Paper #90

Development of Abstract Relation Types for Systems and System-of-Systems Evaluation

Joseph J Simpson
Systems Concepts
6400 32nd Avenue N.W., #9
Seattle, WA 98107
(253) 773-3941
jjs-sbw@eskimo.com

Dr. Scott E Grasman
University of Missouri-Rolla
219 Engineering Management and Systems Engineering Department
Rolla, MO 65409-0370
(573) 341-7011
grasmans@umr.edu

Dr. Cihan H Dagli
University of Missouri-Rolla
229 Engineering Management and Systems Engineering Department
Rolla, MO 65409-0370
(573) 341-4374
dagli@umr.edu

Dr. David L Enke
University of Missouri-Rolla
219 Engineering Management and Systems Engineering Department
Rolla, MO 65409-0370
(573) 341-4565
enke@umr.edu

Dr. Ann Miller
University of Missouri-Rolla
Department of Electrical and Computer Engineering
Rolla, MO 65409-0040
(573) 341-6339
annmiller@ieee.org

Abstract

Abstract relation types (ART) are used to represent, describe, and evaluate systems, as well as establish a computational framework for systems and system-of-systems. The concept of an ART is built on two fundamental ideas: binary relations and abstract data-types. These two ideas are combined to create an ART that provides a well-defined, structured and executable representation of a system or system-of-systems. The complete, holistic systems evaluation approach is based on six ART constructs: context, concept, function, requirement, architecture and test (CCFRAT). The binary matrix component of the ART is used to encode specific system relationships in a manner that clearly separates system structure from system value. The method components of the ART are used to specify the standard acceptable operations included in the ART. The description component of the ART is used to clearly describe each of the three primary components of the ART and the typical application of the ART.

Abstract Relation Type Design

General Components. The ART design has three primary components: binary matrix, methods, and description components. The binary matrix component is used to encode the

specific binary relation associated with the ART. A minimum of one (1) binary matrix is required to represent the structure of the binary relation. However, other binary matrices may be added to represent values associated with the binary relation. The methods component is used to record and communicate the acceptable operations and actions associated with the ART binary matrices. These methods are designed to be implemented in executable computer code. The ART description component details the ART function, purpose and typical application. All three of the ART components integrate in a manner that creates a well-defined, executable system description and evaluation tool.

Binary Relations. The application of binary relations in the practice of system engineering and systems science was introduced by John N. Warfield (Warfield, 1973). Binary relations have two primary types: 1) set A mapped onto itself [set A X set A] and 2) set A mapped onto set B [set A X set B]. While Warfield focused on the first type, set A X Set A, ART expand the use of binary relations to the second type, set A X set B.

The binary relation and associated binary matrix as developed by Warfield (Warfield, 2003) creates a square N by N matrix, where N is the number of elements in set A. A binary matrix **M** is represented by four sets; **M = {*Iv, Ih, IvxIh, EIvxIh*},** where:

Iv is an ordered vertical index set
Ih is an ordered horizontal index set
IvxIh is the set of all ordered pairs of ***Iv,Ih***
EIvxIh is the entry set of the matrix.

A binary relation is represented by a binary matrix **M** that defines a two-block partition **{R; ~R}** on ***IvxIh***, such that all ordered pairs in ***IvxIh*** for which the entry is 1 are in the first block, and all other ordered pairs are in the second block. The first block represents the binary relation **R** on ***IvxIh*** and the second block is the complementary relation **~R** on ***IvxIh***. Some contextual relations developed by Warfield are: "is included in," "is antecedent to," "is subordinate to," and "is adjacent to." These relations are mapped to a binary matrix and used in the analysis of the structure of complex interactions and systems (Warfield, 1974).

The second type of binary relation, set A x set B, has been applied in various systems evaluation activities. One such application is the "formal context" developed by Ganter and Wille (Ganter, 1999). A formal context is composed of two sets and an incidence relation between the two sets: formal context **FC = (A, B, I)**, where:

Set **A** elements are called objects,
Set **B** elements are called attributes,
Set **I** members indicate the incidence relation between Set **A** and Set **B.**

Using this notation, the incidence relation between object **a** (member of Set **A**) and attribute **b** (member of Set **B**) is given as **(a, b)** is a member of **I.** This is only one, of many possible, applications of the set A x set B binary matrix and binary relation application.

The specific binary relation as well as the matrix representation of the relation must be clearly defined and documented in each ART. These aspects will be explored in further detail in the ART examples.

Six Basic ART Construct Classes. Six system views have been developed and used to describe and represent a system and/or system-of-systems (Simpson, et al 2005) These six system views are used to organize ART into similar type classes. One or more ART can be developed for each of these six classes, so each of the six system views will be briefly described next.

The context view of a system is focused on the systems external connections and interactions starting at the system boundary and looking outward from the system. This view records external systems, stakeholders

and value sets as well as other system contextual information.

The concept view of a system is focused on the internal aspects of the system starting at the system boundary and looking inward into the system. The concept view focuses on the internal system abstraction, processes and semantic rule sets.

The function view of a system is focused on system behaviour and activities at the given level of system abstraction and definition. The function view represents the system functions under the processes and rules established by the associated system concept view.

The requirement view of a system is focused on the performance requirements associated with each system function. In general the requirement view details how well each function in the function view must be performed. The exact relation between the function view and the requirement view is given by the concept view.

The architecture view of the system details the system architecture and how the architecture performs the system functions. The architecture representation mechanisms are given by the concept view.

The test view of the system details the activities necessary to determine that the selected architecture performs the system functions to the level required by the system requirements.

The Context ART (XART) Class

Two XART are outlined next, (1) a basic XART that represents the systems in the given context and (2) a connection XART that illustrates the connections between and among the systems in the defined context.

The basic XART consists of a structural binary relation constructed from the external systems in the context, set A, and the attributes associated with these systems, set B. As shown in Figure 1, the context for system 4.1.1 includes systems 4.1, 4.1.2, 4.1.3 and 4.1.4. While the context for system 4.1 includes systems 4, 4.2, 4.3 and 4.4.

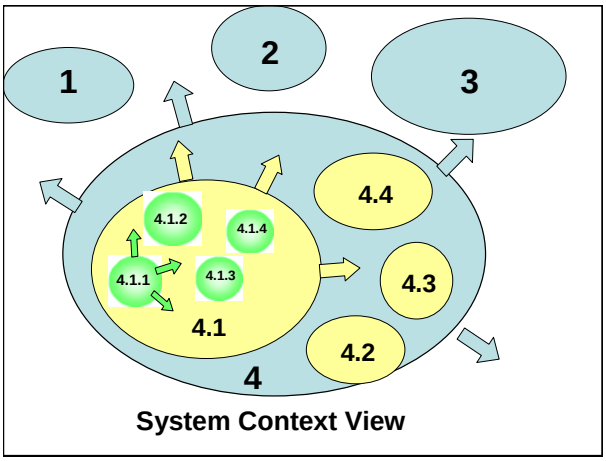

Figure 1. System Context View.

Important contextual system attributes associated with the systems shown in Figure 1 are given in two groups: physical attributes and service attributes. The physical attributes are: a) radio frequency connection, b) internet physical port, c) electrical connection and d) manual interface. The service attributes are: e) IP routing, f) FTP service, g) HTTP service, and h) SSL service. Figure 2 shows the binary matrix for the basic XART for Set A systems. Methods for evaluating and operating on the binary matrix are contained in the methods section of the basic XART. Typical binary matrix operation methods are: create matrix, add item to system set, add item to attribute set, and evaluate attribute set.

Set B -- System Attributes

Set A -- Systems	Physical Attributes				Service Attributes			
	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
4	1	0	1	1	1	1	1	1
4.2	1	0	1	1	1	1	1	1
4.3	1	1	1	1	1	1	1	0
4.4	1	1	1	1	1	1	0	1
4.1	1	1	1	1	1	1	1	1
4.1.1	0	1	0	0	1	1	0	0
4.1.2	0	1	0	0	1	1	0	0
4.1.3	0	1	0	0	1	1	0	0
4.1.4	0	1	0	0	1	1	0	0

System 4.1.1 context is bounded by system 4.1.
System 4.1 context is bounded by system 4.

Set A elements are systems in the context.
Set B elements are attributes of the systems in the context.

Figure 2. Binary Matrix, Set A X Set B.

A system connection binary matrix is the primary component in the connection XART. The system connection matrix is set A mapped onto set A, square binary matrix. Figure 3 provides an example of a system connection matrix for the system represented in Figure 1. From Figure 3 it can be seen that system 4.1 is connected to all the other system components of system 4.

Set A -- Systems	4	4.2	4.3	4.4	4.1	4.1.1	4.1.2	4.1.3	4.1.4
4	1	1	1	1	1	0	0	0	0
4.2	1	1	1	1	1	0	0	0	0
4.3	1	1	1	1	1	0	0	0	0
4.4	1	1	1	1	1	0	0	0	0
4.1	1	1	1	1	1	1	1	1	1
4.1.1	0	0	0	0	1	1	1	1	1
4.1.2	0	0	0	0	1	1	1	1	1
4.1.3	0	0	0	0	1	1	1	1	1
4.1.4	0	0	0	0	1	1	1	1	1

Figure 3. Binary Matrix, Set A X Set A.

The binary matrix representation of system relations provides a clear, effective system notation that increases the fidelity and precision of system design and development communication. The key to creating and using this increased communications capability is the development - by the systems engineering community - of a set of standard, accepted XART constructs. Once the XART constructs have been well defined and communicated then they may be used in a wide variety of system development tasks.

The Concept ART (CART) Class

Each member of the CART class focuses on the interior system boundary and the interior organization of the system. When the CART is combined with the XART a complete system boundary description is created. Partial system boundary descriptions can be given by using either the CART or the XART individually.

The primary purpose of the CART is the definition of the system internal components and representation. The internal system representation can be developed in one of two, mutually-exclusive approaches. The first approach is to represent the internal system components as other systems each of which has a CART and a XART, but on other internal system details. Figure 4 shows a network of systems represented by the XART-CART boundary.

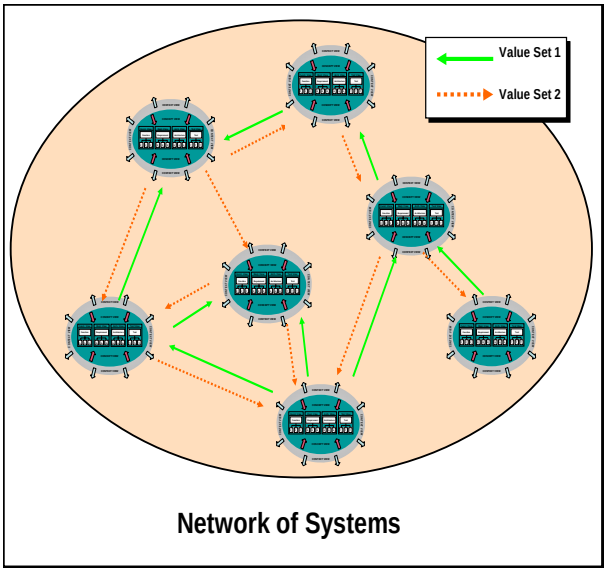

Figure 4 – XART-CART Boundary

The second approach to representing internal systems is the utilization of the function, requirement, architecture and test views of a system. The system internal aspects are detailed in terms these four system aspects in the second approach. The function-requirement aspects describe the problem the system is solving, while the architecture-test aspects detail the mechanism that is used to solve the problem and the tests used to verify the problem solution. Each of these approaches has advantages and disadvantages that depend directly on the system or system-of-systems under consideration. Figure 5

partially depicts the relationships among the six ART classes.

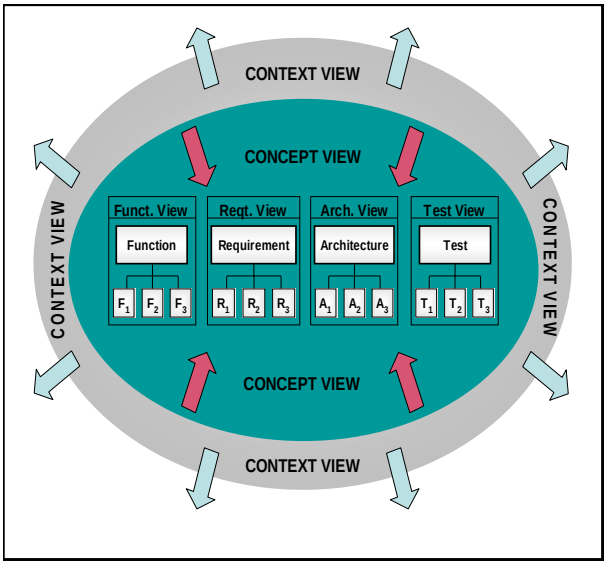

Figure 5 – Six ART Classes

Functional Hierarchy Abstract Relation Type (HART)

Function identification and functional analysis are found at the foundation of the system engineering process and approach. Functional analysis is the recursive process of decomposing the functions and requirements from the system level down to the lower levels of the system. When the functional analysis process is complete the system is defined in functional terms. The HART is designed to support the systems engineering activities of functional identification and analysis. Identification of the needed functionality is a necessary first step in any system design activity. There is a direct logical connection between the functionality provided by the components of a system and the functions and performance delivered by the complete integrated system. The binary relation that is associated with the HART is the "is included in" and "is composed of" relation, which clearly documents the functional aspects of the total system and its components. Figure 6 shows the hierarchical decomposition of the "Execute Task" function.

Num	Function Name
0.0	Execute Task
1.0	Prepare for Task
1.1	Conduct Pre-Task Briefing
1.2	Prepare Personnel for Task
1.3	Prepare Equipment for Task
2.0	Transit to Task Execution Area
2.1	Prepare for Transit
2.2	Load Unit on Transport
2.3	Move to Task Area
3.0	Perform Task
3.1	Evaluate Task Area
3.2	Execute Task Objectives
3.3	Evaluate Task Effectiveness
4.0	Return from Task Area
4.1	Transit to Initial Area
4.2	Evaluate Unit Status
4.3	Record Lessons Learned

Figure 6. Example of Functional Hierarchy.

The Execute Task function is represented as the main function at level 0.0, or the highest level of abstraction. Four functions are used to represent the main function at the next lower level of functional abstraction. These four functions are: Prepare for Task, Transit to Task Execution Area, Perform Task and Return from Task Area. Each of these functions is again decomposed to further elaborate the system function at the next lower level of abstraction. The core of the HART is a binary matrix and associated methods that are used to represent the structure of the functional relation. These computational methods and operations provide a mechanism to calculate, compute and reason about functional hierarchical relations.

Key characteristics and attributes of the functional structure can be precisely described, computed and communicated once the functions have been organized into a hierarchy that is represented by a mathematical matrix. The matrix form

provides access to an existing set of mathematics for use in performing calculations and operations about the HART. The natural language relations and semantics associated with the HART mathematical matrix operations gain increased utility as larger and larger groups of engineers learn how to use, interpret and apply the knowledge encoded in each matrix. Some of the system characteristics and attributes that can be encoded in a binary matrix are: level of abstraction, functional completeness at any level of abstraction, functional sequence and functional timing as well as functional interface and connection information.

The key component of a HART is a binary matrix that maps the given function set F onto itself, set F. Figure 7 shows the translation of the execute task function functional hierarchy listed in Figure 6, into a binary "is included in" and "is composed of" binary relation matrix. This binary relation matrix is read down the columns as "is included in." For example when reading down Column 2, Function 0.0 is composed of all the other functions. Or viewed in another way, every function is included in Function 0.0. Figure 7 shows three complete levels of functional decomposition, therefore Function 0.0 is represented three times, once at each level of abstraction. Cell (2,2) shows the highest level of abstraction and the function is completely included in itself at this level. Cells (2,3), (2,4), (2,5), and (2,6) represent the complete function 0.0 at the second level of abstraction. Cells (2,7), (2,8), (2,9), (2,10), (2,11), (2,12), (2,13), (2,14), and (2,15) represent the complete Function 0.0 at the third level of abstraction. The correct level of abstraction and the correct functional grouping can be determined by calculating the number of cells in each row that contain the number 1.

The binary matrix structure for the HART provides the foundation for other operations and analysis techniques that are used to evaluate the given system structure. Different types of functional grouping can be applied.

	0.0	**1.0**	**2.0**	**3.0**	**4.0**	**1.1**	**1.2**	**1.3**	**2.1**	**2.2**	**2.3**	**3.1**	**3.2**	**3.3**	**4.1**	**4.2**	**4.3**
0.0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.0	**1**	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.0	**1**	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
4.0	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0
1.1	**1**	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0
1.2	**1**	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0
1.3	**1**	**1**	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0
2.1	**1**	0	**1**	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0
2.2	**1**	0	**1**	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0
2.3	**1**	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0
3.1	**1**	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	0
3.2	**1**	0	0	**1**	0	0	0	0	0	0	0	0	**1**	0	0	0	0
3.3	**1**	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0	0	0
4.1	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0	0
4.2	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	**1**	0
4.3	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	**1**

Figure 7 – Binary "Is Included In" Matrix

For example the level one Function 0.0 is static because there is just one function represented at this abstraction level nothing proceeds or succeeds Function 0.0. However, the level two functions shown in Figure 7 are sequential in nature. If the top level Function

0.0 has specific time, energy and/or other constraints, then the sequential logic at the second level of abstraction may be used to allocate and reason about these system constraints and aspects. The HART construct is the foundation for the development and standardization of sequential and concurrent function operators that can be used to communicate across large distributed system development teams in a precise controlled manner.

Requirement Abstract Relation Type (RART)

The concept of a requirement has been refined and constrained to address the degree or level of performance that must be accomplished by each function. The narrow definition of a requirement provides the basis for the development of requirement verification logic as well as system problem validation. The combination of the system function, what the system has to do, and the requirement, how well the functions need to be accomplished, creates the system problem statement. The RART binary matrix structure will mirror the HART structure upon which its form depends. The operations and methods associated with the RART will be different and distinct from the operations and methods associated with the system HART.

Physical Hierarchy Abstract Relation Type (PART)

The PART is based on the "is part of" or "is composed of" binary relation. The physical decomposition of a system is an example of a system architectural design activity that will produce this type of system ART representation. Figure 8 shows an example of physical system decomposition for a wagon. The hierarchical structure of the physical decomposition was selected to match the hierarchical structure of the functional decomposition shown in Figure 6 to highlight the fact that even though the system hierarchical structure is the same, the operations and semantics of the PART are different from, and have different meanings than, those operations and semantics associated with the HART.

Num	Component Name
0.0	Wagon
1.0	Wagon Body
1.1	Metal Floor
1.2	Metal Sides
1.3	Metal Fasteners
2.0	Wooden Sides
2.1	Wood Uprights
2.2	Wood Planks
2.3	Wood Fasteners
3.0	Front Wheel Assembly
3.1	Front Axel
3.2	Front Wheels
3.3	Front Handle Assembly
4.0	Rear Wheel Assembly
4.1	Rear Axel
4.2	Rear Wheels
4.3	Rear Axel Attachment Assembly

Figure 8. Example of Physical Hierarchy.

Physical properties and attributes are important aspects used in PART methods and computations. Weight is an example of a physical property that can be allocated, from the top-down, or computed, from the bottom up using the PART methods and operations. Using the "is a part of" and "is included in" binary relation, the wagon physical decomposition would look exactly like the binary matrix in Figure 7. The second column, of Figure 7, indicates that all of the physical components are included in the wagon. Similar to the HART methods, each row can be evaluated to calculate the proper level of physical decomposition. The

semantics associated with each relation will guide the ART method development.

An example of the PART used to represent and evaluate system weight is shown in Figure 9. In this case the numbers in the cells represent the allocated weight for each physical component. The total wagon weight, 26 units, is located in Cell (2,2). Summing the weight of all of the level three components; rows 7 through 18 again gives the expected weight of the complete system.

	0.0	1.0	2.0	3.0	4.0	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2	3.3	4.1	4.2	4.3
0.0	**26**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0	**10**	**10**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.0	**6**	0	**6**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.0	**6**	0	0	**6**	0	0	0	0	0	0	0	0	0	0	0	0	0
4.0	**4**	0	0	0	**4**	0	0	0	0	0	0	0	0	0	0	0	0
1.1	**4**	**4**	0	0	0	**4**	0	0	0	0	0	0	0	0	0	0	0
1.2	**4**	**4**	0	0	0	0	**4**	0	0	0	0	0	0	0	0	0	0
1.3	**2**	**2**	0	0	0	0	0	**2**	0	0	0	0	0	0	0	0	0
2.1	**2**	0	**2**	0	0	0	0	0	**2**	0	0	0	0	0	0	0	0
2.2	**3**	0	**3**	0	0	0	0	0	0	**3**	0	0	0	0	0	0	0
2.3	**1**	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0
3.1	**2**	0	0	**2**	0	0	0	0	0	0	0	**2**	0	0	0	0	0
3.2	**2**	0	0	**2**	0	0	0	0	0	0	0	0	**2**	0	0	0	0
3.3	**2**	0	0	**2**	0	0	0	0	0	0	0	0	0	**2**	0	0	0
4.1	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0	0
4.2	**2**	0	0	0	**2**	0	0	0	0	0	0	0	0	0	0	**2**	0
4.3	**1**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	**1**

Figure 9 – Physical Component Weight Allocation

Test Abstract Relation Type (TART)

Just as the function (HART) and requirement (RART) aspects of system design are combined to create the system problem statement, the physical architecture (PART) and test (TART) aspects of system design are used to specify the system solution to the stated system problem. The system architecture is the proposed system solution while the test activity is the mechanism that assures the system designers that the selected architecture will perform the system function to the degree specified by the system requirements.

The TART matrix structure will mirror the PART hierarchy structure because a test is prepared for each portion of the PART hierarchy. The TART methods and operations will support the specific system test activities for the given system.

Systems Engineering Executable Language (SEEL) Components

A system can be completely described using the six components presented in this paper. Building on the established function, requirement, architecture and test (FRAT) structured system engineering approach, the CCFRAT approach adds two new system views to clearly communicate information about the system boundary and context. The context view is used to organize and communicate the outward context from the system of interest. The concept view details the internal aspects of the system of interest.

A minimum of three systems must be addressed by the SEEL. These three systems

are: (1) the product system under design, (2) the development system that produces the product system and (3) the environment system that contains both of the other two systems. A complete set of CCFRAT methods and processes need to be applied to these three systems. The product system is usually defined in the most detail, with the production system detail being accomplished at a lower levels of detail, and the environmental system developed at a even a lower level of detail. The environmental system is a rich source of new science, technology and applications. The product system design team must maintain cognizance of the changes in the environmental system.

Incremental Development and Deployment Approach

Current systems engineering methods and tools support the development and application of the abstract relation types. The initial development of the ART needs to focus on both the development of the computer executable code and the development of the human system semantics. Excel spreadsheets are an excellent tool for testing and working with candidate ART constructs. Further ART development needs to make strong ties to model-based systems engineering. There are several basic approaches to model-based system engineering: the approach developed by Wayne Wymore (Wymore, 1993); approaches developed by Dave Oliver (Oliver, et.al., 1997) and the approach used by the CORE™ system engineering tool developed by the Vitech Corporation. The key to the development of human semantics that are also computer executable, is the understanding of model-based systems engineering techniques.

Summary and Conclusions

The development and application of system ART is a promising combination of classical structured systems analysis and computer-executable system evaluation techniques. More research is required to develop the ART executable code components as well as acceptable human semantics.

References

Ganter, Bernhard and Wille, Rudolf, *Formal Concept Analysis: Mathematical Foundations.* Springer-Verlag, NY, 1999.

Oliver, D. W., Kelliher, T.P., and Keegan, J.G., *Engineering Complex Systems with Models and Objects*. McGraw-Hill, New York, 1997.

Simpson, J. Dagli C., Miller A., Grasman S., "A Generic, Adaptive Systems Engineering Information Model", *Proceedings of the 15th Annual International Symposium of the International Council on Systems Engineering*, Rochester, NY, July, 2005

Warfield, John N., *An Assault on Complexity*, Battle Monograph No. 3, Battelle Memorial Institute, Columbus , OH, 1973.

Warfield, John N., *Structuring Complex Systems*. Monograph No. 4. Battelle Memorial Institute, Columbus , OH, 1974.

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

Wymore, A. Wayne, *Model-Based Systems Engineering*. CRC Press, Inc., Boca Raton, FL 1993.

Biography

Joseph J. Simpson's interests are centred 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 University of Missouri-Rolla, is a member of INCOSE, IEEE, and ACM. Currently Joseph is enrolled in a system engineering doctorate program at the University of Missouri-Rolla.

Dr. Cihan H Dagli is a Professor of Engineering Management and Systems Engineering and director of the System Engineering graduate program at the University of Missouri-Rolla. 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 the founder of the Artificial Neural Networks in Engineering (ANNIE) conference being held in St. Louis, Missouri since 1991. He provided the conduit to the dissemination of neural networks applications in engineering and decision making through these conferences 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.

Dr. Ann Miller is the Cynthia Tang Missouri Distinguished Professor of Computer Engineering at the University of Missouri – Rolla. Previously, she was the Deputy Assistant Secretary of the Navy for Command, Control, Communications, Computing, Intelligence, Electronic Warfare, and Space for the U. S. Department of the Navy. She also served as Director for Information Technologies, Department of Defense Research and Engineering. Dr. Miller chairs the NATO Information Systems Technology Panel and is a Senior Member of IEEE. Dr. Miller's research areas include reliability and security of computer-based systems, with an emphasis on networked large-scale systems.

Dr. Scott Grasman is an Assistant Professor in the Engineering Management and Systems Engineering Department at the University of Missouri – Rolla. Prior to joining UMR, he was an Adjunct Assistant Professor at the University of Michigan, where he received his B.S.E., M.S.E., and Ph.D. degrees in Industrial and Operations Engineering. His primary research interests relate to the application of quantitative models to manufacturing and service systems. He is a member of American Society for Engineering Education (ASEE), American Society for Engineering Management (ASEM), Decision Sciences Institute (DSI), Institute of Industrial Engineers (IIE), Institute for Operations Research and Management Science (INFORMS).

Dr. David Enke is an Associate Professor in the Department of Engineering Management and Systems Engineering at the University of Missouri-Rolla (UMR). He is the Director of the Laboratory for Investment and Financial Engineering and is a member of UMR's Intelligent Systems Center. Dr. Enke received his B.S. in Electrical Engineering and M.S. and Ph.D. degrees in Engineering Management, each from UMR. He is the co-chair of the Artificial Neural Networks in Engineering conference and is on the editorial board of the Engineering Economist. His research interests are in the areas of financial engineering, investment, and intelligent systems.