a degree of uncertainty therefore, there is a degree of fuzziness associated with each factor as well as their outcome or system architecture assessment score. MATLAB Fuzzy Toolbox (MathWorks 2014) is used to program the fuzzy assessment framework. The combined effect of these attributes is the system architecture rating. The outputs of the assessment methodology are a kiviat chart, score for each KPA, and an overall architecture score. The methodology is flexible enough to assess architectures early in the process when things are “fuzzy” as well as later when subsystem and component performance are well defined.

A soldier knowledge acquisition system is used as the case study for this framework. The soldier knowledge acquisition system was assessed using the key performance attributes affordability, performance, flexibility, updateability, and availability. Functional hierarchies and associated trees of attributes were key inputs to the assessment methodology. The assessment methodology produced a kiviat chart, scores for each KPA, and an overall architecture score indicating that the baseline functional architecture was superior to one that combined selected functions and requirements. The same model was used to assess a baseline system architecture and an alternative which modified selected system interfaces. The methodology was able to distinguish and assess the value of these competing architectures and highlighted the increased value of the selected interfaces.

1.1. Previous Research

Numerous techniques have been presented in the literature to effectively generate architecture alternatives across the vast design solution space. Acheson, Dagli, and Kilicay-Ergin [1] utilize agent based modeling to iteratively develop an acknowledged System of Systems (SoS) and its potential member systems. The System agents possess attributes that reflect the funding, priorities, capabilities, and behavior of the individual systems. A fuzzy inference engine mechanism is used to represent a systems willingness to evolve, acceptance of future vision, ability to change, and ability to influence change. Genetic algorithms (GA) have been used to represent and explore SoS architecture design alternatives. Architecture functions, components, and interfaces are encoded as a binary strings with accompanying algorithms (reproduction crossover, and mutation) designed to search the solution space.

Haris and Dagli [4] employ a GA to generate architecture alternatives and fuzzy associative memory to assess architecture effectiveness/suitability. The system KPAs were derived using the QFD technique with customers prioritizing KPAs and the architects ranked the candidate physical components with respect to each KPA. A set of Fuzzy rules were developed to assess the competing architectures. The relative importance of the rules was originally assigned by the customer but was later randomly varied to replicate environmental factors and adaptability. Pape [7] combines the use of Genetic Algorithm representation of SoS architecture with Fuzzy SoS evaluation. Possible SoS meta-architectures architectures are depicted by a chromosome consisting of a systems participation and its interfaces with other systems. Appropriate mathematical equations are derived to evaluate each attribute. Membership functions, accompanied by a classification rubric, are developed to map attribute scores to the fuzzy categories of: unacceptable, marginal, acceptable, and exceeds.

Giammarco [3] utilizes conceptual model data to assess domain- independent architecture by using patterns to identify both desirable and undesirable architecture patterns within the architecture. The architecture is first restructured using the Lifecycle Modeling Language (LML) in terms of Assets, Actions, Inputs/Outputs, and Conduits. Observations from the restructured model are used to generate a series of domain-independent axioms that can be used, share, or distributed.

Singh and Dagli [9] present an evolutionary algorithm architecture design space search coupled with a fuzzy architecture assessment. They decompose the Architecture Quality Attributes into a tree of sub-attributes with the lowest level representing a unique architecture strategy. Combinations of architecture strategies are generated then evaluated by a fuzzy assessment. The overall architecture rating is calculated by inputting each architecture attribute score and cost score into a fuzzy inference system which maps these inputs to an associated architecture acceptability rating.

Purewal, Yang, and Grigg [8] present a framework for assessing architectures of embedded systems utilizing a multi-criteria decision making (MCDM) approach that employs Evidential Reasoning (ER). Each attribute is of a system is described by a distributed assessment using a belief structure. The ER approach can handle several types of uncertainty to include: missing data, incomplete attribute descriptions, probabilistic attribute behavior, and fuzzy attribute performance categorization. All attribute values or grades are mapped to utility functions on a common scale which facilitates aggregation to create a total systems architecture effectiveness score.

Dojutrek, Labi, and Dietz [2] use a fuzzy approach to assess transportation infrastructure security. Overall security assessment is formulated as a function of three major factors: Threat Likelihood, Infrastructure Resilience, and Consequence. A supporting hierarchy is established with Factors supported by Measures and Measures supported by Attributes. The measures are input into a fuzzy inference system to calculate a fuzzy degree for their associated factor. Each fuzzy Factor score was input into a fuzzy security rating system which resulted in an overall security rating.

Kenney [6] employs Value-focused thinking (VFT) to place customer-derived values at the center of decision-making. Detailed stakeholder analysis is used to reveal the fundamental elements of value to a customer. The value measures are arranged in a hierarchy with the value measures at the lowest level. Value measures are subordinate to objectives. Objectives are subordinate to functions and functions are subordinate to a fundamental objective. Individual value measures are mapped to value functions which translate raw attribute or parameter values to a dimensionless scale to provide an equitable basis for tradeoff analysis and facilitate mathematical manipulation on a common scale. Alternatives are generated across the value space then scored according to the degree of satisfaction across all value measures.

The KPAs associated with system architectures are "fuzzy" in that exact distinctions or boundaries between attributes values like low, medium, and high are difficult to define. Fuzzy systems provide an ability to reason with ambiguous knowledge and measurement. Fuzzy sets measure the degree to which an element belongs to a set. Rather than strict membership to each attribute category, a fuzzy set allows partial membership. Fuzzy membership functions (triangular, trapeziodal, etc.) are created to represent each attribute value across its range of discourse. Fuzzy rules are created to translate and document fuzzy set input combinations with resulting fuzzy output sets. A fuzzy associative memory (FAM) encodes a compound set of rules that associates multiple input fuzzy sets with one or more output fuzzy sets. A fuzzy inference system allows mathematical manipulation of the fuzzy input sets to translate them to an assessed output value.

In this research we combine the use of value functions with fuzzy assessment to assess a functional and system architecture. There are several benefits of a methodology that combines value-focused thinking and fuzzy assessment. Involving the customer in technical performance measure (TPM) value function development and validation ensures the customer has direct input regarding the TPMs and their associated value across the range of discourse. The methodology can be easily replicated with basic Excel and MATLAB tools. Unlike other methods that require different assessment techniques at each design level of abstraction, the methodology presented is flexible enough to assess architectures early in the process when things are "fuzzy" as well as later when subsystem and component performance are well defined.

2. Proposed Methodology

A need statement, key performance attributes, and scenarios are essential inputs to architecture generation. These inputs are then transformed into a functional architecture and supporting tree of attributes. The functional architecture reflects the aggregation and partitioning of system requirements and functions. The tree of attributes depicts the hierarchical arrangement of key performance attributes (KPAs), measures of effectiveness (MOE), measures of performance (MOP), and technical performance measures (TPMs). The functional architecture and the associated tree of attributes are the inputs to the architecture assessment.

Figure 1 highlights the architecture assessment methodology. A value function is generated for each TPM with customer input and calibration. Value functions are strictly increasing or decreasing functions that convert raw TPM values into a dimensionless scale from 0-100 so that scores can be combined mathematically. TPM values that support a particular MOP are combined to give an overall MOP score. MOPs that support a particular MOE are combined to give an overall MOE score. MOEs that support a particular KPA are combined to give an overall KPA score. At the highest level within the tree of attributes are the customer provided KPAs. The combined effect of these attributes is the system architecture rating. It is assumed that all KPAs are equally weighted. However, the assessment model can be adjusted to account for stakeholder KPA weighting factors. The KPAs are characterized by a degree of uncertainty therefore, there is a degree of fuzziness associated with each factor as well as their outcome or system architecture assessment score. MATLAB Fuzzy Toolbox (MathWorks 2014) is used to program the fuzzy assessment framework. A membership function is generated to assign a degree of membership for each KPA over its range of discourse. Figure 3 illustrates a generic KPA membership function. The membership functions can have numerous shapes to include triangular, trapezoidal, or trigonometric. Lastly, a set of fuzzy rules (See Table 1) are developed that characterize the overall system architecture based upon the combined characterization of KPAs. The two outputs of an architecture assessment are a kiviat chart and an overall architecture score. The kiviat chart is a graphical depiction that reflects architecture satisfaction or performance against each KPA. The architecture score is a fuzzy assessment combination of all KPAs for a particular architecture and therefore reflects the overall architecture effectiveness. This score provides a means to compare alternative architectures.

Table 1. Example Fuzzy Assessment Rules.

Rule 1	If KPA1 is Unacceptable and KPA2 is Unacceptable then the System is Unacceptable
Rule 2	If KPA1 is Excellent and KPA2 is Unacceptable then the System is Marginal
Rule 3	If KPA1 is Unacceptable and KPA2 is Excellent then the System is Marginal
Rule 4	If KPA1 is Excellent and KPA2 is Excellent then the System is Excellent

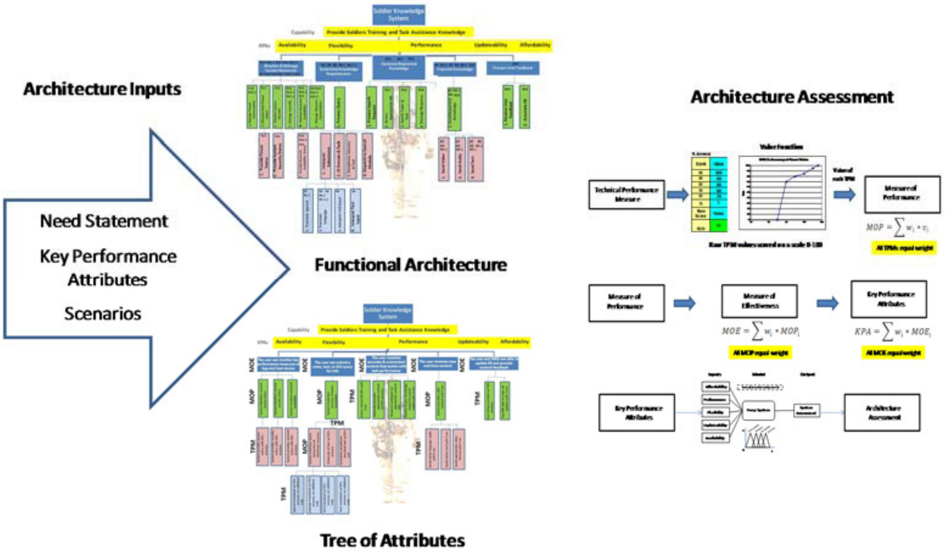


Fig. 1. Architecture Assessment Inputs and Assessment Methodology.

3. Case Study

The Army faces the future challenge of reduced soldier cognitive ability while equipment and tasks become increasingly complex. The Army currently maintains a large collection of print and digital references for training, doctrine, and equipment, however the volume of these printed materials make it impractical for rapid soldier access and reference. Additionally the content of these references may not match the cognitive abilities of current or future soldiers. In this research, we present a method to assess architectures that provide soldiers the capability to acquire task and performance assistance knowledge. The methodology employs a hierarchical application of value focused thinking and fuzzy systems which is flexible enough to assess architectures early in the process when things are "fuzzy" as well as later when subsystem and component performance is well defined. The customer need statement for this case study is highlighted below.

> *Provide a flexible framework that can reduce the cognitive load imposed by increasingly complex technology and situations to more appropriately match the available mental resources of individuals to execute both routine and complex military tasks across several Military Occupational Specialties (MOS).*

The KPAs for this case study are defined in Table 2 and include affordability, performance, flexibility, updateability, and availability. The fundamental capability provided is the transfer of training and task assistance knowledge to soldiers.

3.1. Operational setting and scenarios

Operational Setting: An engineer soldier is on a route reconnaissance mission. During the route reconnaissance mission he is required to identify obstacles to vehicular movement, classify road surfaces, and bridges. During the route reconnaissance he comes across a bridge. A typical engineer soldier task is to "classify" the bridge in order to highlight what equipment in the organization can safely cross the structure. There is a detailed process and calculations for bridge classification. However, this soldier has not "classified" a bridge in a long time and is therefore, very rusty on this task. He could use instructions, video, or a quick "how to" to help jog his memory on performing this task.

Table 2. KPAs for Soldier Knowledge Transfer

Affordability	An evaluation of the cost of the system compared to the current process.
Performance	An evaluation of content provided to the user based upon the reading level of the content and whether the content provided is in compliance with established doctrine, techniques, tactics, and procedures.
Flexibility	An evaluation of the system ability to perform in garrison, field, and deployed environments, support both training and task assistance, with ability to expand to support multiple MOSs.
Updateability	An evaluation of the ease and ability of many SMEs to access and provide Knowledge Base updates and users to provide content feedback.
Availability	An evaluation of the degree to which the system is available when needed due to likelihood of component failures or interface issues.

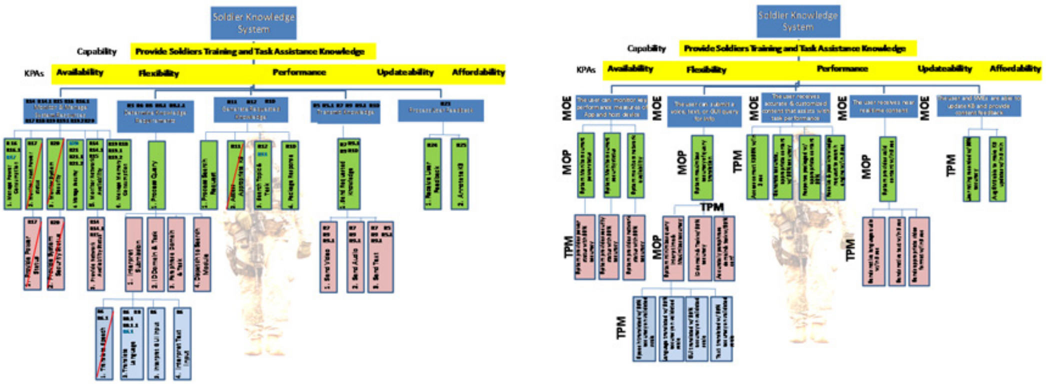
Scenario 1: A soldier enters a query (GUI/text/native language verbal) via a user interface. The query is processed, classified, and sent to the analytical engine for KB search. The analytical engine identifies the requested information and passes it to the content packager and message processor to send a query response to the user in the requested format (GUI/Text/Natural Language audio).

Scenario 2: A soldier receives content from the app regarding route reconnaissance information. The content was of marginal use. The soldier enters, via user interface, feedback (star rating) of the content provided. The KB and analytical engine tag the content with the appropriate rating for the information provided.

Each TPM is mapped to a customer validate value function. Three case study specific TPM value functions are shown in Figure 3. Each KPA has a degree of membership ranging from Unacceptable to Excellent. Membership functions for each three KPAs are shown in Figure 4. Fuzzy rules were developed to determine the fuzzy system architecture assessment score. Table 3 lists a subset of the fuzzy rules that map KPA fuzzy scores to Architecture assessment membership functions.

Table 3. Fuzzy Architecture Rules.

If Affordability is Unacceptable or Performance is Unacceptable or Flexibility is Unacceptable or Updateability is Unacceptable or Availability is Unacceptable, then the System is Unacceptable
If Affordability is Excellent and Performance is Excellent and Flexibility is Excellent and Updateability is Excellent and Availability is Excellent, then the System is Excellent
If Affordability is Acceptable and Performance is Acceptable and Flexibility is Acceptable and Updateability is Acceptable and Availability is Acceptable, then the System is Excellent
If Affordability is Acceptable and Performance is Acceptable and Flexibility is Acceptable and Updateability is Acceptable and Availability is Acceptable, then the System is Excellent
If Affordability is Marginal and Performance is Marginal and Flexibility is Marginal and Updateability is Marginal and Availability is Marginal, then the System is Unacceptable



Functional Architecture 1 Tree of Attributes 1

Figure 2. Alternative Functional Architecture and Tree of Attributes.

An alternative functional architecture and accompanying tree of attributes are generated and depicted in Figure 2. Major changes in Functional Architecture 1 are combining the manage and monitor power requirements (16, 16.1,17), combination of the speech and language translation requirements (6.1,8,8.1,8.1.1), and combining KB access and search requirements (11,12). We should expect more potential errors for a combined speech and language translation but a shorter response time. Utilizing the assessment methodology presented, we assess the baseline and alternative functional architecture.

4. Results

The assessment framework presented was used to assess a baseline functional architecture and a functional architecture alternative for a soldier knowledge transfer system. The functional architecture alternative combined management and monitoring power requirements, combined speech and language translation requirements, and combined KB access and search requirements. The assessment methodology was able to distinguish and assess value of these competing functional architectures. The baseline functional had an overall score of 50 while the alternative scored 47.5 highlighting that the baseline was the best functional architecture to carry forward to system architecture development. The same methodology was used to assess the baseline system architecture and a system architecture alternative created by modification of selected interfaces. The baseline system architecture had an overall score of 95.8. The alternative system architecture added an interface between the message processor and the KB update module and the KB feedback modules. The added interface allows more accurate processing and classification of the feedback before the KB is updated or annotated and resulted in a higher architecture score of 96.

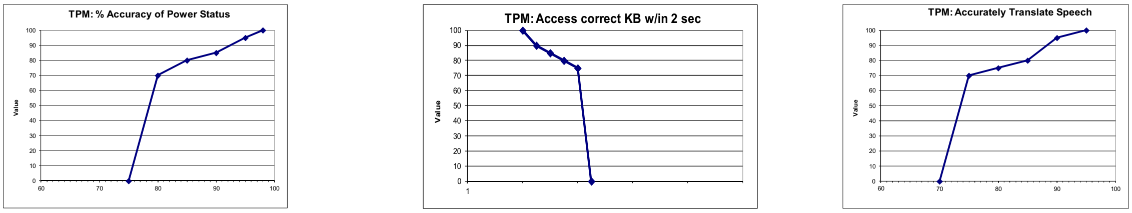
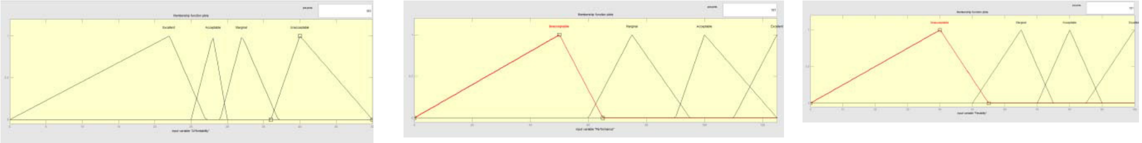


Fig. 3. Value Functions for Selected TPMs.

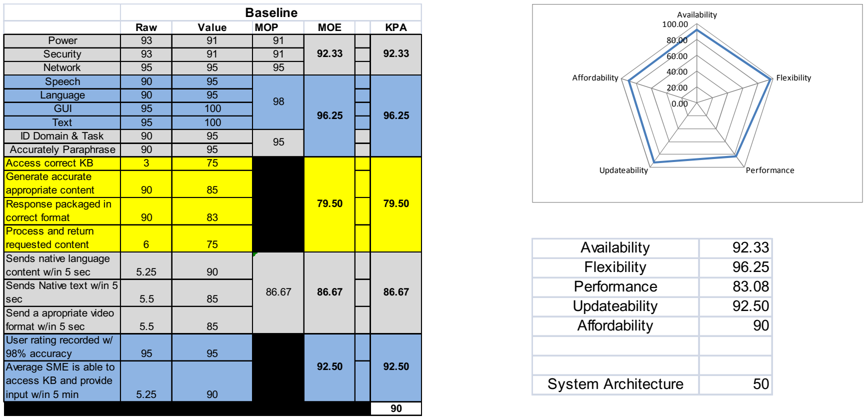


Affordability Membership Function Performance Membership Function Flexibility Membership Function

Fig. 4. Sample KPA membership functions.

5. Conclusions and Future Work

Since a majority of resources are obligated during the design phase of a system lifecycle, critical assessment of candidate functional and system architectures is vital to identify optimal architectures before proceeding to subsequent lifecycle phases. A common challenge associated with evaluation of system architectures is that key performance attributes are particularly “fuzzy” and qualitative in early architecture development. Previous architecture assessment methods often require application of different assessment models as abstraction is refined from function to system to elements. Additionally, explicit customer validated TPM value is not a fundamental factor in TPM, MOP, and MOE assessment. A method that can be utilized at each level of design abstraction and explicitly incorporate customer validated value of TPMs would be a useful method to assess system architecture alternatives.



	Baseline					
	Raw	Value	MOP	MOE		KPA
Power	93	91	91	92.33		92.33
Security	93	91	91			
Network	95	95	95			
Speech	90	95	98	96.25		96.25
Language	90	95				
GUI	95	100				
Text	95	100				
ID Domain & Task	90	95	95			
Accurately Paraphrase	90	95				
Access correct KB	3	75		79.50		79.50
Generate accurate appropriate content	90	85				
Response packaged in correct format	90	83				
Process and return requested content	6	75				
Sends native language content w/in 5 sec	5.25	90	86.67	86.67		86.67
Sends Native text w/in 5 sec	5.5	85				
Send a apropriate video format w/in 5 sec	5.5	85				
User rating recorded w/ 98% accuracy	95	95		92.50		92.50
Average SME is able to access KB and provide input w/in 5 min	5.25	90				
						90

Availability	92.33
Flexibility	96.25
Performance	83.08
Updateability	92.50
Affordability	90
System Architecture	50

Figure 5. Baseline Architecture Assessment

This paper presents a methodology to assess functional and system architecture that employs customer validated TPM value functions coupled with fuzzy assessment of key performance attributes to distinguish among architecture alternatives in an Excel and MATLAB environment. A soldier knowledge acquisition system was used as the case study for this framework. The methodology can be easily replicated with basic Excel and MATLAB tools. Unlike other methods that require different assessment techniques at each design level of abstraction, the methodology presented is flexible enough to assess architectures early in the process when things are "fuzzy" as well as later when subsystem and component performance are well defined. The methodology can also be used to analyze and assess impacts of interface changes within the system architecture. A distinct advantage of the methodology presented is the explicit inclusion of the customer in the assessment process through validation of the TPM value functions. Lastly, the methodology is domain independent and can be coupled with executable models linked to scenarios. The presented methodology does facilitate weighting of TPMs, MOP, MOE, and KPAs. However, experimentation with various weighting schemes for TPMs, MOP, MOE, and KPAs and their impact (and sensitivity) on architecture assessment results could be a useful methodology extension.

References

1. Acheson, P., Dagli, C., Kilicay-Ergin, N. (2013, March 19-22). Model Based Systems Engineering for Systems of Systems Using Agent-Based Modeling. Presented at the Conference on Systems Engineering Research (2013). doi: 10.1016/j.procs.2013.01.02.
2. Dojutrek, M., Labi, S., Dietz, J. A Fuzzy Approach for Assessing Transportation Infrastructure Security.
Elias, G., Jain, R. Attributes Guided System Architecture Assessment. INCOSE 2010.
3. Giammarco, K. (2014, March 21-22). A formal method for assessing architecture model and design maturity using domain-independent patterns. Presented at the Conference on Systems Engineering Research (2014). doi: 10.1016/j.procs.2014.03.068.
4. Haris, K., Dagli, C. Adaptive Reconfiguration of Complex System Architecture. Complex Adaptive Systems, Volume 1. 2011.
5. Kalawsky,R., Joannou,D., Tian, Y., Fayoumi, A. (2013, March 19-22). Using Architecture Patterns to Architect and Analyze Systems of Systems. Presented at the Conference on Systems Engineering Research (2013). doi: 10.1016/j.procs.2013.01.030.
6. Kenney, Ralph L., "Creativity in MS/OR: Value-Focused Thinking-Creativity Directed toward Decision Making", The Institute of Management Sciences, Los Angeles, California, 1993.
7. Pape, L. (2013, March 19-22). A Fuzzy Evaluation Method for Systems of Systems Meta-Architectures. Presented at the Conference on Systems Engineering Research (2013). doi: 10.1016/j.procs.2013.01.026.
8. Purewal, K., Yang, L., Grigg, A. Quantitative Assessment of Quality Attributes in Systems Architecture using Evidential Reasoning. 7th Annual Conference on Systems Engineering Research (2009).
9. Singh, A. and C. H. Dagli, Using quality attributes and computational intelligence to generate and evaluate system architecture alternatives, in 4th Annual IEEE Int. Systems Conf., San Diego, CA, 2010.