Paper #198

Evolutionary Computation Applied to System Architecture Development

Joseph J. Simpson
Missouri University of Science and Technology
6400 32nd Ave. NW #9, Seattle WA, 98107
jjs-sbw@eskimo.com

Dr. Cihan H. Dagli
Missouri University of Science and Technology
Eng. Man. Building 229
600 W. 14th Street, Rolla, MO 65409-0370
dagli.mst.edu

Abstract

Evolutionary algorithms and computational intelligence are developing sciences that have a strong potential to assist in the generation and evaluation of system and system-of-systems architectures. General techniques used by system engineers are integrated with computational intelligence methods to expand the ability of systems engineers and architects to effectively manage large, imprecise, and complex system-architecture development activities. This paper explores a basic set of conceptual, semantic and computation techniques based on computational intelligence that supports system architecture evaluation and development.

Introduction

The art of system architecture development is based on accumulated professional experience and an established set of heuristics that guide the behavior of the system architect. Computational intelligence and evolutionary computation represent a developing science that can be effectively applied to assist in the generation and evaluation of system and system-of-systems architectures. A primary task of the system architect is the design of a system architecture that meets the customer needs using techniques and materials that are provided by the systems engineering community. While the foundation of the art of system architecting is experience, which can be encoded into heuristics, a mathematical approach was developed by John N. Warfield to help organize and structure the system problem space (Warfield 1994). This structured mathematical approach incorporated binary relationships and Boolean mathematics to effectively manage the complexity associated with large-scale system development activities. A general technique used by systems engineering professionals is a binary matrix representation of a system or system-of-systems. The specific meaning and semantics of the binary relationship depends of the type of representation used. Typical representations are, “N squared”, design structure matrix, dependency structure matrix, and implication matrix. A key feature of these typical representations is their direct relationship to the structure required in an evolutionary computational approach. Evolutionary algorithms can be applied to the evaluation and optimization of these matrix structures. An evolutionary algorithm has been developed that focuses specifically on the generation and evaluation of systems and system-of-systems. This evolutionary algorithm incorporates a fuzzy inference system in the calculation of the best fit evaluation.

Conceptual System Architectural Definition

The early phases of architectural development are focused on the definition of the problem the system architecture will solve as well as the development of customer values and system context definition. Interpretive structural modeling (ISM) is a well defined method used to manage complex problems and activities. While ISM is used in the early conceptual and planning phases of an activity it is general enough to be applied in almost any context where a group of people must address a set of complex problems. ISM techniques are supported by a number of software packages that accept user input, perform the mathematical transforms and provide a graphical representation of the controlling system relationship (Warfield 2003). It is necessary to have direct access to the mathematical functions associated with the ISM approach to effectively integrate the ISM methods and the processes of individual computational intelligence techniques. To support direct access to the ISM foundational concepts, the monolithic ISM software approach must be modified to create a more modular software component approach. The modular software components are organized around a specific system structural relationship. The software component associated with a system relationship is called an abstract relation type (ART) (Simpson 2007). Each ART has a mathematical representation of the system structural relationship as well as a set of methods that operate on the mathematical representation.

The evaluation of new system development program must consider a number of contextual factors and characteristics. Two general categories of contextual analysis are associated the customer values and the existing systems that exist in the context of the proposed new system. The relationship among the customer, system architect and system engineer is shown in Figure 1. The rest of this paper will use the City of Seattle as the customer, the IBI Group as the system architect and Black & Veatch as the system engineering firm.

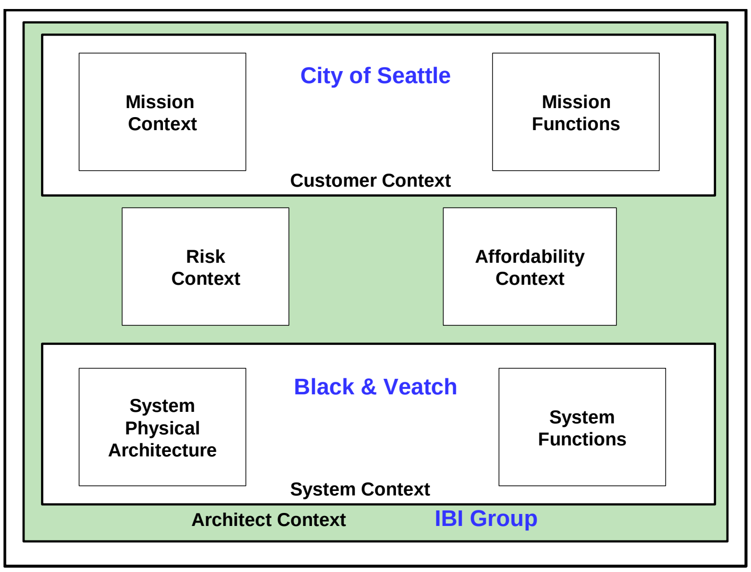

Figure 1. Customer, System Architect and System Engineer Relationships.

Discovering and documenting the City of Seattle's general value set as well as decision values associated with a specific project are two of the system architects foundational activities. A large, diverse customer like the City of Seattle has a complex set of goals, priorities and values. The ISM approach has a well defined approach to gathering and structuring value and priority relationships from large diverse groups. A representative percentage of the customer organization is engaged in an activity organized around the nominal group technique processes that is focused on identifying and prioritizing the customer values. Figure 2 shows an identified list of City of Seattle values in an unstructured matrix representation, a structured matrix representation, as

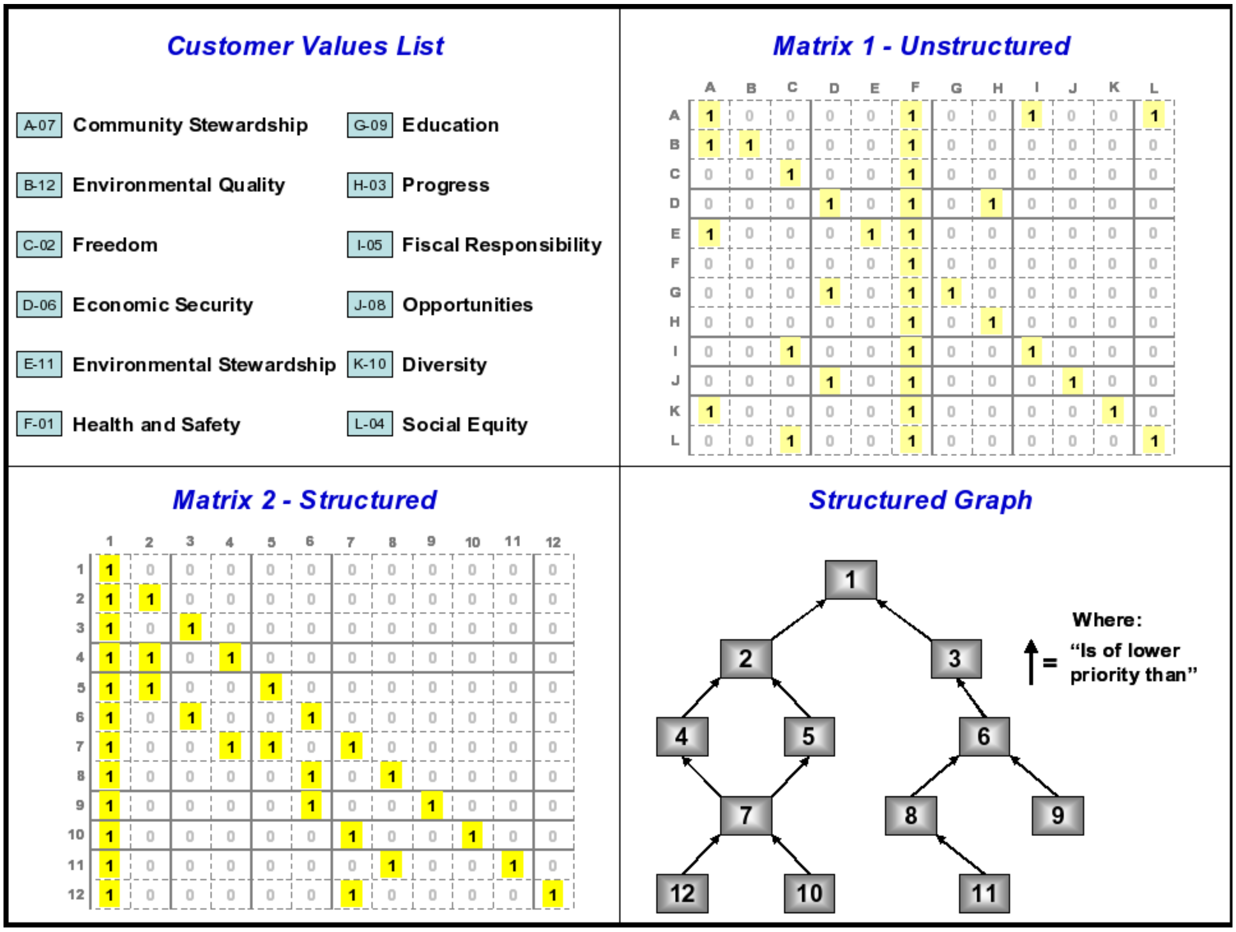

Figure 2. Customer Values as List, Matrix and Graph.

well as a prioritized graphical network structure. The unstructured matrix and the structured matrix entries both represent the same binary relationship. The row entries represent the "is a lower priority than" relationship, and the column entries represent the "is a higher priority than" relationship. A key advantage of this approach for the system architect and engineer is the existence of three equivalent representations of the same customer information: one in prose, one in matrix mathematics and one in graphical form. The model equivalence allows the same information to be clearly communicated to non-technical individuals, computer programs and mathematically-oriented, technical staff.

Formal concept analysis (FCA) has been used to specify the given context and interface set as a binary matrix. The FCA approach shares a number of characteristics with ISM methods and techniques. FCA can represent the same formal concept as prose, graphics and matrices. The entries in the FCA matrix represent a binary relation, and are focused on the incidence relation in the context (Ganter 1999). While FCA methods and techniques focus on a different view of the system and system context, integrating the FCA and ISM approaches will facilitate more powerful system and system-of-systems evaluation, architecting and engineering processes and techniques. The binary form and structure of an FCA concept lattice provides an excellent match with many evolutionary algorithm forms and structures. Evolutionary computational techniques are applied to assist the system architect and engineer in the evaluation of these complex configurations and interface sets. Figure 3 provides an example City of Seattle formal context for the execution of the Seattle mission functions.

City of Seattle Mission Profile (example formal context)

Attributes

Systems	A	B	C	D	E	F	G
National	X		X			X	
State	X		X			X	
County	X	X	X	X			X
Regional	X	X	X	X			X
City of Seattle	X	X	X	X	X		X
Urban Core		X			X		X
Urban Village		X			X		X
Suburban		X			X		X

Where
A = Land Use
B = Utilities
C = Transportation
D = Economic Development
E = Economic / Environmental Security
F = Environmental Impact
G = Growth Mgt Restrictions

Figure 3. Example City of Seattle Formal Context.

Customer Mission Functions and Measures of Effectiveness

Once the system architect has established the customer mission context, then the customer mission functions must be established and documented. The customer mission function tree is a graphical representation of the customer mission functions at different levels of abstraction. At the highest level the customer mission function is a single statement of purpose and/or function. At the next lower level of abstraction, the mission from the next higher level is decomposed into multiple mission functions that, when added together, equal the higher level mission function. This structured functional decomposition approach provides a mechanism to clearly determine the level of mission functional abstraction as well as the identification of any missing functional components. Every candidate solution system evaluated by the system architecture and engineering team will use mission function performance as the primary measure of effectiveness. The City of Seattle mission function tree is shown in Figure 4. The mission function tree has three levels of abstraction, and the sum of all functions at any one level of abstraction is equivalent to the top level mission function. The system architect must use experience and judgment when evaluating the mission functions in the context of the customer value set. Some mission functions may

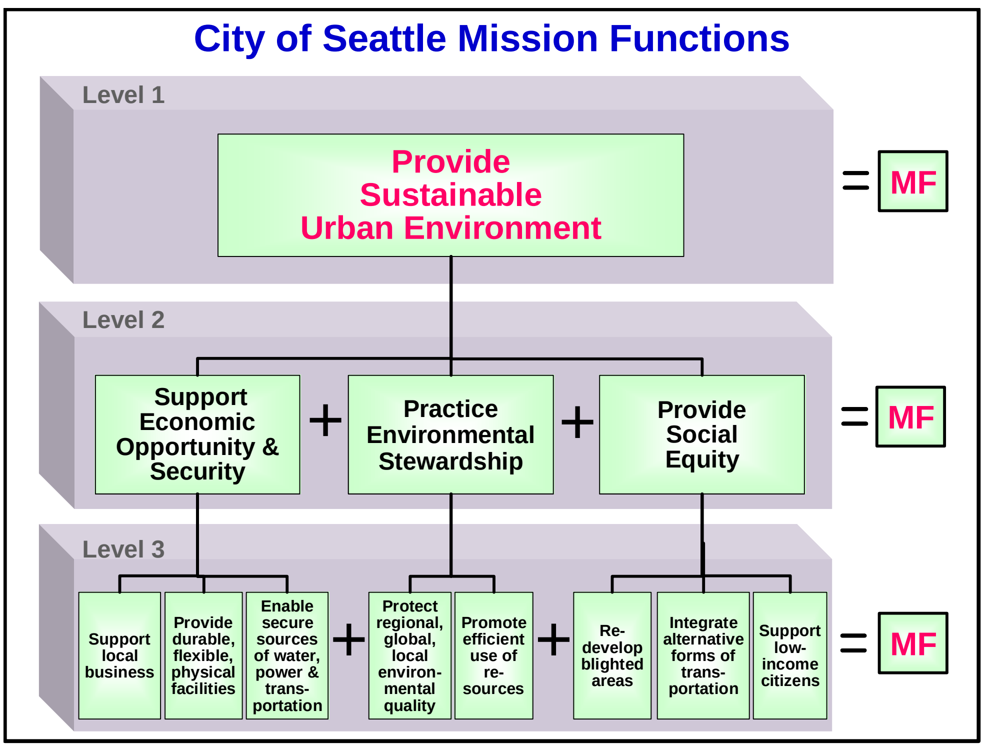

Figure 4. Customer Mission Functions.

carry more weight because of their relationship to the customer value set.

A standard measure of effectiveness (MOE) tree is shown in Figure 5. There are four basic elements in the MOE tree, operational effectiveness, risk, operational suitability and affordability. This MOE tree representation has been modified from the classical percentage of mission function performed, to a fuzzy metric in the form of the "degree to which" the mission function is performed. Specific mission functions are the focus of the operational effectiveness evaluation, while the more general operational issues and attributes are considered in the other categories.

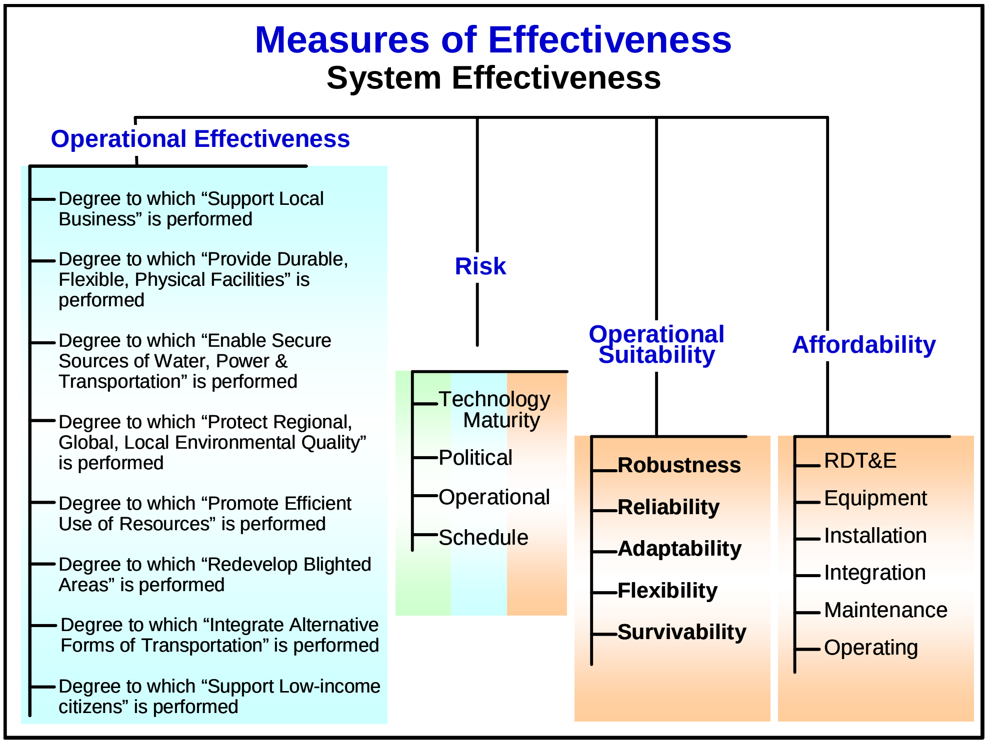

Figure 5. Measures of Effectiveness.

Evolutionary computing techniques are combined with fuzzy associative memories to create a hybrid computational intelligence algorithm that is applied to system-of-systems architecting tasks using a well defined set of measures of effectiveness (MOE). The outline of the computational intelligence algorithm is shown in Figure 6.

The system architect is responsible for the complete system operation and MOE balance, focused on life-cycle cost and risk. The customer is responsible for the mission profile and mission functions. Operational effectiveness and operational suitability areas are the responsibility of the systems engineers. Affordability, risk, operational effectiveness and operational suitability are the four MOE components used to evaluate the candidate system-of-systems architectures.

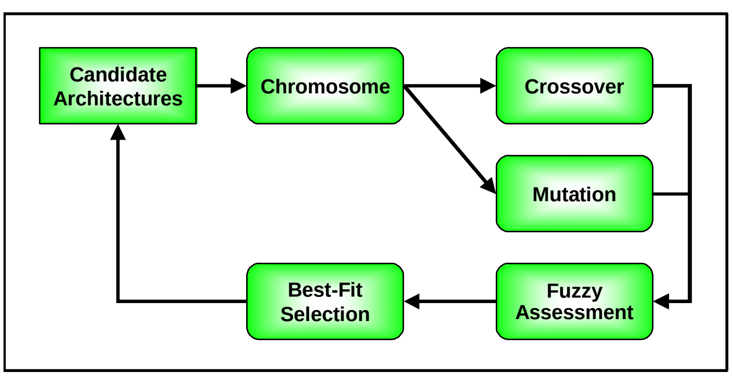

Figure 6. Outline of Computational Intelligence Algorithm.

System Representation

Each candidate architecture must be represented by a genotype, or chromosome. The chromosome content must be designed to hold all of the information associated with a successful system architecture and/or architectural design. As shown in Figure 7, a chromosome with a length of 44 units has been selected to represent this architectural generation effort. The physical system architecture is the primary architectural construct used to describe the system architecture and to support the system chromosome development. The chromosome is divided into two basic parts. Part One is 20 units long and represents system functions that are mapped to the City of Seattle mission functions. Part Two is 24 units long and represents a combination of the robustness, reliability, availability, flexibility, survivability and affordability system metrics. Risk factors can be considered by associating evaluation weights to specific system types or factored in after the computation is complete by the system architects and engineers.

As shown in Figure 7, Part One of the chromosome represents the physical system components that will perform the system functions necessary to fulfill the City of Seattle's mission functions. System component S1 performs mission function M1. System component S2 performs mission function M2. System component S3 performs mission function M3. System

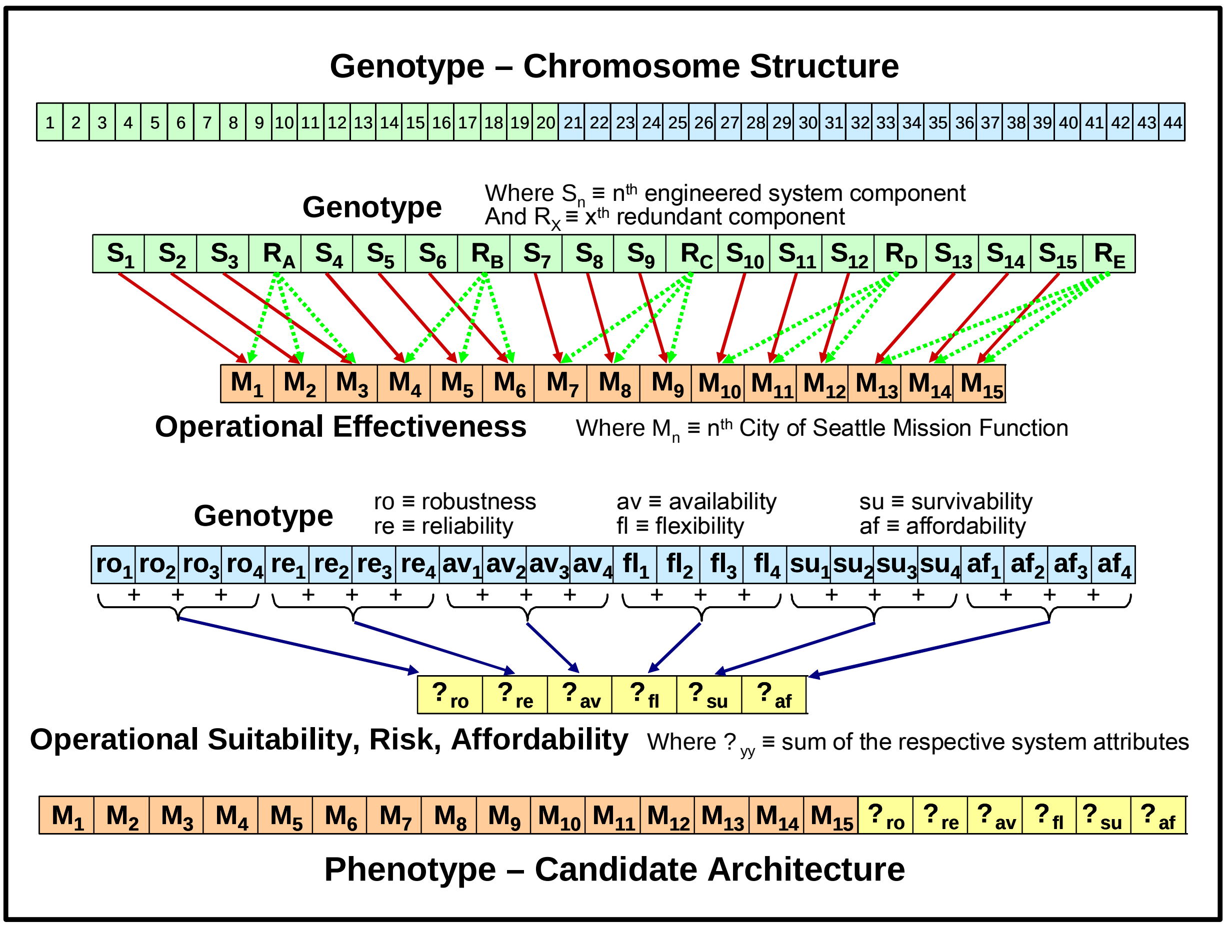

Figure 7. Chromosome Structure and Architecture Representation.

component RA will perform mission functions M1, M2 and M3. The RA system component provides the architecture with a redundant capacity. This operational redundancy provides the capability to create architectures that are adaptable, flexible and reliable. Therefore, physical architectures with redundant components will score high on operational effectiveness as well as in the other metric categories.

Best-Fit Function Fuzzy Inference

The fuzzy assessment and best-fit selection portion of the algorithm are based on the MOE's four subcomponents: operational effectiveness, operational suitability, risk and affordability. These four components are represented by a set of fuzzy inference rules that each have five specific fuzzy membership functions. These membership functions are: (1) very small, (2) small, (3) average, (4) large, and (5) very large. The fuzzy inference rule set has a membership value axis that ranges from zero to one, as well as a value axis that ranges from 0 to 100. Best professional judgment and expert opinion is used to adjust and position each of the five fuzzy membership functions on the value range. The four fuzzy MOE inference rules are aggregated using a fuzzy aggregation rule to combine the four fuzzy rule sets into one MOE for the total system. These membership functions and aggregation functions are designed to be independent of the system context and system application domain. However, there may be some sensitivity to the type of system relationship being evaluated. Therefore, the system architect and engineer will need to be aware of the system relationship attributes.

Summary

Evolutionary computation and computational intelligence are technologies that have a solid scientific foundation, and are well placed to provide the system-of-systems architect a powerful system design and evaluation tool. Classical system engineering tools and techniques have a clear, structured interface to many computational intelligence techniques. When combined with digraphs and other graphical representations of the matrix form, the techniques outlined in this paper provide a powerful tool for the communication of complex system interactions to large system design and evaluation teams. The ever increasing availability and cost effectiveness of computing capability provides additional motivation for the exploration and development of evolutionary computation in system-of-systems architecting, design and evaluation. More research is needed to explore and discover the most effective combination of computational intelligence tools to apply to system architecting and engineering tasks.

References

Ganter, B., Wille, R., *Formal Concept Analysis, Mathematical Foundations*. Springer-Verlag, New York, 1999.

Simpson, J. Dagli C., Miller A., Grasman S., Enke, D., "Development and Application of Abstract Relation Types for Use in Systems and System-of-Systems Design and Evaluation." *Proceeding of the Seventeenth Annual International Symposium of the International Council On Systems Engineering* (San Diego, CA, July, 2007)

Warfield, John N., *The Mathematics of Structure*. AJAR Publishing, Palm Harbor, FL, 2003.

Warfield, John N., *A Science of Generic Design, Managing Complexity Through System Design.* Iowa State University Press, AMES, IA,1994.

Biography

Joseph J. Simpson's interests are centered in the area of complex systems including system description, design, control and management. Joseph has professional experience in several domain areas including environmental restoration, commercial aerospace and information systems In the aerospace domain, Joseph has participated in a number of system development activities including; satellite based IP network design and deployment, real-time synchronous computing network test and evaluation, as well as future combat systems communications network design. Joseph Simpson has a BSCE and MSCE from the University of Washington, an MSSE from the Missouri University of Science and Technology, is a member of INCOSE, IEEE, and ACM. Currently Joseph is enrolled in a system engineering doctorate program at the Missouri University of Science and Technology.

Dr. Cihan H Dagli is a Professor of Engineering Management and Systems Engineering and Director of the System Engineering graduate program at the Missouri University of Science and Technology. He received BS and MS degrees in Industrial Engineering from Middle East Technical University and a Ph.D. from the School of Manufacturing and Mechanical Engineering at the University of Birmingham, United Kingdom, where from 1976 to 1979 he was a British Council Fellow. His research interests are in the areas of Systems Architecting, Systems Engineering, and Smart Engineering Systems Design through the use of Artificial Neural Networks, Fuzzy Logic, and Evolutionary Programming. He is founder of the Artificial Neural Networks in Engineering (ANNIE) conference held in St. Louis, Missouri since 1991, providing a conduit to dissemination of neural networks applications in engineering and decision making for the last fourteen years. He is the Area editor for Intelligent Systems of the International Journal of General Systems, published by Taylor and Francis, and Informa Inc.